

**КОМПЬЮТЕРНАЯ
ТЕХНОЛОГИЯ
ОБУЧЕНИЯ
•
СЛОВАРЬ-
СПРАВОЧНИК
•**

КОМПЬЮТЕРНАЯ ТЕХНОЛОГИЯ ОБУЧЕНИЯ • СЛОВАРЬ- СПРАВОЧНИК •



Под редакцией

чл.-кор. АТН *Гриценко В.И.*,
д-ра техн. наук проф. *Довгялло А.М.*,
д-ра техн. наук проф. *Савельева А.Я.*

Киев
“Наукова думка”
1992

Словарь-справочник вводит в круг проблем, направлений исследований и разработок информационных технологий в системе образования — компьютерной технологии обучения (КТО). Содержит свыше 600 обзорных и терминологических статей по теории и практике КТО — междисциплинарной технологии, включающей дидактические и психолого-педагогические основы; эргономические, медицинские и санитарно-гигиенические рекомендации и нормативы на применение компьютеров в целях обучения взрослых и детей; основы проектирования обучения и решения задач в человеко-машинных системах; развернутую классификацию и квалифицированные описания лучших образцов программ учебного назначения и персональных компьютеров, пригодных для целей обучения.

Для преподавателей вузов, техникумов школ, инженерно-технических работников, а также других читателей, кто желает грамотно ориентироваться в развивающейся системе образования и воспитания. Особо следует отметить стремление авторов снабдить необходимыми сведениями о КТО тех читателей справочника, которые проявляют интерес к быстро растущему рынку программ учебного назначения, автоматизированных учебных курсов, видеокomпьютерных систем и др.

Научные консультанты - спецредакторы: *Г.А.Балл*, д-р психол.н., *В.М.Бондаговская*, канд.психол.н., *Л.А.Гаврилюк*, *Ж.Д.Гудзенко*, *Г.Ж.Желев*, доктор, *Э.Л.Ивахненко*, канд.техн.н., *А.С.Коваленко*, канд.мед.н., *С.П.Кудрявцева*, канд.техн.н., *Ю.И.Лобанов*, канд.техн.н., *Э.Мазак*, доктор, *Е.Д.Маргулис*, канд.психол.н., *В.Н.Никулин*, канд.техн.н., *В.А.Петрушин*, канд.физ.-мат.н., *А.Н.Писарев*, проф., *Э.Ш.Примакова*, *В.И.Сигалов*, канд.техн.н., *Е.М.Синица*, *А.Т.Хавро*, *В.-Х.Хартвиг*, доктор.

Отделение редакционной и рекламно-издательской
деятельности УкрЦЭНДИСИ.

К $\frac{1402010000-190}{221-92}$ 92
ISBN 5-12-003425-X

© Институт кибернетики
АН Украины, 1992

СОДЕРЖАНИЕ

Основные сокращения	4
Предисловие	5
Компьютерная технология обучения	7
Теоретические основы компьютерной технологии обучения	15
Задачный подход к компьютерной технологии обучения	16
Дидактические аспекты компьютерной технологии обучения	18
Психолого-педагогические аспекты компьютерной технологии обучения	26
Санитарно-гигиенические и эргономические основы компьютерной технологии обучения	30
Программное обеспечение компьютерной технологии обучения	33
Техническое обеспечение компьютерной технологии обучения	35
<i>Терминологические статьи,</i>	
<i>Приложение. Компьютерная технология обучения в странах Восточной Европы и на Кубе . . .</i>	628
Список литературы	641

ОСНОВНЫЕ СОКРАЩЕНИЯ

алгебр.	—	алгебраический
арифм.	—	арифметический
В., ВВ.	—	век, века
в т.ч.	—	в том числе
г., гг.	—	год, годы
геом.	—	геометрический
К	—	килобайт
к.-л.	—	какой-либо
лит.	—	литература
М	—	мегабайт
макс.	—	максимальный
матем.	—	математический
миним.	—	минимальный
наз.	—	называется, называемый
напр.	—	например
осн.	—	основной
рис.	—	рисунок
см.	—	— смотрите
табл.	—	таблица
теор.	—	теоретический

ПРЕДИСЛОВИЕ

Широкое распространение компьютерного обучения, одного из прогрессивных методов и средств обучения, привело к появлению большого количества публикаций и документации, в которых описывается состояние развития или использования этих средств, и следовательно, к появлению новых понятий и средств их выражения. При этом отставание решения вопросов, связанных с терминованием новых понятий, с унификацией лексики, используемой для составления документации, приводит к большим непроизводительным затратам времени на нахождения "общего языка" при описании программных средств учебного назначения, при общении с их пользователями и разработчиками.

В мае 1988г. в Киеве в Институте кибернетики им. В.М. Глушкова АН Украины был проведен международный семинар-совещание по разработке базы знаний по компьютерной технологии обучения. На этом семинаре при обсуждении состояния компьютерной технологии обучения (КТО) было принято решение о создании данного словаря. Далее, в ноябре 1988 на базе Софийского технического университета (бывшего - ВМЭИ "Ленин"), а в мае 1989 на базе Дрезденского технического университета были проведены семинары по разработке структуры словаря-справочника. Отграничив область исследования, определили исходное множество компонентов предметной области "Компьютерная технология обучения". Систематизация предметной области выявила также основные тематические разделы. Авторским коллективом рассмотрена концепция КТО, выработанная на основании материалов словаря, психолого-педагогические основы классификации средств КТО, уточнена классификация компьютерных программ учебного назначения.

Предлагаемый читателю словарь-справочник, созданный международным коллективом, представляет собой публикацию, интересную с трех точек зрения. Во-первых, это результат междисциплинарного исследования, в котором органически сочетаются дидактические, психолого-педагогические основы КТО и основы проектирования обучения и решения задач в человеко-машинных системах; эргономические, медицинские и санитарно-гигиенические рекомендации и нормативы на применение компьютеров в целях обучения взрослых и детей; развернутая классификация и квалифицированные описания лучших образцов программ учебного назначения, инструментальных средств для их разработки и современных технических средств КТО. Во-вторых, данный справочник иллюстрирует достигнутое взаимопонимание специалистов разных стран по выработке согласованной концепции и терминологии для такой бурно развивающейся области как компьютерное обучение. В-третьих, этот труд в его теперешней редакции вообрал в себя знания, которые выжили испытание временем, а также, образно выражаясь, "испытание событиями", происходившими в странах Восточной Европы и на территории бывшего СССР.

В результате словарь-справочник "Компьютерная технология обучения" дает полную и в то же время сжатую, разнообразную и унифицированную, популярную и строго научную информацию для всех тех, кто безусловно должен им заинтересоваться.

Кто они — читатели справочника? Это специалисты по новым информационным технологиям и преподаватели средней и высшей школы, разработчики программ учебного назначения и студенты вузов, администраторы учебных заведений ... — все, кто желает чувствовать себя компетентным в вопросах развивающейся системы образования и воспитания, в ее компьютеризации.

“Компьютерная технология обучения” делится на две части. Первая часть состоит из обзорных статей, отражающих сущность КТО как междисциплинарной технологии. Вторая часть содержит около 600 терминологических статей, расположенных в алфавитном порядке. В этих статьях более углубленно и детально изложены проблемы, рассматриваемые в обзорных статьях. Развитие компьютерной технологии обучения в некоторых странах освещено в Приложении.

В словаре применяется единая система отсылок на другие статьи, дополняющие разъяснения данного вопроса.

В написании статей принимали участие авторы из разных государств: Украины, Азербайджана, Литвы, Молдовы, России, Болгарии, Венгрии, Германии (восточные земли), Кубы, Польши, Румынии, Чехо-Словакии.

Предлагаемый словарь является первым опытом систематизации и описания знаний о предметной области “Компьютерная технология обучения”. Очень полезны все замечания и дополнения, которые просим присылать по адресу: 252207, Киев, проспект Академика Глушкова, 40, Институт кибернетики имени В.М. Глушкова АН Украины.

Благодарим всех авторов, научных консультантов, рецензентов и редакторов словаря.

А.М.Довгялло, З.Ш.Примакова.

КОМПЬЮТЕРНАЯ ТЕХНОЛОГИЯ ОБУЧЕНИЯ (англ. computer — компьютер, вычислитель, электронная вычислительная машина и греч. *τεχνη* — искусство, мастерство; *μάθησις* — учение, понятие), (КТО) — технология обучения, осн. зная на принципах *информатики* и реализуемая с помощью *электронных вычислительных машин* (компьютеров); совокупность средств (программного и технического обеспечения, теоретических знаний, методических приемов) и способов их применения для эффективной деятельности обучаемых и преподавателей при самостоятельной работе, на лекциях, практических и лабораторных занятиях и т.п.

Необходимость использования КТО вызвана стремительным ускорением научно-технического прогресса, обуславливающего необходимость информатизации системы образования. В результате усложнения и быстрого видоизменения технологий непрерывно увеличивается объём и изменяется содержание знаний, умений и навыков, которыми должны владеть специалисты, работающие в различных областях хозяйственной деятельности, науки и техники. Повышаются требования к качеству обучения. Особую актуальность приобретает вопрос непрерывной подготовки и переподготовки специалистов различных категорий с целью эффективного использования ими в своей деятельности вычислительной техники. Умение пользоваться компьютерами рассматривается как вторая грамотность. От уровня *компьютерной грамотности* и ее распространения зависят развитие прогресса, модернизация и интеллектуализация производства и системы образования. Разработка КТО — необходимое условие достижения целей информатизации образования, среди которых в первую очередь выделяют: обеспечение доступности *знаний и данных* для каждого члена общества; развитие интеллектуальных и творческих способностей обучаемых на основе *индивидуализации обучения*; обеспечение опережающей подготовки специалистов; овладение компьютерной грамотностью путем обучающего сопровождения информационных технологий; интенсификацию учебного процесса. Специфическими целями КТО на современном этапе ее развития являются: обеспечение интеграции учебной, научной и организационной деятельности учебных заведений на основе всеохватывающего применения компьютеров; достижение типовой и унификации учебно-методических, программных и технических средств с целью широкого использования КТО или ее компонентов в учебных и научных учреждениях, на производстве и в быту.

КТО предшествовали *программированное и автоматизированное обучение* 60-х гг., электронизация образования 70-х гг., компьютеризация системы образования, начавшаяся в 80-х гг. В англоязычных странах термину “компьютерная технология обучения” соответствуют термины “технология образования” (educational technology), “информатика в образовании” (informatics in education) и др.; во франкоязычных странах — “учебная информатика” (informatique

d'enseignement); в германоязычных странах — Technologie Computer gestutzter Ausbildung; в испаноязычных странах — “технология образования” (Technologia Educational).

В конце 80-х гг. в странах Восточной Европы, в республиках бывшего СССР получили распространение КТО, основанные на качественно новой концепции модульного многофункционального прагматического представления и поэтапной детализации и активизации знаний учебного назначения. *Представление знаний* с помощью компьютеров обычно связывают с *экспертными системами, системами, основанными на представлениях знаний* и методах *искусственного интеллекта*. Многофункциональное прагматическое представление знаний учебного назначения предполагает более широкую трактовку этого нового понятия: компьютерными знаниями могут быть не только знания названных систем, но и текстовый материал, размещенный в *памяти ЭВМ*, справочная информация по какому-либо предмету, программа контроля знаний обучаемых. Каждая форма представления знаний размещается в памяти компьютера, изменяется, дополняется с помощью специальных инструментальных средств.

В процессе развития вычислительной техники появлялись различные формы компьютерных знаний учебного назначения и инструментальных средств для их представления, происходила и эволюция форм представления знаний о самом компьютере (рис.). Компьютеры 50-60-х гг. создавались как большие арифмометры (“перемальватели” чисел) с неразвитыми *устройствами ввода—вывода информации*. Знания о компьютерах их пользователи черпали из печатной документации, а роль “инструментальной поддержки” этих знаний выполняли типографские машины. Первым важным шагом в компьютеризации знаний о компьютере было создание *встроенной документации, справочных материалов, руководств для пользователей с помощью текстовых редакторов*. Дальнейшее развитие знаний о компьютере связано с возникновением *информационно-справочных систем*, работавших в конце 60-х гг. в *режиме разделения времени* и обеспечивавших пользователя необходимыми сведениями о компьютере - командах *операционной системы*, составе библиотек программ, *языке программирования* и т.п. Инструментальной поддержкой информационно-справочных систем явились *системы управления базами данных*.

Расширение возможностей режима разделения времени позволило в начале 70-х гг. создать первые программы контроля знаний и сценарные *автоматизированные обучающие системы* (АОС). В последних во многих случаях был реализован на основе методик *программированного обучения* достаточно полный объем знаний, напр., о языках программирования. Инструментальной поддержкой АОС становятся специализированные языки (*КУРСРАЙТЕР, ТБЮТОР, Язык описания курсов*) и системы программирования,

Годы	Формы представления знаний	Инструментальные средства
	Экспертные обучающие системы	Пустая экспертная обучающая система
80-е	Решатели задач, экспертные системы	Инструментальные средства обработки знаний (ИНТЕР-ЭКСПЕРТ, ПРОЛОГ, естественно-языковые лингвистические процессоры)
Конец 70-х	Тренажёры, учебные игры, многофункциональные автоматизированные учебные курсы	Инструментальные пакеты прикладных программ (типа АОС/РДВ, АОС-ВУЗ/ПРИМУС), интерактивная графика
Начало 70-х	Программы контроля знаний, сценарные автоматизированные обучающие системы.	Специализированные языки (Курсрайтер и системы программирования АОС-ВУЗ)
Конец 60-х	Информационно-справочные системы	Системы управления базами данных
50-60-е	Встроенная документация, руководства пользователей	Текстовые редакторы
	Печатные инструкции для пользователя	Типографские машины

Детализация и активизация знаний

напр., АОС-ВУЗ. Средства представления и обработки знаний в этих языках ограничены жесткими сценариями взаимодействия человеко-машинного, управляемого компьютером (информационный кадр — вопрос/задание — анализ ответа обучаемого — поощрение или помощь — следующий информационный кадр). В конце 70-х гг. благодаря оснащению компьютеров графическими дисплеями, средствами интерактивной машинной графики стало возможным широкое применение тренажеров и учебных игр на базе ЭВМ. Особенность таких тренажеров и игр — наличие как логической, так и аудиовизуальной модели изучаемого процесса, явления, ситуации,

что позволило реализовать более гибкое и эффективное обучающее взаимодействие человека и компьютера, предусматривающее, в частности, их обоюдную активность или *диалог*. В дальнейшем формировании знаний о компьютере примером могут служить *многофункциональные автоматизированные учебные курсы* (напр., АФРОДИТА), позволяющие работать как под управлением компьютера (в случае затруднений), так и самостоятельно, приобретая и закрепляя навыки общения с операционной системой, умения алгоритмизировать свои задачи и т.п. Поддержкой таких курсов стали более сложные инструментальные средства — *интегрированные пакеты прикладных программ*, напр., АОС-РРВ, АОС-ВУЗ/ПРИМУС.

В начале 80-х гг. появились *решатели* задач и экспертные системы, относящиеся к классу систем представления знаний. Большинство экспертных систем характеризуется возможностью интерпретировать или понимать сообщения пользователя на *естественном языке ограниченном*, решать нечётко сформулированные задачи определённого класса и объяснять полученные результаты. Для обеспечения возможности формирования знаний разрабатываются специальные инструментальные средства: “пустые” или инструментальные экспертные системы (ИНТЕР-ЭКСПЕРТ), системы логического программирования (ПРОЛОГ); настраиваемые естественно-языковые *лингвистические процессоры* и др. Решатели задач и экспертные системы начинают применять и в учебном процессе; при этом возникла необходимость в их дальнейшем развитии. С этой целью разработаны *экспертные обучающие системы* для обучения определённому предмету на основе знаний соответствующих экспертов.

Современная КТО должна обеспечить:

обучаемому - возможность свободного доступа к любым формам компьютерных знаний учебного назначения, организованным в виде *учебных сред*;

преподавателю - применять для достижения целей обучения на различных занятиях как отдельный вид машинных знаний учебного назначения, так и любой их набор, т.е. проектировать учебную среду;

автору или разработчику программ учебного назначения возможность эффективно использовать как отдельное программное средство, так и любое их сочетание.

КТО включает теоретические основы, освещающие психолого-педагогические, дидактические, санитарно-гигиенические и эргономические аспекты, философские аспекты компьютеризации обучения, речевое взаимодействие с компьютером, основы *технологии программирования* и т.д., а также программные и технические средства, автоматизирующие процессы обучения (см. *Теоретические основы компьютерной технологии обучения, Программное обес-*

печение компьютерной технологии обучения, Техническое обеспечение компьютерной технологии обучения).

Знания и методы др. научных дисциплин претерпевают в КТО специальную "учебную" трансформацию. Так, на основе понятия модульности, одного из базовых понятий в технологии программирования, разрабатываются учебно-технологические линии и модули, позволяющие с малыми затратами компоновать и собирать, напр., автоматизированные учебные курсы из заранее заготовленных частей-модулей. В качестве других примеров могут служить развитие проектирования психолого-педагогического на основе методов проектирования программного обеспечения, а также понятие педагогического программного продукта — полностью готового к применению программного изделия, способствующего решению конкретных педагогических задач. Создание экспериментального или промышленного педагогического программного продукта — новый вид массового творчества в сфере образования, который влечет за собой значительные затраты человеческих, финансовых и иных ресурсов. Так, при создании программ учебного назначения для предметов инженерной подготовки, связанных с проектированием, затраты на обеспечение 1 ч работы студента составляют от 500 до 1000 ч совместного труда преподавателя, психолога, художника, программиста. Такие затраты могут быть эффективными только при централизованной разработке учебных программ с последующим тиражированием и распространением их многими потребителям. Это обуславливает необходимость унификации технических и программных средств КТО. С этой целью в системе образования многих стран в качестве технических средств приняты персональные компьютеры типа IBM PC или совместимые с ними машины, а единство программной среды обеспечивается разработкой унифицированных и согласованных между собой инструментальных средств.

О том, как средства КТО способствуют достижению целей информатизации образования, свидетельствуют следующие факты. Создание и повсеместное распространение средств поддержки обучения (справочных текстовых и графических материалов, компьютеризованных учебников и лекций, информационно-справочных систем учебного назначения, имитационных моделей) на базе ПЭВМ, локальных и распределенных сетей сможет обеспечить доступность знаний и данных для каждого члена общества и возможность непрерывного образования и повышения квалификации. Тренажеры, экспертные обучающие системы составят основу для индивидуализированного опережающего обучения и развития творческих и профессиональных способностей обучаемых различных категорий. Экспертные системы, решатели задач, учебные игры с многими участниками дадут возможность формировать навыки сотрудничества, обмена, кооперации. Оснащение программных продуктов и перспективных информационных технологий автоматизированными учеб-

ными курсами обеспечит путь ко всеобщей компьютерной грамотности, к реализации непрерывного повышения профессионального уровня всех членов общества. Согласно экспериментальным данным применение АОС, программ контроля знаний в 1,5-2 раза сокращает время и повышает качество усвоения многих предметов по сравнению с традиционной формой обучения. При этом резко возрастает активность обучаемых, количество решаемых ими *учебных задач*. КТО освобождает преподавателя от изложения и отработки с обучаемым значительной части учебного материала. Высвободившееся время преподаватель использует для дополнительной индивидуальной работы с отстающими или для постановки новых, более сложных и интересных заданий успевающим. Такое видоизменение работы педагога улучшает познавательную деятельность всей учебной группы в целом, усиливает творческие компоненты труда преподавателя. Интенсивная работа обучаемого и преподавателя в условиях информационного комфорта, обеспечиваемого средствами КТО, приводит к гармоничному комплексному достижению всех целей информатизации образования.

Обычно выделяют три проблемы КТО. Первая проблема относится к этапу проектирования деятельности обучаемых в условиях реализации различных дидактических функций (учение, обучение и др.). Одно из решений этой проблемы, предлагаемое в данном словаре, состоит в разработке теоретического аппарата и приёмов его применения для представления человеко-машинных алгоритмов совместного решения прикладных и учебных задач обучаемым и программно-технической системой. Вторая проблема относится к этапу реализации (программирования) спроектированных алгоритмов. Решение ее состоит в создании семейства взаимосвязанных легко переносимых или мобильных программных сред. Третья проблема связана с этапом эксплуатации средств КТО. Для ее реализации необходимо организовать их тиражирование и внедрение. Один из ключевых вопросов — опережающая подготовка и повышение квалификации профессорско-преподавательского состава в области КТО, создание сети автоматизированных учебных центров по КТО и новым информационным технологиям.

Компьютеризация и информатизация образования требуют разгрома и повышения профессионального мастерства преподавателей, поскольку: увеличивается объем учебной нагрузки на практических, лабораторных и др. занятиях, относящихся к активным формам учебной деятельности; *организация самостоятельной работы студентов* становится одним из основных видов деятельности преподавателя; интеграция научной и учебной работы требует от преподавателя дополнительных усилий, связанных с реализацией межпредметных связей; широкое информационное обеспечение учебного процесса на основе регулярного пополнения прикладных программ учебного назначения, *баз данных и баз знаний* учебного назначения

повлечет за собой значительные трудозатраты на постоянное обновление и модернизацию учебного материала.

Общее развитие КТО идёт по пути усложнения и обогащения её средств *видеокомпьютерными системами*, интерактивной графикой, системами анализа/синтеза устной речи. Другим перспективным направлением развития КТО является все более широкое применение методов и средств искусственного интеллекта для поддержки эффективного обучающего диалога, конструирования учебных сред, совместного решения с обучаемым учебных и практических задач с целью подготовки к работе в информатизированном обществе, где первостепенным становится владение не самими знаниями, а умение найти и усвоить информацию о них, чётко формулировать задачу и находить алгоритм её оптимального решения, а также находить партнеров и сотрудничать с ними в быстром решении класса задач.

Применение средств искусственного интеллекта в деятельности преподавателей и администрации учебного заведения позволит эффективно решать дидактические задачи как для отдельного обучаемого, так и для целых коллективов с помощью экспертных систем-"советчиков", экспертных обучающих систем и др. Первоочередная задача здесь состоит в быстрой компоновке автоматизированных учебных курсов, отвечающих творческим запросам преподавателя.

Для авторов автоматизированных учебных курсов и учебно-технологических модулей применение средств искусственного интеллекта предполагает генерацию учебно-технологических модулей по их спецификации на ограниченном естественном языке; автоматизацию изготовления различных форм знаний учебного назначения, т.е. переработку и активизацию учебных текстов и материалов, включая *машинный перевод* с естественных языков, применяемых для представления знаний учебного назначения.

Изложенные выше концепция и проблематика КТО в той или иной степени являются общими для ряда стран Восточной Европы, специалисты которых принимали участие в написании данного словаря. Вместе с тем конкретная реализация компьютерного обучения имеет в каждой стране свои специфические особенности, что нашло отражение в кратком обзоре (см. Приложение).

Лит.: 1. Batanov D., Dovgiallo A. A technological approach toward creating computerized learning and teaching environment. // 30 Conference Proceedings of the Association for the Development of Computer-Based instructional Systems (ADCIS). Belling Washington, 1988.

2. Белов В.Н., Брановицкий В.И., Довгялло А.М. и др. ПРОЛОГ — язык логического программирования // Прикладная информатика. —1986.— Вып.1.

3. Grizenko V., Dovgiallo A. Role of Computerized Teaching Technologie in Informatization of Education Preprints of the Third International Conference // Children in the information Age. —1989.— V.2.

4. Гриценко В.И., Довгялло О.М. [та ін.]. Навчальна інформатика: засоби і завдання // Радянська школа. —1989. —N 2.
5. Дановски П., Довгялло А.М., Кирова К.Н. и др. Автоматизированные обучающие системы на базе СПОК // Современная высшая школа. —1983. —N1.
6. Dubrenil V. Training of teacher educators in the use of microcomputers in secondary education on the basis of French and English experiments. Paris, 1982.
7. Диалоговые системы. Современное состояние и перспективы развития. / Под ред. А.М.Довгялло Киев, 1987.
8. Ibragimov O., Petrushin V. Expert—Tutoring Systems: Concepts and Examples. Preprints of the Third International Conference // Children in the Information Age. —1989. —V.2.
9. Лаутербах Р., Фрей К. Программное обеспечение процесса обучения // Перспективы. — 1988. —N 3.
10. Мазак Э., Холи К. Руководство индивидуальной работой студентов вузов и проблемы его исследования // Современная высшая школа —1984. —N4 (48).
11. Машбиц Е.И. Психолого-педагогические проблемы компьютеризации обучения. М., 1988.
12. Михалевич В.С., Гриценко В.И., Довгялло А.М. Основные направления автоматизированного обучения кадров в области кибернетики, информатики и вычислительной техники // Управляющие системы и машины. —1985. —N6.
13. Михалевич В.С., Савельев А.Я., Довгялло А.М., Когдов Н.М. Экспертно-обучающие системы в комплексе компьютерных средств обучения // Современная высшая школа —1988. —N1.
14. Петрелла Р. Человек в меняющемся мире // Газ. За рубежом. —1988. —N2.
15. Психолого-педагогические основы использования ЭВМ в вузовском обучении. / Под. ред. А.В.Петровского, Н.Н.Нечаева. М., 1987.
16. Савельев А.Я., Новиков В.И., Лобанов Ю.И., Подготовка информации для автоматизированных обучающих систем. М., 1986.
17. Стефик М., Эйкинс Я., Балзер Р. и др. Организация экспертных систем // Кибернетический сборник —1985. —N2.

*В.И. Гриценко, А.М. Довгялло,
Н.М.Когдов, А.Я. Савельева.*

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — теоретические знания и приемы их использования, обеспечивающие достижение целей *компьютерной технологии обучения*.

Разработка КТО междисциплинарная проблема, решение которой требует привлечения различных областей научных знаний. Среди них важное место занимают задачный подход и теория *задач*, положенные в основу формализации целого ряда ключевых понятий, используемых при проектировании программно-педагогических средств (в т.ч. и таких традиционных понятий педагогической психологии и дидактики, как, напр., *метод обучения*). Теория задач выступает промежуточным звеном между естественно-математическими, техническими и гуманитарными научными исследованиями, обеспечивающими успешную разработку и внедрение КТО в учебно-воспитательный процесс. На всех этапах создания *программ учебного назначения*, при выборе и разработке оптимальных спецификаций, способов *проектирования*, средств *кодирования* (или *программирования*) и режимов эксплуатации необходимо учитывать различные факторы, влияющие на принятие решения. Среди них важную роль играют факторы, относящиеся к дидактическим, психолого-педагогическим, санитарно-гигиеническим, эргономическим и другим аспектам КТО.

А.М.Довгялло, Н.М.Когдов, А.Я.Савельев.

ЗАДАЧНЫЙ ПОДХОД К КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — направление в проектировании взаимодействия *человеко-машинного* и *решения задач* с помощью компьютера. Подразумевает прежде всего разработку теоретического аппарата для представления *человеко-машинных алгоритмов* совместного решения задач пользователем и компьютером. Разработка таких алгоритмов и сценариев взаимодействия требует специальных средств формализации, в значительной мере независимых как от конкретного содержания *предметных областей*, на которые ориентируют создаваемые системы, так и от технических и программных средств, на которых будут реализованы эти системы. В качестве теоретического аппарата задачного подхода все чаще начинают применять так наз. общую теорию задач (разработана в Институте кибернетики им. В.М.Глушкова АН Украины и Институте психологии Украины). Это обосновывается следующими аргументами: задачи и процессы их решения изучаются в математике и логике, психологии и физиологии, педагогике и социологии, кибернетике и науковедении и т.д., а также на стыках наук (разработка систем *искусственного интеллекта* и *человеко-машинных систем*, обучение с помощью вычислительных машин и т.п.); существенные расхождения в трактовке понятия задачи и описании процессов решения задач представителями разных наук и научных школ затрудняют организацию их творческого сотрудничества в разработке сложных и актуальных проблем; объекты, называемые задачами, обладают достаточно выраженной спецификой по сравнению с другими объектами.

Сущность задачного подхода состоит в выделении в каждой рассматриваемой ситуации систем (людей, коллективов, машин), решающих задачи, в определении типов этих задач и указании средств их решения. Речь идет, по сути, о конкретизации системного подхода применительно к ситуациям решения задач с тем уточнением, что понятие задачи должно трактоваться достаточно широко. Задачный подход подтвердил свою продуктивность как в психолого-педагогической сфере (при разработке систем учебных заданий и

учебного материала в целом), так и в информатике. Применительно к КТО этот подход используется для создания достаточно сложных программных средств, в той или иной мере способных к решению задач и к *диалогу*, для построения *автоматизированных учебных курсов* по различным дисциплинам и для проектирования учебной деятельности пользователя, вступающего во взаимодействие с компьютером.

Методической предпосылкой к разработке общей теории задач послужила идея выделить “задачные” *модели* основных структурных единиц взаимодействия пользователя (обучаемого и компьютера) и представить каждый конкретный сценарий обучения, диалога, игры и др. как процесс обработки этих “задачных” моделей двумя решающими системами — человеком и компьютером. В число указанных моделей входят модели *познавательной задачи, задачи обучения, задачи программирования, задачи установления понимания, планирования* и др.

Г.А.Балл, А.М.Довгялло.

ДИДАКТИЧЕСКИЕ АСПЕКТЫ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — теоретические средства (напр., знания о наиболее эффективных методах обучения) и приемы их использования (конкретизирующиеся в методиках обучения), которые обеспечивают решение дидактических задач при разработке и внедрении КТО. Самые современные персональные электронные вычислительные машины с развитым математическим обеспечением не могут обеспечить эффективную реализацию КТО, если при составлении обучающей программы не используются рациональные приемы отбора содержания обучения по темам и разделам конкретного учебного материала, не учитываются требования принципов дидактики и оптимальных методик обучения, воспитательной и развивающей функций компьютерного обучения. Весьма существенны проблемы определения места компьютера в системе средств обучения, содержания деятельности и функций преподавателя в условиях компьютеризации. Значительный интерес представляют дидактический анализ информационной культуры, специфические особенности компьютеризированного обучения. К важнейшим дидактическим вопросам относится также проблема организации урока с использованием компьютера.

Дидактические и психолого-педагогические аспекты во многих случаях связаны с рассмотрением одних и тех же проблем, однако при этом выделяются различные линии анализа.

Е. Д. Маргулис.

ПСИХОЛОГО-ПЕДАГОГИЧЕСКИЕ АСПЕКТЫ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — теоретические средства и приемы их использования, основанные на принципах *педагогической психологии* и обеспечивающие проектирование, реализацию (*программирование*) и применение эффективных методов, содержания и организационных форм *компьютерного обучения*. При анализе этих аспектов предусматривается различная направленность программных и технических средств КТО, обеспечивающих прежде всего обучающую или учебную деятельность, определенные этапы процесса решения учебных задач (ориентировочный, исполнительный или контрольный), реализацию на этих этапах тех или иных поддерживающих функций.

В вычислительной технике и информатике чаще всего используется эмпирически сложившаяся система обозначений различных средств КТО, непосредственно связанная с их функциями (напр., *информационно-справочная система, экспертная обучающая система*). Аналогично и при классификации *программ учебного назначения* (ПУН) применяются преимущественно эмпирически сложившиеся классификации, акцентирующие те или иные особенности обучающей или учебной деятельности. Так, по одной распространенной классификации различают ПУН тренировочные, наставнические, проблемного обучения, имитационные, моделирующие и игровые. Иногда *обучающие программы* различают по дидактическим функциям, на реализацию которых они главным образом направлены (изложение нового материала или его иллюстрирование, моделирование и имитация реальных объектов и отношений между ними, контроль и закрепление знаний и т.д.), а также в зависимости от особенностей *технического обеспечения компьютерной технологии обучения и программного обеспечения компьютерной технологии обучения* (способы представления информации, обеспечения диалога, интерфейса с обучаемым и т.д.). Одной из альтернатив эмпирическому подходу в классификации является методическая концепция создания *педагогических программных продуктов*, другая альтер-

натива связана с классификацией обучающих программ по обеспечиваемому ими типу *обучения* (в основу различия типов обучения положена та часть способа действий по решению учебных задач, которая выступает в качестве прямого продукта обучения).

Актуальность психолого-педагогических аспектов КТО резко возросла в связи с *компьютеризацией* системы образования, отражающей объективные закономерности нового этапа НТР и социально-экономического развития общества. Исходные рубежи в этом направлении были намечены в нашей стране в начале 80-х гг., когда была осознана необходимость вооружения учащихся знаниями и навыками использования современной вычислительной техники, обеспечения широкого использования компьютера в учебном процессе, создания кабинетов информатики и вычислительной техники. Были определены меры по обеспечению *компьютерной грамотности* учащихся и широкого внедрения вычислительной техники в учебный процесс, предусматривалось введение в программы учебных заведений новой дисциплины "Основы информатики и вычислительной техники", а также проведение широкомасштабного эксперимента по использованию ЭВМ в преподавании традиционных учебных дисциплин начиная с младших классов общеобразовательной школы.

Одно лишь наличие компьютеров без научно обоснованного применения их не ведет к повышению эффективности обучения, к успешному овладению учащимися компьютерной грамотностью, к положительным сдвигам в сфере воспитания и развития личности учащихся. Отмечаются случаи недостаточно продуманного использования ЭВМ и иногда разочарование педагогов и учащихся в их возможностях. Поэтому говорить об эффективности и целесообразности применения компьютеров в обучении возможно лишь в случае повышения его качества.

Решение задач компьютеризации обучения во многом зависит от психолого-педагогического обеспечения, создания качественных учебно-методических пособий для преподавателей и учащихся, ведения интенсивной экспериментальной работы по изучению психолого-педагогических особенностей учебно-воспитательного процесса. Перед психолого-педагогической наукой выдвинут ряд серьезных проблем: разработка теоретических основ компьютерного обучения (возможно, достаточной окажется модификация уже имеющихся теоретических конструкций), определение его содержания, методов и организационных форм, обеспечивающих наиболее эффективное использование ЭВМ в учебном процессе, построение системы психологически обоснованных требований к компьютерам учебного назначения, их программно-методическому обеспечению и т.д. Все эти задачи должны решаться в тесной взаимосвязи с проблемами воспитания и развития личности обучаемого.

Психолого-педагогические проблемы компьютеризации обучения не должны рассматриваться в отрыве от социально-исторического контекста, от актуальных задач, которые поставлены перед наукой на новый этап НТР. Интенсивное развитие электронной вычислительной техники и широкое ее применение в различных сферах народного хозяйства привели к появлению новых областей психологической науки — *психологии программирования* и *психологии компьютеризации*. Предпосылками изучения этих проблем стали результаты психолого-педагогических исследований, посвященных вопросам обучения различных категорий пользователей ЭВМ программированию и *решению задач с помощью ЭВМ*. Можно выделить четыре основных направления использования компьютеров в системе образования: 1) компьютер как объект изучения; 2) компьютер как средство обучения; 3) компьютер как компонент системы управления образованием; 4) компьютер как средство научно-педагогических исследований.

Каждое из этих направлений отличается своими психолого-педагогическими аспектами. Наиболее важны аспекты, связанные с применением компьютера как средства обучения. По существу, термин “компьютерное обучение” отождествляется именно с рассмотрением компьютера как средства обучения. В то же время понятие “компьютеризация обучения” соотносится и с первым, и со вторым направлениями, т.е. с применением компьютера в качестве объекта изучения и средства обучения. Можно выделить несколько групп психолого-педагогических проблем, обусловленных задачами эффективного применения компьютера как средства обучения. Первая из них связана с концептуальными представлениями, анализом теоретико-методологических предпосылок решения указанных задач. Следует отметить, что необходимость учета психолого-педагогических аспектов осознается не всеми, нередко в основе обучающих программ лежат личный опыт преподавания, интуитивные представления и эвристические посылки, которые нередко оказываются малоэффективными (разработка обучающих программ и ныне является преимущественно прерогативой профессиональных математиков и программистов, не имеющих достаточной психолого-педагогической подготовки). Важно не только привлекать к составлению таких программ преподавателей-предметников, методистов, педагогов и психологов, но и вооружить их действенной теорией, т.е. необходимо усвоение не только уже известного в области компьютерного обучения, но и исследование фундаментальных проблем. Получение новых знаний, связанных со специфическими психологическими механизмами обучения, обучающих воздействий, структуры способа управления учебной деятельностью в условиях компьютеризации, необходимость эффективно использовать компьютер в учебном процессе привела к переосмыслению требований, предъявляемых к теории обучения и, в частности, к пересмотру тех ключевых понятий,

которые в ней используются. Установлено, что традиционные для педагогической психологии и дидактики представления даже о таких базовых понятиях, как метод обучения, обучающее воздействие, учебная задача и т.д. недостаточно технологичны, т.е. не дают возможности технологизировать заложенные в них теоретические предпосылки. По этой же причине существующие психолого-педагогические теории обучения не удовлетворительны, поэтому использование их для компьютерной реализации требует как существенных дополнений, так и выполнения ряда специфических требований.

Существуют две основные точки зрения относительно разработки психолого-педагогической теории компьютерного обучения. Согласно первой из них необходима разработка принципиально новой, отличной от уже известных, теории компьютерного обучения, согласно второй — возможна модификация и уточнение известных психолого-педагогических теорий обучения в направлении их технологизации. Для более детального анализа этих возможностей следует использовать известные теоретические конструкции (см. *Проблемное обучение*, *Программированное обучение* и др.). Вторая группа проблем связана с разработкой технологии компьютерного обучения т.е. с разработкой знаний и приемов, связывающих психологические и педагогические теории с практикой обучения и позволяющих эффективно использовать их при решении конкретных практических задач, возникающих в учебном процессе. Особенно важно добиться максимальной реализации тех специфических возможностей, которые предоставляет компьютер, прежде всего в области эффективного управления учебной деятельностью. Это, в частности, возможность строить “историю” обучения данного учащегося и целой группы (класса) учащихся, выяснять подлинные причины допущенных ошибок (а не ограничиваться их фиксацией и локализацией в процессе решения), разрабатывать *модель обучаемого* с учётом его индивидуальных особенностей и описывать на *языке программирования* высокого уровня обучающие воздействия, которые необходимо осуществить, дать управляющую программу, обеспечивающую выдачу таких воздействий в нужной последовательности, и т.д. Реализация этих возможностей требует определенного психолого-педагогического обеспечения, напр, уточнения концепций и понятий, описывающих процесс обучения, а также их технологизации с рассмотренной выше первой группой. В рамках указанной группы проблем возникло также немало специфических для компьютерного обучения задач, связанных, в частности, с необходимостью уточнения места и функций компьютера в учебном процессе (и соответственно роли преподавателя в новых условиях), выявления *оптимальных организационных форм компьютерного обучения*, сочетания компьютерных и некомпьютерных форм обучения, *индивидуализации обучения*, разработки *диалога* в обучающей системе и т.п. Проблемы

этой группы должны решаться в тесной связи с дидактическими, санитарно-гигиеническими и эргономическими и др. аспектами КТО.

Среди рассматриваемых психолого-педагогических аспектов КТО основным является обеспечение эффективного управления процессом обучения с помощью компьютера, тесно связанного, с одной стороны, с особенностями того или иного теоретического подхода к проблемам обучения, а с другой — опосредствующего технологию выбора и применения тех или иных *обучающих воздействий*. Кроме того, для эффективного управления процессом обучения необходимы выбор, разработка (генерирование) и применение обучающих воздействий на основе анализа динамической модели обучающего. Для выявления индивидуальных особенностей учащихся и особенностей их деятельности широко используются различные психодиагностические методики, в т.ч. и реализованные с помощью компьютеров (см. *Компьютерная психодиагностика*).

Поскольку в условиях компьютерного обучения его технология реализуется программами учебного назначения, выделяют третью группу проблем, связанную с проектированием таких программ. Проектирование рассматривается как составной компонент общей исследовательской стратегии, предусматривающей решение вопросов теории и технологии проектирования в комплексе с исследованием теории и технологии компьютерного обучения. Под проектированием подразумевается в данном случае *проектирование психолого-педагогическое* (в отличие, например, от проектирования инженерно-педагогического, социально-экономического), рассматриваемое как многоуровневый, иерархически организованный процесс, осуществляемый итеративным образом — апробация проекта влечет за собой его уточнение и доработку.

Выделенные три группы проблем нельзя рассматривать изолированно, поскольку от того, как решаются проблемы одной из них, существенно зависит решение проблем других групп (в частности, исходные концептуальные представления об учении и его психологических особенностях существенно влияют на разработку его технологии и проектирования). Кроме того, в условиях компьютерного обучения возникают качественно новые требования к решению основных проблем педагогической психологии и дидактики, а также принципиально новые, специфические для условий компьютеризации обучения, проблемы (см. *Психологическая готовность*). Основу компьютерного обучения составляет обучающая программа, которая осуществляет управление деятельностью учащихся, выполняя при этом некоторые функции учителя.

Ключевым вопросом применения компьютеров в целях педагогического управления является вопрос о том, какие функции могут быть возложены на компьютерную технику на разных уровнях педагогического управления. За исходную классификацию уровней управленческой деятельности с использованием ЭВМ может быть

принята следующая: 1-й уровень — управление обучением и развитием отдельного учащегося; 2-й — управление учебным процессом в рамках одного учебного заведения; 3-й — управление работой группы родственных учебных заведений; 4-й — управление работой учебных заведений, объединённых по территориальному признаку.

Последнее направление применения компьютеров в системе образования связано с научными, прежде всего с психолого-педагогическими исследованиями. Наиболее очевидна возможность оперативной обработки педагогической информации с помощью компьютеров — сбор, фиксация, обработка и хранение данных об учебной деятельности отдельных учащихся и функционировании групп учащихся. Ведутся исследования и практические разработки, позволяющие значительно расширить функции компьютерной техники в этом направлении, привлекая её в процессе отбора и конструирования содержания обучения, построения квалификационных характеристик, учебных планов и программ, систематизации и структурирования учебного материала в учебниках и учебных пособиях и т.д. Накоплен опыт разработки математического аппарата и алгоритмов составления учебных планов и программ, использования машинно-ориентированных сетевых методов совершенствования учебно-программной документации и учебного процесса, обоснования принципов структурирования и систематизации учебного материала с последующей формализацией соответствующих информационных моделей и т.д. Наиболее интенсивно компьютеры применяют при внедрении результатов теоретических опытно-экспериментальных разработок в массовую педагогическую практику, для информационного обслуживания педагогов-практиков (эта задача особенно актуальна в связи с организацией в массовой школе психологической службы), а также в целях педагогической прогностики.

Использование компьютера в исследовательских целях наиболее эффективно в том случае, когда компьютер сам выступает в качестве средства психолого-педагогического исследования компьютерного обучения. Здесь предметом специального изучения становится информация о процессе учебной деятельности и эффективности обучающих воздействий, осуществляемых с помощью компьютера. В этом проявляется триединая функция психолого-педагогического исследования, включающая задачи собственно исследования, формирования и проектирования учебной деятельности. Иллюстрацией может служить психолого-педагогическое проектирование новых информационных технологий, в частности компьютерных учебно-игровых сред.

Анализируя психолого-педагогические аспекты КТО, следует учитывать социальные и экономические последствия информатизации общества, развития информационной инфраструктуры и компьютеризации большинства отраслей общественного производства, философские и социально-педагогические аспекты этих явлений. Целесо-

образно также рассмотреть роль и перспективы применения компьютеров для решения актуальных задач воспитания и развития личности учащихся. Этот аспект важен еще и потому, что нередко вычислительная техника в школе представляется исключительно средством решения математических задач и производства разнообразных вычислений. Поскольку компьютер может быть эффективно использован для развития восприятия, памяти, воображения и мышления учащихся различного возраста, способствовать решению задач воспитания и развития личности, целесообразно проводить интенсивные исследования в этом направлении и, возможно, создать научную дисциплину - педагогическую информатику. В ее контексте психология должна раскрывать законы умственной деятельности человека, педагогика - разрабатывать на их основе методы и принципы организации процесса обучения как главной детерминанты умственного развития, а информатика - интерпретировать эти законы, методы и принципы в виде программ, реализуемых средствами вычислительной техники. С их помощью преподаватель сможет управлять процессом усвоения знаний, формирования умений и навыков, умственного развития учащихся. Совершенно очевидно, что результаты, полученные в этом направлении исследований, станут важным и постоянно развивающимся компонентом компьютерной грамотности, а также существенной предпосылкой повышения эффективности применения компьютера в качестве средства обучения.

Лит.: 1. Велихов Е.П. Новая информационная технология в школе // Информатика и образование. —1986. —N 1.

2. Гершунский Б.С. Компьютеризация в сфере образования. Проблемы и перспективы. М., 1987.

3. Гриценко В.И., Довгялло А.М. Пути развития информатизации образования // Информатика и образование. —1989. —N 6.

4. Ершов А. Информатизация: от компьютерной грамотности учащихся к информационной культуре общества // Коммунист. —1988. —N 2.

5. Компьютер в обучении: психолого-педагогические проблемы (Круглый стол) // Вопросы психологии. —1986. —N6; —1987. —N1.

6. Ломов Б.Ф. Научно-технический прогресс и средства умственного развития человека // Психологический журнал. —1985. —N 6.

7. Машбиц Е.И. Психолого-педагогические проблемы компьютеризации обучения. М., 1988.

8. Машбиц Е.И., Бабенко Л.П., Верник Л.В. и др. Основы компьютерной грамотности. 1988.

9. Машбиц Е.И., Андриевская В.В., Комиссарова Е.Ю. Диалог в обучающей системе. Киев, 1989.

10. Психолого-педагогические основы использования ЭВМ в вузовском обучении. М., 1987.

11. Средства формирования учебно-игровых сред для младших школьников. Киев, 1988.

12. Талызина Н.Ф. Управление процессом усвоения знаний. М., 1984.

13. Тихомиров О.К., Бабанин Л.Н. ЭВМ и новые проблемы психологии. М., 1986.

Е. Д. Маргулис.

САНИТАРНО-ГИГИЕНИЧЕСКИЕ И ЭРГОНОМИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — комплекс исследовательских, программных и проектных решений по приведению технических и программных средств КТО в соответствие с эргономическими и гигиеническими нормами и требованиями, направленный на создание комфортных условий работы *обучаемых* и сохранение их здоровья. Включают эргономические требования к техническим и программным средствам деятельности пользователя, эргономические и гигиенические требования к *рабочим местам* и рабочим помещениям, гигиенические рекомендации по организации режимов труда и отдыха пользователей, проведение профилактических мероприятий и изучение заболеваемости пользователей.

Эргономические требования к техническим средствам (в первую очередь, к монитору и к *клавиатуре*) — это требования к *качеству изображения*, к конструкции корпуса монитора и клавиатуры и их цветовому оформлению (см. *Эргономика клавиатуры*). Эргономические требования к программным средствам КТО состоят из требований к дисплейным форматам (см. *Формат изображений*), ко времени ответа ЭВМ на различные запросы учащегося (см. *Время ответа компьютера*), к кодированию информации, предъявляемой на экране ЭВМ (см. *Ассоциативное кодирование*, *Трансформационное кодирование*, *Цветовое кодирование изображения*), к *формату сообщений*, к формированию экранных окон (см. *Многооконный интерфейс*), к воспринимаемости и читаемости информации на экране, к организации и содержанию информации, выводимой на экран ЭВМ (см. *Читабельность текста*). К санитарно-гигиеническим и эргономическим основам построения рабочих мест, выбора и подготовки рабочих помещений относятся прежде всего требования к специализированной компьютерной мебели, к организации рабочего пространства и *рабочей позы*. В состав *автоматизированного рабочего места* должны обязательно входить специализированные рабочий стол и рабочее сиденье. Эргономичная компьютерная мебель является важнейшей предпосылкой обеспечения комфортных условий

обучения. При использовании неспециализированной мебели, даже при наличии самого высококачественного компьютерного оборудования и обеспечении гигиенических требований к рабочей среде в помещениях, у обучаемых неизбежно возникает зрительный и мышечный дискомфорт. Поэтому размеры стола должны быть достаточными для размещения всего необходимого оборудования; высота его рабочей поверхности должна быть регулируемой и легко изменяться в положении сидя. Рабочее кресло обучаемого должно быть вращаемым и регулируемым по высоте, надлежащим образом поддерживать тело, способствовать равномерному распределению веса по поверхности сиденья и не допускать неправильной позы обучаемого. Важнейшей характеристикой организации работы обучаемого с ЭВМ является его рабочая поза, которую невозможно обеспечить, используя современную неспециализированную школьную мебель. Исключительное внимание поэтому должно уделяться разработке и применению в кабинетах, оснащенных вычислительной техникой, специализированной компьютерной мебели. Учебные кабинеты, в которых размещена вычислительная техника, должны быть оборудованы в соответствии с эргономическими и гигиеническими требованиями (см. *Освещённость, Микроклимат кабинета автоматизированного обучения*). Особого внимания требует ориентация, площадь и интерьер помещения. При этом следует учитывать, что вопросы организации и оборудования помещения должны решаться в тесной связи с особенностями дизайна компьютеров. Так, цвет стен кабинета должен подбираться с учётом цвета корпусов монитора и клавиатуры. Рабочие места в кабинете также должны размещаться в соответствии с требованиями эргономики — исходя из площади, приходящейся на одно рабочее место, на проходы, а также с учётом исключения бликов и отражений на экране от источников света и исключения повышенной переадаптации зрительной системы учащихся. Особое внимание уделяется влажности воздуха в кабинете, степени содержания ионов и аэрозольных примесей в воздухе. Система вентиляции помещения, уборка помещений позволяют улучшить микроклимат в помещении с ЭВМ.

Важнейшей составляющей санитарно-гигиенических основ КТО является организация и регламентация режимов работы обучаемых с компьютером (см. *Режим работы с видеотерминалом*). При регламентации режима работы необходимо учитывать возраст обучаемых, особенности их деятельности на ЭВМ и др. (см. *Психолого-педагогические аспекты компьютерной технологии обучения*).

Для диагностики и изучения психофизиологических особенностей обучения с помощью ЭВМ разработан целый ряд методик, направленных как на изучение психофизиологических состояний пользователей, особенностей различных психических процессов, так и на диагностику и изучение различных жалоб и нарушений здоровья пользователей, связанных с работой за видеотерминалом. Работа с

ЭВМ требует значительного напряжения зрения вследствие того, что изображение на экране недостаточно высокого качества (см. *Качество изображения*), неустойчиво (см. *Устойчивость изображения*) и высококонтрастное (см. *Контраст*). Всё это является причиной возникновения *утомления зрительного* пользователей, а также различных нарушений зрительных функций при работе с ЭВМ (см. *Астиопия, Миопия*). К неблагоприятным факторам относятся также неравномерность распределения яркости между экраном, клавиатурой и документом, рабочими поверхностями и поверхностями оборудованной (см. *Блескость*). Чтобы обеспечить правильное фокусирование на относительно расплывчатых символах, необходимо непрерывное действие хрусталика глаза, поэтому так важна четкость и хорошая читаемость изображения на экране. Важное значение в определении зрительной нагрузки обучаемого при работе с экраном имеет содержание учебных заданий.

Комиссия Всемирной Организации Здравоохранения выделила три основных направления исследований зрительных нарушений, связанных с работой за ЭВМ: астиопия; функциональные изменения в зрительном анализаторе; патологические изменения зрительной функции. Функциональные изменения в зрительном анализаторе изучались в полевых и лабораторных условиях. У пользователей видеотерминалов обнаружены расстройства в функциях *аккомодации, конвергенции*, снижение остроты зрения и др. У обучаемых, работающих с ЭВМ в режиме ввода, наблюдается умственное утомление через 40 мин непрерывной работы. Одним из методов получения информации об умственном утомлении являются психологические тесты оценки показателей внимания. С их помощью измеряются распределение, переключение и объем внимания.

Большое внимание уделяется исследованию различных стрессовых состояний и нервно-психических расстройств, появляющихся в результате работы с ЭВМ (см. *Стресс*). Особое внимание следует уделять эргономическим особенностям временных параметров диалога, содержанию деятельности и заданий пользователя, снятию стнобразных монотонных заданий (см. *Монотония*), повышению *мотивации* работы с ЭВМ, *творческой деятельности* пользователя.

Для предотвращения утомляемости пользователей, профилактики нарушения их здоровья, а также повышения эффективности КТО необходимо: проведение эргономической и гигиенической оценки и специализированной экспертизы программных и технических средств КТО, рабочих мест и рабочих помещений, оборудованных компьютерами; эргономическое проектирование составляющих КТО, проведение ежегодных осмотров обучаемых — пользователей ЭВМ врачами (офтальмологами, невропатологами, ортопедами).

Лит.: 1. Бондаровская В.М., Мартиросова В.Г., Навакатилян А.О. Влияние работы с видеотерминалами на зрительную систему пользователей // Измерения, контроль, автоматизация. -1989. -N 3.

2. Бондаровская В.М., Поякель Н.И. Психология и программное обеспечение ЭВМ: теория и практика. Киев, 1988.

3. Visual Display Terminals and Worker's Health. 1987.

4. Ergonomic Aspects of Visual Display Terminals: London, 1980.

В.М.Бондаровская, А.С.Коваленко, Н.И.Поякель.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — проблемно-ориентированное программное обеспечение ЭВМ, позволяющее использовать ее для создания и исполнения программ учебного назначения (ПУН). Различают программное обеспечение КТО прикладное и инструментальное. Прикладное обеспечение состоит из программ учебного назначения, непосредственно доступных учащимся и выполняющих некоторую дидактическую функцию (ознакомление с новым материалом, контроль знаний и др.), и вспомогательных программ, обеспечивающих поддержку учебного процесса (напр., формирование учебных планов, составление расписания занятий, автоматизированная система управления учебным заведением, системой образования). Пользователи прикладного программного обеспечения — учащиеся, а также преподаватели, организующие учебный процесс. Инструментальное программное обеспечение состоит из программ, предназначенных для создания программ учебного назначения. Его пользователями являются специалисты или коллективы специалистов — авторы учебных материалов и программ.

Создание эффективной программы учебного назначения — сложный и трудоемкий процесс. В коллектив разработчиков программного обеспечения КТО входят специалисты: эксперт по обучаемому предмету; методист — специалист по теории обучения; психолог — специалист по диагностике ошибок обучаемого и психологии восприятия учебного материала; художник — специалист по эстетическому оформлению учебного материала; композитор — специалист по музыкальному и шумовому оформлению учебного материала; администратор — организатор работ. В целом структура коллектива разработчиков подобна структуре творческой группы по созданию кинофильмов с заменой операторской группы программистами. Процесс создания или жизненный цикл ПУН состоит из следующих этапов: анализ проблемы и принятие решения о разработке; концептуальное проектирование (спецификация); детализированное проектирование; реализация, включающая прог-

раммирование, отладку, апробацию и оптимизацию; внедрение в учебный процесс и эксплуатация. На каждом этапе производится разработка документации и проверка правильности полученных результатов. Так, на этапе анализа проблемы и принятия решения рассматривается вопрос целесообразности создания ПУН, автоматизирующей учебный процесс или часть его. Этап концептуального проектирования состоит в описании основных характеристик программы и ее места в учебном процессе. Результатом является документ, содержащий характеристики ПУН и предназначенный для анализа и оценки экспертом данной области знаний. На этапе концептуального проектирования, как правило, определяются: состав коллектива разработчиков; возрастной и количественный состав предполагаемого контингента обучаемых; цели ПУН; требования к предварительной подготовке обучаемых; способ использования программы при обучении данному предмету; способы использования в программе методик обучения, методов представления данных и знаний. Концептуальный проект ПУН рецензируется и, в случае необходимости, дорабатывается. Детализированное проектирование состоит в нисходящей поуровневой разработке структуры ПУН в виде сценария, включающего информационный (текстовые и графические кадры, музыкальные фразы и т.п.) и управляющий компоненты. Процесс как концептуального, так и детализированного проектирования существенно зависит от типа разрабатываемой программы.

Реализация ПУН заключается в выборе необходимых инструментальных средств, *программировании, кодировании* и отладке программы в соответствии с детализированным проектом. Разработанная программа предъявляется специалистам для оценки ее качества. Апробация ПУН состоит в опытной эксплуатации программы на некотором контингенте обучаемых, сборе и обработке полученных результатов и, если необходимо, изменении сценария и программы. С получением удовлетворительных результатов программа внедряется в учебный процесс. Каждому этапу жизненного цикла ПУН соответствуют свои инструментальные средства. Наиболее гибкая их классификация имеет вид матрицы, столбцами которой служат этапы разработки, а строками — классы ПУН. Каждое инструментальное средство попадает в некоторую клетку или занимает несколько клеток матрицы. Чем оно универсальнее, тем большее число клеток покрывает. Инструментальными средствами концептуального проектирования служат формы или бланки определенного вида, которые предъявляются пользователю в печатном виде (напр., формы *Zeitgeist*) или в виде специализированного редактора текстовой информации (напр., авторская система *КОНЦЕПТ*). К инструментальным средствам автоматизации детализированного проектирования относятся специального вида бланки и формы (*Zeitgeist*, *МАКРО*), *авторские языки программирования* высокого уровня (напр., *ЛЯОК*)

и инструментальные программы, получившие название *авторских систем (АСТРА, AUTOOL, ADAPT, LDS, WISE)*. Развитые авторские системы имеют выход в *системы программирования*. На этапе реализации программы используются универсальные или авторские языки *Курсрайтер, Язык описания курсов, Тьютор, ABC-1*, система программирования *ТЕНКОР*, а также библиотеки программ и модулей.

Лит.: 1. Довгялло А.М. [и др.]. Диалоговые системы. Современное состояние и перспективы развития. Киев, 1987.

2. Klass R. The TenCore language and authoring system for the IBM (personal computer.) // Journal of Computer Based Instruction. Summer. —1984, —V.11. —N 3.

3. Савченко А. В. [и др.]. Разработка автоматизированных обучающих систем на базе микроЭВМ в Московском институте электронной техники. М., 1987.

4. Лаутербах Р., Фрей К. Программное обеспечение процесса обучения // Перспективы вопросы образования. —1988. —N3.

В.А.Петрушин.

ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — совокупность технических средств, применяемых в *компьютерной технологии обучения*. Базовым техническим средством для реализации КТО является универсальная *электронная вычислительная машина*, которая поддерживает *диалог* обучаемого с обучающей системой, функционирующей в специально организованной *операционной среде*. Разнообразие форм обучения (индивидуальное, групповое), а также ряд других факторов определяют технические требования и возможные решения при выборе технических средств для *автоматизированного обучения*. Так, *персональная электронная вычислительная машина* (ПЭВМ) — совершенное вычислительное средство для создания систем, ориентированных на обеспечение индивидуальных форм обучения, комплексы ЭВМ, в т.ч. и ПЭВМ, системы коллективного пользования ресурсами высокопроизводительных ЭВМ — для организации групповых занятий, сети ЭВМ как средство коммуникации и распределения ресурсов — для индивидуального и группового обучения. Во всех формах обучения компьютер выступает в качестве средства для активизации и интенсификации индивидуальной или коллективной учебной деятельности различных категорий обучаемых, что достигается за счет управления процессом обучения с помощью ЭВМ, а также предоставления необходимого вычислительного и *информационного обеспечения*. На ранних этапах развития КТО базировалась на простых специализированных устройствах *программированного обучения* и контроля знаний с фиксированным набором вопросов — ответов. С развитием ЭВМ делались первые шаги в реализации принципов централизованного управления программируемым учебным процессом с помощью универсальной ЭВМ. *Рабочие места* обучаемых оборудовались пультами в виде индикаторных цифровых табло и наборов функциональных клавиш.

Совершенствование структур и базового (системного) *программного обеспечения* ЭВМ расширило возможности доступа пользователей к их ресурсам, а переход от пакетной обработки заданий к

диалоговым режимам оказал революционное влияние на развитие КТО. Таким образом, развитие аппаратно-программной и организационной частей ЭВМ вызывало совершенствование технологии обучения с применением *автоматизированных обучающих систем*. В *инструментальном техническом комплексе, комплексе учебной вычислительной техники (КУВТ), кабельной вычислительной сети, локальной сети учебного назначения, терминальной сети* учебного назначения, *системе коллективного пользования* и других раскрываются аппаратная часть КТО, организация взаимодействия подсистем и устройств, входящих в состав специализированных систем и комплексов. Новые возможности применения аппаратных и программных средств в обучении дала компьютеризация процесса профессиональной подготовки обучаемых в различных сферах деятельности. Интенсивные методы обучения реализуются на модельных установках, в которых *модели реальных объектов* соединены с ЭВМ и управляются ими, моделируя реальные условия функционирования. Созданы учебные комплексы (компьютерные лаборатории) для освоения профессиональных навыков у технологов и операторов гибких автоматизированных производств (см. *Учебный кабинет гибкого автоматизированного производства*), робототехнических систем, автоматизированных систем управления технологическими процессами и т.д. В некоторых учебных ПЭВМ состав аппаратных средств расширяется введением дополнительных устройств ввода — вывода информации и устройств сопряжения с ними (синтезаторов звука, видеомагнитофонов, видеомониторов, видеотерминалов, аналого-цифровых и цифро-аналоговых преобразователей для связи с аппаратурой, лабораторным оборудованием и т.п.). Эти аппаратные средства расширяют возможности КТО, способствуя более глубокому и более наглядному представлению знаний об изучаемом предмете. Особое место среди аппаратных средств занимают *видеокомпьютерные системы*, органически объединяющие возможности визуализации информации с целью представления учебного материала и видеointерактивного взаимодействия в процессе обучения.

Таким образом, выбор тех или иных технических средств для поддержки КТО представляет собой многокритериальную задачу, в которой должны учитываться цели и задачи обучения, форма обучения, наличие *автоматизированных учебных курсов* по предметным областям обучения, поддерживающие их автоматизированные обучающие системы в требуемой операционной среде, технические, экономические, эргономические и другие факторы. Проектирование компьютерных систем обучения в каждом конкретном случае является самостоятельной задачей и решается применительно к избранной предметной области.

В.Н.Никулин.

ТЕРМИНОЛОГИЧЕСКИЕ СТАТЬИ

ABC-1 (англ. Author Builds Course — автор строит курс) — авторский язык программирования, предназначенный для описания обучающе-контролирующих программ учебного назначения на микро-ЭВМ. Разработан в 1988 в Институте кибернетики им. В.М. Глушкова АН Украины. В основу языка положены проблемно-ориентированные конструкции авторского языка *Тьютор* и языка описания курсов, расширен с средствами производного вывода и сравнения по образцу. Алфавит языка включает 26 латинских заглавных букв, цифры, " (кавычки), /, запятую, точку, круглые и квадратные скобки, знаки арифм. операций: +, -, *, :, ^ (степень) и знаки операций отношения: <, ≤, =, ≥, >. Эти символы предназначены для записи операторов и переменных. В строковых константах разрешается использование любых допустимых для конкретной ЭВМ символов.

Данные ABC-1 включают числовые и строковые проблемные переменные, системные переменные, параметры. Проблемные переменные кодируются таким образом: m_n — числовые и W_n — строковые, где m, n — номера переменных, причем макс. значение m и n задаются во время компиляции (см. *Компилятор*). Системные переменные имеют следующие значения: TIME — текущее время дня; WTIME — время работы с начала сеанса; ATIME — время ответа на последний вопрос; DATE — текущая дата; ANSW — ответ обучаемого на последний вопрос; EXIT — содержит метку с которой будет выполняться программа после перезапуска; GOTO — содержит метку, на которую будет осуществлен переход, если в операторе BR, CL или HL по значению строковой переменной метка не обнаружена; ERROR — код ошибки после выполнения сравнения по образцу или после выполнения операторов AR либо LD. Параметры могут принимать значения 0 (ложь) или 1 (истина). Если параметр имеет значение "истина", указанное ниже действие выполняется, в противном случае — не выполняется: CRLF — перевод курсора на следующую строку после выполнения оператора вывода текста; SPACE — удаление пробелов из ответов обучаемого; LCASE —

перекодировать прописные буквы в строчные в ответе обучаемого; CAPS — перекодировать строчные буквы в прописные в ответе обучаемого; NHELP — не выполнять *запрос* о помощи; UNDEF — если в фрагменте отсутствует оператор UN — выдать сообщение “Ваш ответ неверен”; STEP — пошаговое вычисление выражения. Кроме того, в ABC-1 допускается использование арифм. выражений над числовыми переменными и интерпретация строк символов как числовых выражений. В этом языке реализованы пошаговый режим вычисления арифм. выражения, арифметика с плавающей точкой и структурные конструкции типа IF—THEN—ELSE и DOWHILE — OD; расширены возможности сравнения по образцу; реализованы три модели графического ввода—вывода (“графическое перо”, “графические примитивы”, вывод фоновое графического изображения). В ABC-1 помощь обучаемому назначается условно в зависимости от состояния программы; имеются средства генерации звука.

В.А.Петрушин, В.А.Третьяк.

ABC-2 — *авторский язык программирования для проектирования экспертных обучающих систем на основе фреймового представления знаний*. Создан в 1989 в Институте кибернетики им. В.М.Глушкова АН Украины как развитие языка ABC-1. Программа на ABC-2 состоит из шести разделов: *описания предметной области, описания решаемых задач, модели обучаемого, интерфейса с обучаемым, диагностики знаний обучаемого и управления процессом обучения*.

Раздел описания ПО предназначен для задания структуры и свойств ПО, являющейся предметом изучения. Состоит из совокупности *фреймов*, описывающих отдельные понятия ПО и связи между ними. Фрейм описания понятия содержит имя понятия, список имен базовых понятий, список имен смежных понятий, список описаний атрибутов, список свойств понятия, определение понятия, объяснение понятия, примеры понятия, иллюстрацию понятия, звуковое сопровождение. Имя понятия должно быть уникальным, т.е. раздел не может содержать два понятия с одинаковыми именами. Список имен базовых понятий включает имена понятий, входящие в определение данного понятия, и служит для построения встроенных *стратегий* обучения. В список имен смежных понятий входят имена понятий, связь с которыми интерпретируется *автором*. Список описаний атрибутов необходим для формального представления понятий, а в списке свойств понятия задаются отношения между атрибутами в виде системы продукционных правил (см. *Продукционные методы представления знаний*). Таким образом эти слоты формализуют осн. характеристики понятия. В остальных слотах находится текстовая, графическая и музыкально-шумовая дидактическая *информация*, представленная на соответствующих подязыках.

Раздел описания решаемых задач предназначен для задания условий и *способов решения* задач. Он состоит из совокупности фреймов, описывающих задачи, каждый из которых содержит информацию о связях с разделом описания ПО, др. задачами, а также формальную и неформальную постановку задачи с использованием систем продукционных правил и подязыков вывода *текста*, графики и музыкально-звуковых эффектов.

Раздел описания модели обучаемого предназначен для описания принимаемых во внимание характеристик обучаемого и его уровня *знаний*. Представляется фреймом: содержащим рабочие переменные и переменные, описывающие психо-физиологические характеристики обучаемого, отношение обучаемого к понятиям ПО и задачам, историю обучения.

Раздел описания интерфейса с обучаемым предназначен для создания *учебной среды*, через которую обучаемый и программа взаимодействуют друг с другом. Состоит из набора интерфейсных *процедур*, посредством которых визуализируются объекты программы и вводится информация от обучаемого. В языке имеется несколько уровней интерфейса. Интерфейс нулевого уровня составляют процедуры ввода-вывода для элементарных объектов: *символа*, строки и числа. Интерфейс первого уровня включает процедуры работы со сложными объектами: *меню*, формами, таблицами, текстами, гипертекстами и рисунками. Использование процедур нулевого и первого уровня в интерфейсных процедурах позволяет создавать пользовательские интерфейсы более высоких уровней. Для каждого слота понятия и задачи существует стандартный интерфейс, который можно переназначить на пользовательский или модифицировать, изменив значения системных переменных.

Раздел диагностики знаний обучаемого предназначен для описания метода диагностики знаний на основе симптомов, полученных в результате *анализа ответов обучаемого* в процессе решения задачи. Раздел состоит из диагностических процедур, которые могут взаимодействовать между собой на основе вложенности.

Раздел управления процессом обучения предназначен для описания *алгоритма* этого управления на основе состояния модели обучаемого исходя из ограничений имеющегося учебного материала. Состоит из описания метапонятий, операторов инициализации и цели. Метапонятия представляют собой процедуры управления процессом обучения, каждая из которых соответствует некоторому *учебному диалогу*, реализующему определенную педагогическую цель. Операторы инициализации осуществляют подготовку ПО и модели обучаемого для дальнейшей работы. В качестве цели указываются имя понятия, которое должно быть изучено, и список условий, отражающих требуемое состояние модели обучаемого.

В.А.Петрушин, В.А.Третьяк.

АВТОМАТИЗАЦИЯ ПЛАНИРОВАНИЯ ОБУЧЕНИЯ — применение компьютера для составления учебных планов, учебных программ и расписаний занятий. Первые автоматизированные системы для решения данных задач появились в начале 60-х г. Ныне существуют многочисленные системы планирования обучения, работающие в автоматическом или диалоговом режиме. При автоматической генерации система на входе получает требования и строит план, программу или расписание без участия человека, используя точные или эвристические методы. Точные методы применяются для решения задач о потоках в сетях, раскраски графов, целочисленного линейного программирования. Недосток точных методов — высокая трудоемкость. Осн. эвристические методы, не гарантирующие достижения решения: моделирование работы человека; так наз. “жадные” алгоритмы, выбирающие первые подходящие частные решения; упрощенные варианты точных методов. При работе в режиме диалога составитель вводит и редактирует варианты плана, программы или расписания, используя специализированный строковый или табличный редактор; система проверяет их соответствие общим и заранее определенным частным требованиям. Анализ учебных планов и программ производится с помощью моделей, использующих сетевые графики и так наз. сети Петри; при анализе расписаний применяются методы контроля целостности и непротиворечивости баз данных.

Современные системы А.п.о., ориентированные на персональные электронные вычислительные машины, включают: редактор для ввода и редактирования входных данных и фрагментов учебного плана или расписания, а также получения информации о расписании по запросам и оформления выходных форм; подсистему диагностики, позволяющую работать с совокупностью диагностических сообщений как с БД; подсистему автоматической генерации, основанную на точных и эвристических методах дискретной оптимизации с автоматизацией выбора методов, использованием диалога при решении оптимизационных задач, возможностью генерации, сохраняющей введенные или сгенерированные ранее фрагменты.

В. А. Бардадым.

АВТОМАТИЗИРОВАННАЯ ДИПЛОМНАЯ РАБОТА — дипломная работа, для выполнения которой автоматизируют некоторые виды деятельности с целью применения знаний, приобретенных в процессе обучения. Автоматизация дипломной работы охватывает автоматизацию вычислительной деятельности, автоматизацию информационной-справочной деятельности, автоматизированную подготовку документации и графической деятельности по черчению. При автоматизации вычислительной деятельности используют программные системы вычислений, применимые в реальной практике в процессе конструирования, проектирования, оптимизации, создания нового изделия

или исследования определенных характеристик. Направление “Информационно-справочная деятельность” охватывает процессы, связанные с библиографическими справками, работой с каталогами, справочниками и т.д. Для этого создаются *информационно-справочные системы с базами данных* по учебным и научным направлениям. При этом экономится время *обучаемого*, и вместе с тем ему предоставляется полная *информация*, необходимая для решения проблем дипломной работы. Направление “Документация и графическая деятельность по черчению” представляют системы обработки *текста*, с помощью которых можно оформить дипломную работу и приложения к ней, системы графической деятельности по черчению при конструировании и проектировании, позволяющие получить графическую информацию с помощью компьютера.

И. Панков.

АВТОМАТИЗИРОВАННАЯ КОНТРОЛЬНАЯ РАБОТА — форма выявления и оценки степени овладения *обучаемыми* знаниями, умениями и навыками, контролирующая уровень усвоения определенной дозы учебного материала и организуемая *преподавателем* с помощью технических средств *компьютерной технологии обучения*. Может применяться для осуществления предварительного, текущего, рубежного, промежуточного и итогового контроля. Основу А.к.р. составляет система контрольных знаний (тестов). Перспективным направлением развития А.к.р. является создание генератором учебных контрольных заданий, которые позволяют автоматизировать процесс синтеза различных заданий. Их классифицируют по степени автоматизации формирования заданий, форме аппарата представления *знаний*, форме *ответа* на генерируемый *вопрос* и др.

При вводе ответа в А.к.р. *обучаемый* использует следующие методы ответа: выборочный — ввод элемента из ряда предложенных; числовой — *кодирование* ответа в виде числа; результативный — ввод результата расчётов; конструируемый ввод совокупности *символов* в определенной последовательности; графический — представление ответа фигурой, графом или структурой; естественный метод, предполагающий использование грамматики языка, применяемого в системе.

Формирование оценки за выполнение контрольных заданий может проводиться с учетом числа допущенных ошибок, отношения числа ошибок и правильных ответов, характера допущенных ошибок и их весовых коэффициентов. Результаты выполнения А.к.р. отображаются на экране и/или документально. В итоговый документ (табл.) включаются наименование курса и А.к.р., фамилия *обучаемого*, номер варианта контрольных заданий, число правильных и неправильных ответов и оценка. Во избежание конфликтных ситуаций при несогласии *обучаемого* с оценкой в итоговый документ включаются

Выходной документ выполнения автоматизированной контрольной работы в МФ АУК "АИДА".

Кишиневский государственный университет им. В.И.Ленина
МФ АУК АИДА"

Режим конструирования и описания алгоритмов

Студент: Лемперт Р.М.

Протокол выполнения контрольных заданий. Вариант 3

Вопрос	Ответ на вопрос	Оценка
1.	1,4	Правильно
2.	1 0 (N+1): N M 2 8 3	Правильно
3.	3,4,6,7,8	Правильно
4.1	MOD-4,NAME-10,BOX-6	Правильно
4.2	NOMER-2,BOUND-12,SOL-4	Правильно
4.3	NUMER-4,ADRESS-40,FAMIL-15	Правильно
5.1	MON-0004,ZARA-00000003FFFFFFFFE	Неправильно
5.2	PAK-324D324D324D,OXE-00000025	Правильно
5.3	АВАК-0000001200000012,ZUR-00003C	Правильно
5.4	MORA-FFF4,RUB-C2C2C2C2C2C2	Правильно
6.	1,2,4,5	Правильно
7.	FEBB	Правильно
8.	P'+0000017'	Правильно
9.	F'5'	Правильно
10.	S1 D1 M1 I1 C1(T1,I1) T1 P1 E1	Правильно
11.	6	

Результат выполнения контрольной работы

	Количество вопросов	Количество правильных ответов	Количество не- правильных ответов
Сведения об алгоритмах	2	2	0
Лексические конструкции и операторы			
Операторное описание алгоритма	12	11	1
	2	2	0
Всего	16	15	1

его ответы на контрольные задания. При выполнении нескольких А.к.р. в составе *автоматизированного учебного курса* для конечного и промежуточного контроля результаты могут накапливаться в банке данных для их последующей обработки.

А.Ф.Лысков.

АВТОМАТИЗИРОВАННАЯ КУРСОВАЯ РАБОТА — курсовая работа, выполненная с использованием компьютерной техники для автоматизации вычислительных процессов. Дает возможность многократно сократить время проведения трудоёмких, рутинных процессов, освобождая внимание *обучаемого* для выбора оптимального решения из множества возможных вариантов. В процессе разработки А.к.р. *обучаемый* применяет вычислительные программы с конкретными вводными *данными*, анализирует полученные результаты и применяет их (при необходимости несколько раз) при решении своей проблемы. Иногда использование готовых программ для автоматизации вычислительных процессов может привести к их механическому применению без осмысления методики. Поэтому обычно проводятся предварительные тесты для определения уровня подготовки *обучаемого* и при удовлетворительных результатах он допускается к работе с программой.

И.Панков.

АВТОМАТИЗИРОВАННАЯ ЛЕКЦИЯ — систематическое последовательное изложение проблемных ситуаций разделов конкретной науки с использованием видеокomпьютерной техники для демонстрации рисунков, графиков, подвижных изображений и др. Проводится в аудитории, оснащенной *автоматизированными рабочими местами* *обучаемых*, соединенными между собой в *локальную сеть учебного назначения*. Центральный компьютер, управляющий работой компьютеров *обучаемых*, находится у лектора.

На ранних этапах автоматизации учебного процесса были предприняты попытки автоматизации лекции с помощью оперативного контроля *знаний* *обучаемых*, рабочие места которых оснащались простейшими кнопочными *устройствами ввода-вывода информации* типа “да”, “нет”, “не знаю”. В дальнейшем центр тяжести автоматизации лекции переместился с функции контроля *знаний* на функцию *преъявления информации* с помощью компьютера.

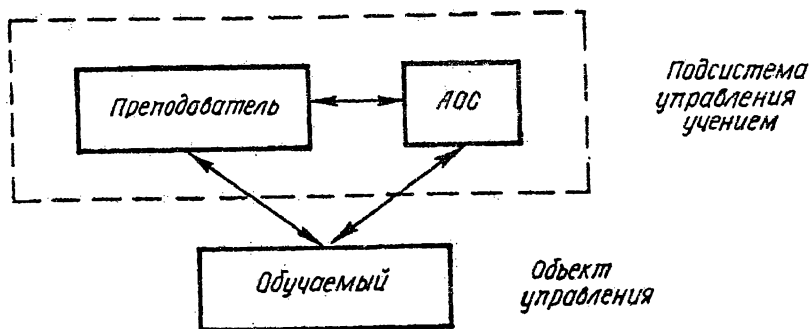
А.л. проходит в два этапа. На первом этапе лектор устно излагает один из разделов учебного материала; при этом на компьютеры *обучаемых* выводится сопроводительный учебный материал, иллюстрации, графики и т.п. Лектор задает темп изложения материала, одинаковый для всей аудитории. На втором этапе на экраны *обучаемых* выводится план лекции, осн. понятия, связи и логические структуры материала. *Обучаемый* может задать вопрос лектору в ус той форме либо использовать свой компьютер для передачи сообщения/вопроса на центральный компьютер лектора. Диалоговая форма взаимодействия активизирует внимание и позволяет избежать монотонности изложения материала.

Задачи А.л. — сформировать у *обучаемых* при помощи компьютера системное представление об изучаемой дисциплине;

указать основные, наиболее существенные моменты изучаемой темы и побудить обучаемых к дальнейшему самостоятельному изучению вопроса. В основе А.л. лежит план лекции — чёткая формулировка темы и цели раздела, записи наиболее сложных выкладок, заранее намеченный *алгоритм* развития темы, логика перехода от одного вопроса к другому.

В.И.Васильев, А.М.Довгялло.

АВТОМАТИЗИРОВАННАЯ ОБУЧАЮЩАЯ СИСТЕМА (АОС) — комплекс технических средств, программного и учебно-методического обеспечения для активизации и интенсификации индивидуальной и/или коллективной познавательной деятельности различных категорий *обучаемых* на основе реализуемого компьютером управления данной деятельностью и предоставления при этом необходимых информационных и вычислительных ресурсов. Ориентирована на решение двух классов *задач*. Первый, основной, класс составляют *задачи обучения*. В основу функционирования таких систем в рамках учебного процесса положено представление об *обучении* как о процессе решения задачи управления *учением* в человеко-машинной системе (рис.). Программные и информационные компоненты,



непосредственно связанные с решением задач обучения, наз. обучающими. Ко второму классу относятся задачи разработки обучающих компонентов. Возможность решения задач этого класса предполагает наличие в составе системы средств, обеспечивающих эффективную разработку обучающих компонентов *пользователями*, не имеющими, как правило, профессиональной подготовки в области *программирования*. Программные и информационные компоненты, ориентированные на решение этих задач, наз. инструментальными.

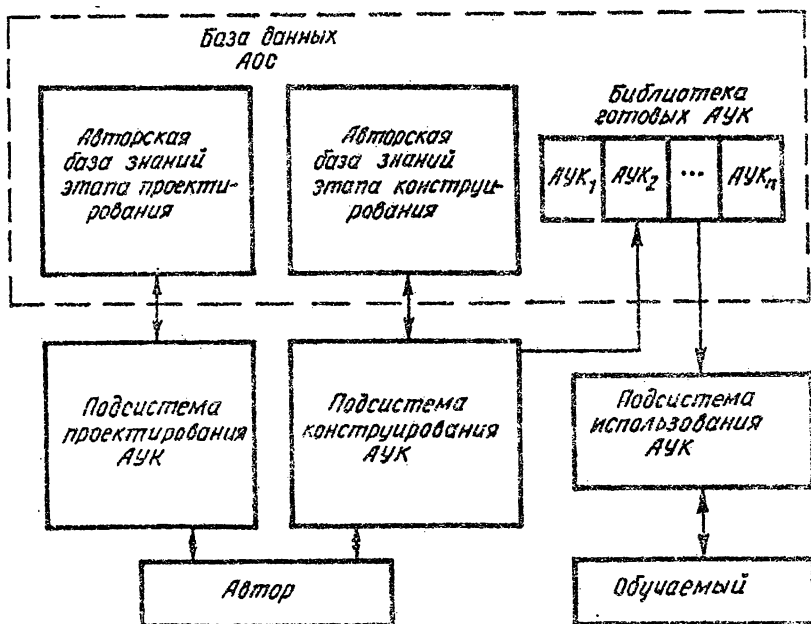
Принципиальное отличие *обучающей системы* от любой другой системы (технической, управляющей, информационной) состоит в

том, что наличие различных структурных элементов системы само по себе не обеспечивает эффективного достижения результата. Необходима активность управляемого субъекта. Обучаемый, являясь потребителем информации (учебной, научной и т.п.), в процессе обучения создает новую информацию в виде ответов на вопросы, вопросов и требований помощи. Управление обучением состоит из двух взаимосвязанных процессов: организации деятельности обучаемого и контроля деятельности. Эти процессы непрерывно взаимодействуют: результат контроля влияет на содержание управляющих воздействий, т.е. на дальнейшую организацию деятельности. В свою очередь, организация определенной деятельности требует определенной формы контроля и конкретного способа регистрации этой деятельности. Другой метод управления заключается в такой организации деятельности, когда каждый шаг к поставленной цели контролируется, но осуществляется самостоятельно, без управляющих воздействий извне. Пока преждевременно говорить об абсолютном преимуществе того или другого метода управления, т.к. эффективность управления зависит от многих факторов. Тем не менее, практика показывает, что более реалистичными являются различные сочетания этих двух основных методов.

Использование АОС непосредственно для обучения связано с выполнением компьютером следующих функций: управление учебной деятельностью; хранение и выдача учебной информации; моделирование лабораторных экспериментов, явлений, ситуаций, закономерностей и т.д.; анализ сообщений и ответов обучаемых; регистрация, хранение и обработка результатов учебной деятельности обучаемых. Принципиально важно при создании АОС обеспечение совместной реализации указанных функций. Управление учебной деятельностью требует двустороннего обмена информацией. Характерные виды обучающего диалога — совместное решение учебных задач, тренировка на машинной модели к.-л. объекта или процесса, учебные игры и др. Большинство АОС собирает и обрабатывает статистические данные о ходе обучения и, при необходимости, предьявляет их обучаемому либо преподавателю.

Обучаемый может работать с АОС как в условиях класса, так и индивидуально. Современная АОС на базе системы коллективного использования состоит из следующих функциональных подсистем: библиотеки автоматизированных учебных курсов (АУК) — совокупности прикладных программ управления обучением, контроля знаний и др.; исполняющей системы, обеспечивающей выполнение АУК; архива — пс системы, осуществляющей сбор данных о процессе обучения каждого обучаемого по соответствующим прикладным программам и предьявляющей эти данные преподавателю; подсистемы сервиса, облегчающей администратору базы данных или диспетчеру АОС регистрацию АУК и пользователей АОС, получение печатных данных о функционировании системы и др.; подсистемы общения и

управления, обеспечивающей взаимодействие между различными компонентами АОС. В современных системах индивидуального обучения (рис.) выделяются две осн. подсистемы. Первая реализует



задачи проектирования учебных курсов, в частности определения и корректировки информационных порций материала; определения и корректировки обучающих программ; определения и корректировки эталонов ответов для проверки правильности ответов обучаемых; формирования базы данных для поддержки вопросно-ответного режима работы системы. Вторая подсистема предназначена для непосредственного ведения процесса обучения и решает следующие задачи: регистрацию обучаемых; интерпретацию обучающих программ; поддержку вопросно-от. тного режима работы; проверку метода решения задачи обучаемым; выдачу статистической информации о ходе процесса обучения. Кроме основных, есть и вспомогательная подсистема, обеспечивающая функционирование осн. подсистем. Она решает задачи редактирования текстов, графических изображений, отображения информации на дисплей, обмена информацией с дисками и т.д. База данных АОС содержит учебные модули, составляющие АУК, список обучаемых, статистическую информацию

о ходе процесса обучения и др. информационные и программные компоненты.

В процессе разработки АОС пройден путь от частных реализаций на малых и средних компьютерах с двумя-тремя обучаемыми до сетей компьютеров учебного назначения, наиболее характерный представитель которых — система PLATO-IV фирмы CDC, работающая на базе сети CYBERNET из 15 больших компьютеров. С развитием *персональных электронных вычислительных машин* наблюдается процесс децентрализации АОС до уровня *локальных сетей учебного назначения*. Разработкой средств, методов, устройств и программ для АОС занимаются главным образом фирмы по производству вычислительной техники — прежде всего IBM, Hewlett-Packard, DEC, CDC, а также университеты. Наиболее характерные системы: MALT — для обучения *машинно-ориентированным языкам*; COACH — ориентированная на обучение выполнению простых арифм. операций; SCHOLAR — для обучения географии. На базе отечественных АОС реализовано несколько весьма эффективных обучающих программ, напр., по языкам ПЛ/1 (в системе КОНТАКТ) и Фортран (в системе СПОК-ВУЗ), по химии (в системе АТОС).

Применение АОС в 1,5 — 2 раза сокращает время и повышает качество усвоения многих предметов по сравнению с традиционной формой обучения. При этом резко возрастает активность обучаемых, количество решаемых ими учебных задач. Интенсивная самостоятельная работа обучаемых в условиях информационного комфорта обеспечивает более высокую эффективность обучения с применением АОС.

И.И.Мархель, В.И.Отенко.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА НАУЧНЫХ ИССЛЕДОВАНИЙ (АСНИ) — совокупность технических, программных средств, методического обеспечения и математических методов, необходимых для проведения научного эксперимента, обработки и анализа полученных результатов.

В автоматизации научных исследований, связанных с проведением эксперимента, выделяют два уровня в единой системе автоматизации: *объектный*, где осн. требования к АСНИ задаются объектом исследований, и *инструментальный*, на котором определяющими являются теор. база *данной предметной области*, информационная база и средства *обработки информации*. На первом уровне в АСНИ должны входить вычислительные средства для сложных видов обработки и накопления информации и управления средствами отображения. К ним относятся компьютеры типа СМ-3, СМ-4 и другие компьютеры *системы малых ЭВМ*. Второй уровень составляют локальные системы автоматизации экспериментов — *объектные*

терминалы, имеющие выход через стандартный интерфейс на объекты исследований и строящиеся на базе микро-ЭВМ Электроника-60 и др., а также системы КАМАК. В современных системах вычислительные машины не только обрабатывают информацию, но и координируют работу всей системы, реализуя взаимодействие исследователя и системы. Другим важным элементом АСНИ является электронный интерфейс, обеспечивающий совместную работу различной аппаратуры, прежде всего взаимодействие экспериментальной установки и компьютера. Наибольшее распространение при создании АСНИ получили средства сопряжения, выполненные с использованием принципов программной управляемости, модульности (агрегируемости), магистральности, машинезависимости и междунациональных стандартов на электронный интерфейс (АСТБАС, КАМАК, МАЛТИБАС и др., ориентированные на создание устройств сопряжения функциональных модулей; МЭК-625.1 низкоскоростной приборный интерфейс).

В зависимости от использования средств вычислительной техники в АСНИ выделяют два уровня: нижний, или уровень объектных компьютеров, непосредственно связанных с объектом исследования; верхний, или уровень инструментальных компьютеров, предназначенных для реализации сложных видов обработки, накопления больших массивов информации, организации банков данных и баз данных. Инструментальные компьютеры в рамках АСНИ коллективного пользования являются по существу информационно-вычислительным ресурсом системы в целом, а также объектом интеграции АСНИ в более широкие системы и сети (напр., в вычислительную систему коллективного пользования). Магистрально-модульный принцип построения АСНИ позволяет формировать различные конфигурации систем в зависимости от сложности объекта и задач исследований.

Задачей базового программного обеспечения АСНИ является обеспечение универсальности применения компьютера, его высокой производительности, адаптируемости к возможным изменениям конфигурации аппаратных средств, к введению новых программ, реализующих те или иные возможности. В вычислительной системе можно использовать несколько различных режимов работы, отличающихся друг от друга организацией вычислительного процесса, способом обмена информацией с объектами исследований, принципами взаимодействия исследователя с компьютером. Операционная система должна быть достаточно развитой, чтобы можно было реализовать несколько различных режимов. Требования, предъявляемые к ОС компьютеров верхнего и нижнего уровней АСНИ, должны быть различны: для компьютеров нижнего уровня, где основным является режим реального времени, в ОС включаются программные средства, обеспечивающие взаимодействие компьютера и экспериментальной установки (объекта исследования); для компьютеров верхнего

уровня ОС должны включать программы, обеспечивающие создание и обработку БД, машинную графику, взаимодействие компьютеров различного уровня. Прикладное программное обеспечение АСНИ чаще всего подразделяют на следующие части: пакеты прикладных программ общего назначения; пакеты сервисных программ технического обслуживания; пакеты специализированных программ; проблемно-ориентированные программы. Пакеты программ общего назначения включают набор программ, предназначенных для решения наиболее распространенных задач матем. характера: решения систем линейных алгебраических и дифференциальных уравнений, вычисление элементарных функций, статистический анализ и др. Пакеты сервисных программ технического обслуживания представляют собой совокупность тестовых и диагностических программ.

Общей тенденцией развития АСНИ, связанной с использованием средств вычислительной техники, является повышение производительности и пользовательских возможностей объектных машин. Это достигается увеличением объема их оперативных запоминающих устройств, оснащением микро-ЭВМ компактными и ёмкими накопителями на магнитных дисках, кассетными накопителями на магнитных лентах. Развитие средств технического обеспечения АСНИ осуществляется в направлении миниатюризации вычислительных систем при сохранении архитектуры и программного обеспечения. Эволюция АСНИ идёт в направлении децентрализации традиционных функций компьютеров всех уровней, когда задачи многоцелевой обработки информации будут реализовываться на интерфейсном уровне путём применения микро-ЭВМ и микропроцессоров, встроенных в интерфейсные блоки.

В методическое обеспечение АСНИ, как правило, включают матем. методы и алгоритмы, образующие теор. базу реализации осн. функций АСНИ (экспериментирование, сбор, обработка, представление, интерпретация информации и принятие решений), а также методы и алгоритмы анализа и синтеза (проектирование) АСНИ. Осн. элементы методического (теоретического) обеспечения АСНИ ориентированы на решение типовых задач планирования эксперимента, обработки экспериментальных данных, интерпретации результатов. Матем. обеспечение АСНИ представляет собой совокупность матем. методов и алгоритмов, реализующих эксперимент, накопление, обработку, анализ и интерпретацию результатов эксперимента, а также методы и алгоритмы принятия решений.

Н.М.Когдов.

АВТОМАТИЗИРОВАННАЯ ТРЕНИРУЮЩАЯ СИСТЕМА (АТС) — специализированная автоматизированная обучающая система, в которой обучение производится в форме тренировочных занятий по закреплению знаний, умений и навыков в процессе

самостоятельной работы *обучаемых*. Автоматизация процесса тренировки предполагает наличие в таких системах подсистем автоматизированной оценки правильности действий (*ответов*) тренируемых на основе их сравнения по к.-л. критериям с оптимальными в некотором смысле решениями, автоматизированной оценки уровня подготовленности тренируемых по некоторой *предметной области* на основе принятой *модели обучаемого*, автоматизированной выдачи новых заданий в соответствии с заданной моделью обучения, автоматизированной регистрации процесса тренировки и т.п. В АТС дидактический материал разделяют по степени сложности. В ходе тренировочных занятий система автоматически адаптируется к каждому обучаемому по сложности выдаваемых заданий в зависимости от достигнутого им уровня подготовленности. Степень сложности задания определяется в соответствии с выбранными критериями; напр., она может зависеть от количества базовых понятий, умений и навыков, необходимых для выполнения задания, а также от его трудоемкости. Процесс обучения в общем случае происходит следующим образом. Вначале тренируемому выдается некоторое количество заданий увеличивающейся сложности, с тем чтобы определить начальный уровень его подготовленности и сформировать модель обучаемого. Затем в соответствии с этой моделью, производится целенаправленное управление тренировкой до достижения заданных на данный сеанс обучения параметров модели обучаемого или исчерпания лимита времени. Результаты тренировки фиксируются, следующий сеанс тренировки начинается с достигнутого уровня.

АТС имеют широкий спектр применения. В системах, предназначенных для выработки навыков по принятию управленческих решений, используются учебно-методические материалы и *программное обеспечение* деловых игр. С помощью *деловых игр* решаются, в частности, задачи по распределению выделенных предприятию ресурсов, прибыли, по выбору производственной программы и др. Имеются АТС, в которых обеспечиваются тренировки персонала по обслуживанию сложных технических систем (тепловые и атомные электростанции, химическое производство), управлению движущимися объектами (космическими кораблями, самолетами, судами, автотранспортом). Такие АТС разрабатываются и используются в специальных центрах по подготовке специалистов гражданских и военных профилей. Так, за рубежом весьма популярны программы для персональных компьютеров "Полётный тренажёр", "F-19", "Абрамс", для которых не требуется специального дополнительного оборудования. Первоначально они создавались как видеоигры, однако оказались настолько удачными, что их стали использовать при профессиональной подготовке. Напр., графические возможности программы "Полётный тренажёр" позволяют показывать на экране дисплея панель управления самолета "Цессна-182" и панораму

аэропорта. Показания приборов на панели управления и вид набегающей взлетно-посадочной полосы меняются в полном соответствии с командами, которые подает с помощью джойстика "пилот", получающий практический опыт управления самолетом в различных ситуациях. Применение специального оборудования (электронных планшетов, имитаторов пультов управления техническими системами пр.) повышает эффективность АТС и превращает их в *интегрированные обучающие тренирующие системы*. В условиях вышших и средних специальных заведений использование тренировочных режимов с игровыми компонентами в АОС позволяет обучаться играя, делает тренировку увлекательной, обеспечивает хороший педагогический эффект.

С.Т.Адрианов, В.М.Зеленин.

АВТОМАТИЗИРОВАННОЕ ЛАБОРАТОРНОЕ ЗАНЯТИЕ — вид учебной деятельности, связанный с выполнением исследований на лабораторном оборудовании с применением компьютера. Включает проверку *знаний* обучаемого, необходимых для выполнения работы (допуск к лабораторным занятиям на ПЭВМ); разработку *обучаемым* плана проведения лабораторного эксперимента; выполнение исследований на лабораторном оборудовании, сбор и обработку данных эксперимента на компьютере, подготовку и распечатку отсчета на ПЭВМ; интерпретацию полученных результатов и выводов.

Один из вариантов полной автоматизации лабораторного упражнения — создание компьютерной *модели* изучаемого процесса, явления. Компьютер дает возможность провести эксперимент, моделируя действия на реальной опытной установке. Обучаемый задает начальные условия и отсчитывает результаты при помощи *устройств ввода—вывода информации*. Следовательно, за один урок можно провести множество наблюдений и подвести итог результатов, которые сокращают время для технической подготовки и для реальной опытной постановки. Полученный результат представляют в табличном и арифм. виде, а также на бумажном *носителе информации* при окончательном оформлении протокола с результатами.

Частичная автоматизация лабораторного занятия достигается при работе с реальной опытной постановкой, причем в компьютерную *программу* вручную вводятся полученные и отсчитанные измерительной аппаратурой результаты. Программа в этом случае совершает вычисления и дает результат в арифм. и табличном виде. В некоторых случаях лабораторное занятие можно автоматизировать при помощи прямого введения и отсчета компьютером исходных результатов. Для этой цели в опытную установку необходимо вмонтировать устройства для автоматизированной слежки за изменением величин, их преобразования из аналогового вида в цифровой, для введения в компьютер и последующей обработки данных. Возможен и комбини-

рованный вариант, где компьютер моделирует входные воздействия на реальный объект, набирая, обрабатывая и выводя полученные исходные величины. Использование компьютера в А. л.з. повышает степень закрепления знаний обучаемых, прививает необходимые практические навыки использования компьютеров в лабораторном эксперименте.

В.И.Васильев, И.Панков.

АВТОМАТИЗИРОВАННОЕ ОБУЧЕНИЕ — обучение, при котором управление познавательной деятельностью осуществляется с помощью компьютера, использующего *модель процесса обучения*, сформированную преподавателем-автором обычно в виде обучающе-контролирующей программы. На ранних этапах развития *вычислительной техники* А.о. осуществлялось с помощью обучающих машин. В 70-х гг. для реализации А.о. использовались в основном *автоматизированные обучающие системы* на базе систем *коллективного пользования*. Ныне А.о. — основа большого количества *программ учебного назначения*, реализованных на *персональных электронных вычислительных машинах* как автономно (в виде современных АОС с графическими возможностями, вопросно-ответным режимом и др.), так и в составе *диалоговых систем* и компьютерных обучающих сред. Осн. отличием системы управления познавательной деятельностью при А.о. от управления техническими системами является то, что управление познавательной деятельностью человека (объект управления) осуществляется опосредствованно, через его психическую деятельность. В соответствии с этим управление познавательной деятельностью можно определить как способ организации процесса обучения, обеспечивающий достижение дидактических целей, осн. из которых являются *знания, умения, навыки*.

Процесс управления техническими системами включает: формулирование задач управления, описание и анализ начального состояния объекта управления, выбор и определение характера управляющих воздействий на объект, способа реализации этих воздействий (исполнительные устройства), получение результатов управления (измерение), введение корректирующих управляющих воздействий. Если цель управления при однократном акте не реализуется, то процесс управления итерационным образом продолжается произвольное число раз. Все эти фазы можно выявить и в процессе *управления процессом обучения*, а следовательно, полностью или частично реализовать его с помощью АОС. Однако содержание их может существенно измениться. Если первой фазе соответствует задание целей обучения, то вторая фаза — определение начального состояния или, в данном случае, степени обученности — носит принципиально иной, а именно вероятностный характер с высокой

степенью неопределенности. Действительно, измерить степень обученности с помощью, напр., ранговых (балльных) оценок можно лишь приблизительно и с большим допуском. Проявление (измерение) степени обученности или, точнее, ее значения зависят от ряда факторов, в т.ч. субъективных. АОС не обладает интуицией преподавателя, который на основе опыта делает, как правило, статистически правильный вывод о степени обученности. С другой стороны, А.о. снимает некоторые субъективные факторы преподавателя, могущие повлиять на оценку.

А.Я.Савельев.

АВТОМАТИЗИРОВАННОЕ ОБУЧЕНИЕ ПРОИЗНОШЕНИЮ — обучение навыкам правильного синтезирования иноязычных звуков, слов и фраз на основе использования компьютерных средств. Успешное применение компьютерных средств в обучении звуковому строю неродного языка базируется на таких принципах традиционного обучения, как *системности принцип*, принцип коммуникативности, *наглядности принцип* и др. Однако компьютерные средства поднимают некоторые из них на качественно новую степень развития. Принцип коммуникативности воплощается в реальное диалоговое взаимодействие *обучаемого* с компьютером в устной форме. Компьютерная техника позволяет осуществлять процесс визуализации звучащей речи и обеспечивать эффективный контроль за постановкой правильного произношения.

Главная трудность в обучении произношению - влияние звуковой системы родного языка. Работа над развитием произносительных навыков начинается со звуков и слогов. На этом этапе необходимо использование аудиосредств (магнитофонов, лингафонной техники), позволяющих записывать и воспроизводить звучащую речь и заменяющих эти традиционные средства синтезаторов речи, при работе с которыми не затрачивается время на поиск нужной записи на магнитной ленте. Подключение магнитофонов и синтезаторов речи к компьютерам позволяет обучаемым не только тренироваться в их произнесении, имитируя те или иные звуки, но и самим контролировать свою работу. С помощью компьютера можно наблюдать на экране *дисплея* и делать спектральный анализ произносимых обучаемым звуков, определяя их интенсивность и длительность. Речевые сигналы можно также записывать в память компьютера, хранить их, вызывать и сравнивать. Важную роль играет зрительная наглядность. На экране дисплея можно не только демонстрировать рисунки и схемы, иллюстрирующие положение органов речи при произнесении звуков, но и графически моделировать движения этих органов, используя элементы мультипликации.

В университете г. Виктория (Канада) разработана фонетическая база данных на микро-ЭВМ, используемая для обучения иностранным

языкам. Она позволяет накапливать, хранить, осуществлять поиск, а также аудиторное и визуальное сравнение фонетических данных (звуков, звукосочетаний, слов). Для этого имеется аппаратно-программный комплекс, используемый совместно компьютерами IBM (РС,ХТ,АТ). БД охватывает несколько языков, в т. ч. редких, напр. турецкий, корейский (иногда трудно бывает найти преподавателя, знающего тот или иной язык; выход — *компьютерное обучение*). *Обучающая программа* содержит задания типа “найдите все слова, содержащие звук Х”, “найдите звуки, общие для следующих слов” и т.д. В США ведутся разработки с целью корректировки и улучшения произношения изучающих немецкий язык. Была создана программа для поиска и анализа ошибок восприятия, связанных с классификацией немецких гласных звуков. Слыша звуковые сигналы, обучаемые должны определить, каким звукам они соответствуют. Результаты контроля можно увидеть на экране или получить в напечатанном виде. Помимо количественных оценок (сколько звуков было воспринято правильно, а сколько нет), программа выдаёт объяснение ошибок и рекомендации по их исправлению. Разработанное фирмой “Тексас инструментс” портативное компьютерное устройство “Спик энд спелл” (“Говори и пиши”) предназначено для обучения на начальном этапе правильному произношению и умению писать слова английского языка. *Словарь* устройства насчитывает около 200 трудных с точки зрения орфографии слов. Эти слова были отобраны с учётом частоты их встречаемости в языке и разделены на несколько списков по возрастанию сложности. Устройство может работать в нескольких режимах. В одном из них обучаемый, слушая слова, произносимые синтезатором речи, должен записывать их тексты с помощью *клавиатуры*. При ошибке предлагается вторая попытка. Если и на этот раз человек неправильно записывает слово, то устройство само показывает правильное написание этого слова и произносит его по буквам. Закончив работу над одним списком и получив оценку, обучаемый переходит к более сложному списку слов. В другом режиме, видя сначала текст слова на маленьком экране, обучаемый должен произнести его. Затем это же слово произносит синтезатор. Таким образом осуществляет необходимая для постановки правильного произношения и накопления словарного запаса тренировка. При этом приобретаются навыки не только произнесения звуков, но и расстановки ударений в словах. Однако не менее важны навыки, связанные с темпом речи и интонацией. Использование компьютеров в этой области только начинается.

Современная речевая технология позволяет обучаемым не только слушать синтезированную речь, но и самим разговаривать с компьютером. В Институте кибернетики им.В.М.Глушкова АН Украины разработана многоязычная система речевого диалога, которую можно использовать для обучения устной речи на иностранном языке. Режим распознавания речи предусматривает

настройку системы на голос обучаемого и на ограниченный словарь одного из шести языков (русского, украинского, английского, французского, немецкого, испанского и итальянского). При этом на экране персонального компьютера последовательно предъявляются слова или фразы, произносимые которые в микрофон, человек настраивает систему. По желанию обучаемого система может "подсказывать" правильное произношение слов, синтезируя их на основе имеющихся в памяти фонетических транскрипций. Результаты настройки можно хранить в памяти системы, и при следующем сеансе взаимодействия с ней обучаемому достаточно лишь назвать своё имя. Словари, используемые в системе, могут соответствовать традиционно изучаемым разговорным темам, напр., "почта", "транспорт", "театр" и т.п. Режим синтеза речи не зависит от объема словаря; предусмотрено звучивание произвольных слов и фраз на любом из семи языков, вводимых с клавиатуры дисплея в виде транскрипций. В памяти системы содержатся учебные тексты на каждом из семи языков, озвучивание которых иллюстрирует произношение отдельных слов и фраз. В частности, представлены осн. типы интонации, и при синтезе текстов демонстрируются различные способы их выражения на разных языках. Темп синтезированной речи можно изменять автоматически. В начале курса компьютер может говорить медленно, затем темп речи постепенно приближается к естественному.

Т.В.Людовик.

АВТОМАТИЗИРОВАННОЕ ПРАКТИЧЕСКОЕ ЗАНЯТИЕ — вид учебной деятельности, связанный с приобретением практических навыков в соответствующей области знаний с использованием компьютера. Строится на сочетании традиционных и компьютерных форм обучения и контроля знаний и ориентировано на решение задач, обеспечивающих преемственность между практическими, лабораторными и лекционными занятиями на основе внутри- и междисциплинарных логических связей.

Различают две формы А.п.к.: построенное по методу "вопрос—ответ" и использующее коллективное решение проблемы в форме деловой игры. Первая форма обеспечивает действенный контроль знаний, умений и навыков обучаемых. Вторая форма максимально способствует развитию самостоятельного мышления и умению отстаивать свою точку зрения при решении научно-технических задач. Диалоговая форма работы группы стимулирует обучаемых к активному участию в коллективном мышлении, способствует приведению разрозненных знаний в определенную систему. В конце А.п.к. проводится контроль знаний по результатам решения индивидуальных заданий.

В.И.Васильев.

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО (АРМ) — *рабочее место*, оснащённое специальными техническими средствами для автоматизации определенной деятельности. Различ. эт АРМ преподавателя, конструктора, проектировщика, технолога и др. Основу АРМ составляет компьютер вместе с базовым программным обеспечением: операционной системой, трансляторами, пакетами программ машинной графики. Комплекс специального программного обеспечения в виде инструментальных средств и пакетов прикладных программ для решения задач конкретной предметной области определяет специализацию АРМ. Широко известны АРМ, в которых используются компьютеры системы малых ЭВМ (СМ-4, СМ-1420). АРМ преподавателя представляет собой комплекс программных и технических средств обучения для подготовки и ведения занятий в кабинете автоматизированного обучения. Такие кабинеты должны быть оснащены достаточным количеством рабочих мест для обучаемых, однако организация кабинета и оснащение одного рабочего места могут быть выполнены по-разному — на базе универсальных компьютеров, группой отдельных микро-ЭВМ или локальной сетью учебного назначения.

Первые кабинеты создавались на базе терминальной сети мощных универсальных компьютеров. В этом случае АРМ — отдельный терминал, управляемый системой коллективного пользования, средствами которой пользуются авторы обучающих программ на этапе их создания, а также преподаватели и обучаемые на этапе использования. Широко применяемыми системами этого типа являются АОС-ВУЗ, АОС-ВУЗ/ПРИМУС и др., однако стоимость эксплуатации таких кабинетов весьма высока. Дисплейные классы могут создаваться и на базе мини-ЭВМ типа СМ с использованием, напр., системы АОС-ВУЗ/СМ. Во втором случае АРМ преподавателя и АРМ обучаемого реализуется в виде отдельных компьютеров, оснащенных внешней памятью ЭВМ и периферийными устройствами. Поскольку между компьютерами нет связи, затрудняется подготовка к занятию и контроль процесса обучения. В третьем случае используются кабинеты, получившие название комплексов учебной вычислительной техники. В них имеется рабочее место преподавателя и несколько рабочих мест обучаемых, объединенных локальной информационной сетью. АРМ преподавателя — это ведущий компьютер сети, ресурсы которой обычно доступны для всех компьютеров сети (рабочих мест обучаемых). В состав программного обеспечения входят инструментальные средства разработки обучающих программ, контроля за ходом занятий, оперативной помощи обучаемым и накопления данных о ходе работы.

Для подготовки программ учебного назначения используются различные инструментальные средства программирования или создания обучающих программ. Так, для простых программ программированного обучения, расчетных или моделирующих программ

необходимы эффективные трансляторы с языков программирования высокого уровня и соответствующие отладчики. Для разработки автоматизированных учебных курсов в состав программного обеспечения могут входить авторская система, система управления базами данных или т.п. Ядро программного обеспечения АРМ обучаемого составляют готовые к выполнению обучающие программы.

Р.Ковалюнос.

АВТОМАТИЗИРОВАННЫЙ ЗАЧЁТ - вид учебной деятельности, связанный с контролем с помощью компьютера знаний, умений и навыков обучаемых по определенной учебной дисциплине или ее разделу. Для А.з. применяются автоматизированные системы контроля качества обучения. В основе А.з. лежат контролирующая учебная программа, содержащая контрольные вопросы. Качество обучения оценивается в соответствии с выбранным критерием, задаваемым автором контролирующей программы. Такой критерий основан, как правило, на соотношении числа верных и неверных ответов. В более развитых программах вычисляется рейтинг обучаемого и устанавливается личная траектория "движения" каждого обучаемого по конкретной учебной дисциплине. Цель А.з.

установление количественной меры знаний, умений и навыков каждого обучаемого в конкретной области. Эту меру можно выразить не только оценкой, но и системой различных коэффициентов, по совокупности которых преподаватель принимает решение о качестве знаний обучаемого.

В.И.Васильев.

АВТОМАТИЗИРОВАННЫЙ ОБУЧАЮЩИЙ КОМПЛЕКС (АОК) — совокупность двух и более технических средств компьютерной технологии обучения, способных поддерживать выполнение дидактических функций самостоятельно и объединенных с целью придания этим средствам возможностей, которыми они порознь не обладали. Выделяют три осн. группы автоматизированных обучающих комплексов: АОК, сочетающие процедуры компьютерного обучения и контроля знаний по различным дисциплинам с реализацией вычислительных операций и моделированием процессов в ходе автоматизации учебно-научного эксперимента; АОК, обеспечивающие формирование различных умений и навыков в процессе лабораторного практикума на базе полунатурных тренажеров с использованием компьютера, формирующего исходные задания и управляющего процессом по данным исполнительных устройств; АОК, базирующиеся на вычислительных сетях, управляющих комплексами групповых технических средств обучения, способных самостоятельно реализовывать учебные процедуры.

АОК применяются для автоматизированного контроля знаний, умений, навыков и социологических исследований больших коллективов *пользователей*, проведения лабораторных и практических занятий, контроля посещаемости, самоконтроля, обучения *преподавателей* работе с различными техническими средствами, проведения лекций с использованием телевизионной техники и аудиосистем. Функциональные особенности АОК предполагают активное участие *диспетчера* или преподавателя в организации и проведении учебного занятия.

В.Н.Сороко.

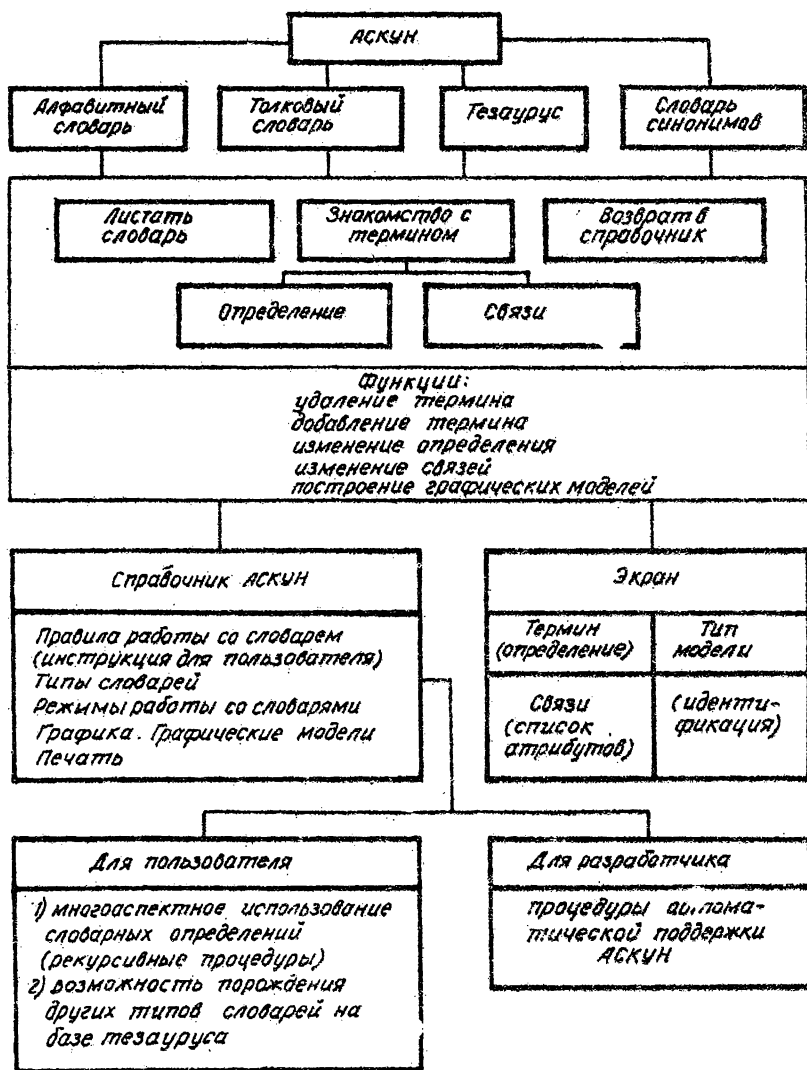
АВТОМАТИЗИРОВАННЫЙ СЕМИНАР — вид учебной деятельности, связанный с совершенствованием навыков и закреплении теоретических положений с помощью *автоматизированной обучающей системы*. Включает четыре этапа. На подготовительном этапе проверяется готовность *обучаемого* к занятию. На этом этапе в памяти обучаемых целесообразно актуализировать некоторые теоретические знания, для чего в автоматизированном режиме одна и та же информация может предоставляться параллельно на всех *дисплеях* одновременно всем обучаемым. Кадры должны следовать в строгой последовательности, а время выдержки кадра зависит от сложности предъявляемого материала. В другом варианте обучаемый может сам управлять скоростью выдачи кадров. Целесообразно разрешить обращаться к исходной установочной информации в течение всего А.с. Дополнительные возможности АОС на этом этапе выражаются в накоплении и анализе *данных* по всей предыстории обучения каждого обучаемого и всей группы. В конце первого этапа можно дать обучаемому информацию о степени его подготовленности к занятию. В *обучающей программе* есть возможность предусмотреть предъявление обучаемым подсказки или демонстрацию решения *задач*. На этапе постановки цели преподаватель формулирует осн. цель занятия. АОС может дать дополнительные теоретические сведения по теме занятия, привести примеры решения задач и упражнений, дать методические рекомендации. Этап текущей работы является главным в А.с. Существенным преимуществом *автоматизированного обучения* является то, что обучаемые могут получить различные задания путём случайной выборки заданий из *памяти ЭВМ*. В течение занятия обучаемые работают самостоятельно под управлением АОС. Преподаватель может получить оперативную информацию о ходе занятий, наблюдать за работой любого обучаемого, а при необходимости оказать индивидуальную помощь. На этапе анализа работы АОС может оценить успешность работы, выдать домашнее задание, дать методические установки на следующее занятие.

В.И.Васильев.

АВТОМАТИЗИРОВАННЫЙ СЛОВАРНЫЙ КОМПЛЕКС УЧЕБНОГО НАЗНАЧЕНИЯ (АСКУН) — словарная система, предназначенная для представления понятийной структуры изучаемой предметной области. Ориентирован на помощь обучаемым в освоении осн. терминологической лексики изучаемой ПО. Применение АСКУН позволяет формировать различные типы учебных моделей объектов и данных с привлечением экспертов по ПО, лингвистов, методистов, психологов, педагогов и др. специалистов, участвующих в создании компьютерной учебной среды, обеспечивает ассоциативное представление элементов знаний о данной ПО на основе многоаспектного использования различных типов словарной информации: толкового словаря, тезауруса, отдельных вспомогательных словарей, которые можно построить на базе исходных, а также графики. Реализуемая в среде АСКУН структуризация учебной деятельности осуществляется посредством согласования множества целей, стоящих перед обучаемым, с множеством отношений, описанных в тезаурусе. В системе имеются средства настройки на различные ПО.

Использование АСКУН обеспечивает: формирование вторичных словарей на базе тезауруса; комбинирование различных режимов работы, в результате чего пользователю предоставляется возможность всестороннего изучения некоторого понятия (объекта); создание адекватных для различных групп обучаемых учебных моделей объектов ПО. АСКУН можно использовать как самостоятельную систему или в качестве словарной подсистемы в составе соответствующего автоматизированного обучающего курса.

АСКУН состоит из трёх осн. словарей: терминологического (содержит алфавитный и толковый словари), тезауруса и словаря синонимов (рис.). Режимы работы включают: добавление, удаление терминов, их определений и ссылок на связь с др. терминами. Возможно также образование различных словарных подмножеств на основе исходных словарей. Терминологический словарь содержит список осн. подмножества терминов и понятий по ПО, а также их определения. Структура словарной статьи: <имя термина> — <определение>. Режимы работы со словарем: листать словарь (последовательный просмотр алфавитного словаря); знакомство с термином, включающее определение термина, связи данного термина с другими (обращение к тезаурусу) и печать терминологического словаря; возврат в справочник. Указанные режимы применимы к любому термину — как толкового словаря, так и тезауруса. Тезаурус представляет собой словарь семантических отношений между понятиями данной ПО. Осн. назначение тезауруса — представлять совокупность гл. понятий и систему предметно-логических связей между ними. Кроме того, автоматизированный тезаурус позволяет получить вторичные словари по различным словарным входам, а также генерировать особый тип семантических отношений между терминами (понятиями). ПО — цепочки рекурсивной связи по



некоторому отношению. Структура словарной статьи: <имя термина> — <перечень отношений>. Режимы работы со словарем: листать словарь (просмотр списка терминов и их отношений с др. терминами); знакомство с термином (определение термина и всех связанных с ним терминов); выдача словарной статьи тезауруса по

любому термину, связанному с данным); построение "обратного" словаря (формирование списка терминов по имени отношения); образование линейного словаря, являющегося подмножеством тезауруса; построение иерархического (рекурсивного) словаря — цепочки связи терминов по определенному отношению; возврат в справочник. АСКУН допускает работу с несколькими словарями. Для удобства каждому создаваемому словарю следует дать краткую аннотацию. Для рекурсивных словарей обязательно указать имя исходного термина и имя отношения, по которому строится словарь.

Т.Б.Андрусенко, А.Е.Стрижак, О.В.Шкода.

АВТОМАТИЗИРОВАННЫЙ УЧЕБНЫЙ КУРС (АУК) — компонент *автоматизированной обучающей системы*, предназначенный для решения *задач обучения* по данной *предметной области (ПО)* в человеко-машинной системе. Представляет собой интегрированную систему *учебных программных модулей (УПМ)*, снабженную учебно-методическими материалами, поясняющими и регламентирующими ее использование для достижения целей, определяемых задачами обучения, на которые ориентирован АУК. В состав учебно-методических материалов АУК могут входить учебные пособия по ПО, методические разработки и другие материалы, относящиеся к АУК в целом.

Современная система классификации АУК исходит из набора дидактических функций, реализация которых обеспечивается курсом. В зависимости от поддерживаемой дидактической функции АУК могут быть информационно-справочными, контролирующими, тренирующими, обучающими. Информационно-справочным считается АУК, реализующий сценарий, в соответствии с которым *обучаемый* обеспечивается информационным сервисом по данной ПО. Контролирующие АУК обеспечивают оценку степени усвоения *знаний* обучаемым. Тренирующие АУК предназначены для передачи умений и выработки навыков по выбранной разработчиками курса ПО. Обучающими являются те АУК, в основу которых положены сценарии, направленные на передачу *знаний* обучаемому. На практике АУК чаще всего сочетают несколько из указанных выше дидактических функций. Примерами *многофункциональных автоматизированных учебных курсов* могут служить АУК-ДМ, обеспечивающий обучение, контроль и тренировку при изучении дисциплины "Детали машин", и АУК "Афродита", обеспечивающий обучение *программированием* на языке *Фортран*.

Другую группу классификационных признаков составляют признаки, характеризующие возможности АУК по управлению *учением*, в частности свойства механизмов формирования учебной информации и адаптации к индивидуальным особенностям обучаемых. С учетом свойств этих механизмов рассматривают два больших

класса АУК: с фиксированными сценариями обучающего диалога и адаптивные. АУК с фиксированным сценарием осуществляет управление учением в соответствии с заранее разрабатываемыми спецификациями обучающего диалога. Именно эти спецификации и получили название сценария обучения. Сценарий обучения представляет собой описание последовательности и содержания актов взаимодействия обучаемого и АУК, которое по определению не может изменяться в процессе обучения. В связи с этим уровень гибкости управления познавательной деятельностью посредством АУК этого типа в общем случае определяется возможностями адаптационного механизма, предусматриваемого автором при разработке сценария обучения. Примером АУК с фиксированным сценарием обучения является АУК-ДМ. Особенность адаптивных АУК — использование некоторой модели предметной области, изучение которой предполагается. Существенными компонентами АУК этого класса являются также модель обучаемого и стратегия или план обучения. Управление учением осуществляется в соответствии с планом обучения, который можно автоматически скорректировать в зависимости от текущего состояния модели обучаемого и модели ПО. Механизмы формирования учебной информации и адаптации к индивидуальным особенностям обучаемых встроены в АУК и не требуют вмешательства со стороны автора курса.

Третья группа классификационных признаков представлена способом интеграции УПМ и по существу является классификацией возможных структур АУК. Способ интеграции УПМ определяется, как осуществляется взаимное кооперирование УПМ при решении конкретной задачи обучения. Простейший способ интеграции — независимая интеграция. Для получения доступа к обучающим функциям УПМ пользователь должен вызвать его. Вызванный модуль становится активным, в то время как остальные пассивны. Пользователь может вызвать любой УПМ, входящий в состав АУК. Чтобы обратиться к пассивному модулю, пользователь должен отключиться от активного или завершить с ним работу. Следовательно, пользователь, работая с активным модулем, не имеет возможности вызывать непосредственно из него пассивный УПМ. Существенное свойство независимой интеграции — автономность УПМ. Второй способ интеграции — подчиненная интеграция, основанная на существовании доминирующего УПМ. Все остальные модули зависят от доминирующего. Особенностью работы пользователя АУК этого типа является то, что для вызова любого из подчиненных УПМ необходимо прежде активизировать доминирующий, который выполняет функции осн. среды обучения и использования других УПМ. Подчиненные модули могут использоваться независимо друг от друга, но использование их независимо от доминирующего невозможно. Третий способ интеграции УПМ в составе АУК — синергическая интеграция, отличительная

особенность которой — возможность прямого взаимодействия между любой парой УПМ при отсутствии единого доминирующего модуля. В рамках АУК такого типа УПМ могут использоваться либо независимо друг от друга, либо через любой УПМ системы. Возможность прямого взаимодействия между УПМ способствует увеличению функциональной гибкости АУК и его адаптируемости к индивидуальным особенностям обучаемых.

Технология АУК в качестве основы использует передовые технологии программирования с привлечением идей из области методологии и психологии обучения, эргономики и др. дисциплин. Разработка АУК весьма трудоемка и осуществляется, как правило, коллективом, в состав которого должны входить преподаватель-предметник, методист и программист. В связи с этим одна из осн. задач развития технологии АУК — снижение трудовых затрат, необходимых для создания достаточно представительных АУК. Поскольку наиболее трудоемким этапом АУК является этап кодирования программ, то одна из целей развития технологии АУК состоит в устранении этого этапа. Это подтверждает и осн. линия эволюции технологии АУК: от использования традиционных языков программирования различного уровня и авторских языков программирования, специально ориентированных на создание диалоговых программ учебного назначения, к авторским и далее к экспертным обучающим системам, действующим на основе практического использования идей искусственного интеллекта.

И.И.Мархель, В.И.Отенко.

АВТОМАТИЗИРОВАННЫЙ ЭКЗАМЕН — вид итогового контроля знаний с применением компьютера, целью которого является проверка и оценка уровня полученных обучаемым знаний, умения применять эти знания в решении практических задач, овладения практическими навыками и умениями в объеме требований учебных программ. Осн. особенность А.э. состоит в диалоге экзаменуемого с компьютером, снабженным контролирующей программой, которая состоит, как правило, из набора контрольных знаний, системы анализа действий экзаменуемого, методики оценки его работы. Экзаменуемый может использовать следующие способы ответов: кодированный, состоящий из ввода совокупности символов в определенной последовательности; графический — представление ответа графиком, фигурой или структурой; числовой — кодирование ответа в виде числа; результативный — ввод результатов расчетов; естественный, предполагающий использование грамматики применяемого в системе языка; выборочный, состоящий в вводе элементов из ряда предложенных.

При проведении А.э. обучаемые занимают автоматизированные рабочие места (АРМ) и по команде преподавателя регистрируются.

Количество обучаемых, одновременно сдающих А.э., устанавливается преподавателем в зависимости от возможностей контролирующей программы, времени, отведенного на А.э., сценария его проведения и т.п. После завершения регистрации на экране каждого экзаменуемого появляется в определенной последовательности серия *вопросов*, предназначенных для проверки уровня полученных им знаний. Тему (раздел) дисциплины выбирает компьютер случайным образом в соответствии с заданным *автором* контролирующей программы законом распределения. Методика составления вопросов позволяет последовательно и в полном объеме оценить знание заданной темы. Об окончании ответов экзаменуемый сообщает преподавателю, который анализирует результаты машинного опроса и в зависимости от этого принимает решение: завершить проверку знаний; продолжить опрос, перейдя к другим разделам дисциплины; углубиться в учебный материал с анализом допущенных экзаменуемым ошибок (если это не делает сама контролирующая программа). Проверка знаний завершается формированием оценки, проводимым с учетом количества допущенных ошибок, веса ошибочно выполненных тестов, веса типа допущенной ошибки, затраченного времени и т.д. Уровень умений применять знания в решении практических задач проверяется путем предъявления на экране компьютера экзаменуемому задания разрешить конкретную конфликтную ситуацию, отыскать неисправность в работе системы (механизма), изложить последовательность действий для достижения требуемого результата и т.д. При подготовке ответа экзаменуемому можно предоставить возможность пользоваться специально разработанной к экзамену *информационно-справочной системой*, содержащей формулы, графики, таблицы, наставления и пр. Контролирующая программа анализирует и оценивает ответ экзаменуемого, учитывая полноту и правильность последовательности его действий. Проверка овладения практическими навыками и умениями (решение задачи, составление *алгоритма*, перевод *текста* с иностранного языка, формирование изображения и т.д.) может производиться на АРМ или другом *рабочем месте*. В последнем случае оценка выставляется преподавателем.

Итогом А.э. может служить общепринятая в учебных заведениях оценка по четырехбалльной системе, но применяются и другие шкалы. А.э. повышает эффективность труда преподавателя, дает возможность объективно оценивать уровень подготовленности обучаемых.

Б.А.Мельников.

АВТОР — создатель произведения, т.е. результата духовной, творческой деятельности одного или нескольких человек, выраженного в определенной форме. В условиях *компьютерной технологии обучения* А. — *пользователь*, являющийся специалистом по созданию

автоматизированного учебного курса. Задачи А.: разработка сценария АУК, подбор и структуризация учебного материала и заданий, написание программы курса, ввод, редактирование и отладка программы курса. Для создания АУК А., как правило, использует соответствующие инструментальные системы.

В.А.Третьяк.

АВТОРСКАЯ СИСТЕМА — диалоговая программа, предоставляющая авторам учебного материала, не являющимся специалистами в области программирования, средств проектирования и реализации программ учебного назначения. Преимущество А.с. по сравнению с авторскими языками программирования — предоставление пользователю возможности работы в режиме выбора из предлагаемых вариантов, уточнение параметров заранее подготовленных схем и наполнение системы учебной информацией, что позволяет значительно повысить производительность труда. Каждая А.с. характеризуется набором различных свойств. В зависимости от них различают А.с. универсальные (охватывают ПУН различных классов) или специализированные (предназначены для создания ПУН либо их проектов заданной жесткой структуры); открытые (в создаваемую ПУН можно включать программы либо их фрагменты на др. языках программирования) или закрытые; многоуровневые (с иерархической структурой компонентов, причем компоненты более высоких уровней используют результаты работы компонентов более низких уровней; напр., в графическом редакторе автор готовит фрагменты учебной информации, которые затем в редакторе межфрагментных связей соединяются в последовательности, представляемые обучаемому) или одноуровневые; адаптивные (для пользователей с разным уровнем подготовки предусмотрены разные способы взаимодействия) или неадаптивные; интерпретирующие или компилирующие (определяются наличием интерпретатора или компилятора для создаваемой в А.с. ПУН; в составе А.с. возможно наличие как интерпретатора, так и компилятора, причем интерпретатор используется для отладки ПУН — см. *Отладка программы*, — а компилятор — для получения окончательной версии ПУН, предназначенной для обучаемого); с наличием промежуточного языка программирования (все или часть результатов, полученных при работе с А.с., выражаются на некотором др. ЯП, напр. авторском, и могут быть доступны для дальнейшей разработки) и без него.

А.с. представляет собой интегрированную систему, состоящую из набора универсальных или специализированных редакторов, интерпретатора и/или компилятора, а также системы отладки. Для хранения информации А.с. использует специализированную либо универсальную базу данных. В архитектуре А.с. выделяют три

компонента, выполняющие следующие функции: 1) создание информационных учебных материалов; 2) создание управляющей части ПУН; 3) сервисные функции. Первый компонент в своем составе имеет *текстовый редактор*, графический редактор, редактор шрифтов, редактор мультимедиа, музыкально-звуковой редактор, редактор видеoinформации. Второй компонент состоит из специализированных редакторов для таких объектов ПУН, как *понятие*, *задача*, фрагмент, урок, *модель обучаемого*, *стратегия* управления и т.п. Редакторы этого уровня основаны на заполнении форм, выборе из *меню*, формировании таблиц, исполнении директив. Третий компонент А.с. включает: справочную подсистему по возможностям А.с., интерпретатор и/или компилятор ПУН, подсистему отладки ПУН, *систему программирования* для промежуточного языка, средства работы с БД, *файлами* и библиотеками.

Современные А.с. (*РАКУРС*, *AUTOOL*, *WISE*) предназначены для проектирования и реализации простых классов ПУН, таких, как демонстрационные, контролирующие, обучающе-контролирующие и информационно-справочные. Осн. проектируемой единицей А.с. является фрагмент, состоящий, как правило, из учебной информации, *вопроса*, ввода и сравнения *ответа* обучаемого с текстовыми образцами. В основе большинства А.с. лежит весьма примитивная бихевиористическая *модель процесса обучения*, а осн. эффект от применения компьютера в обучении достигается за счёт совершенных технических средств — многоцветных *терминалов*, мультимедиа и компактных видеодисков. Дальнейшее развитие А.с. будет происходить по таким направлениям: 1) поддержка более ранних этапов жизненного цикла ПУН, напр., концептуального проектирования и проектирования сценариев ПУН (см. *КОНЦЕПТ*); 2) поддержка более развитых психолого-педагогических моделей обучения, что приведет к выделению высокоуровневых единиц проектирования, таких, как стратегия обучения, цель обучения, учебное взаимодействие, модель обучаемого, изучаемая *предметная область*, диагностика ошибок обучаемого и т.п. (подобные единицы имеются в А.с. *LDS*); 3) с одной стороны, создание мощных интегрированных универсальных сред, а с другой — разработка узко специализированных А.с., предназначенных для различных специалистов, входящих в коллектив разработчиков ПУН, или определенных категорий авторов и позволяющих значительно повысить их производительность труда; 4) создание адаптивных и обучающих А.с., обеспечивающих гибкий и дружественный *интерфейс* с авторами ПУН (см. *ADAPT*).

Т.А.Лисюк, В.А.Петрушин.

АВТОРСКИЙ ЯЗЫК ПРОГРАММИРОВАНИЯ — проблемно-ориентированный язык программирования, предоставляющий авторам средства разработки программ учебного назначения. Состоит из

встроенных (системных, проблемных, специализированных) типов данных и управляющих конструкций, позволяющих эффективно программировать обучающий диалог. А.я.п. содержит также средства для хранения и поиска учебной информации; представления информации пользователю и ввода ответа; сопоставления ответа пользователя с образцами правильных и предполагаемых неправильных ответов, ...одготовленных автором или вычисленных программой; ведения протокола процесса взаимодействия системы с пользователем; обработки числовой и текстовой информации; управления процессом выполнения ПУН. А.я.п. входят, как правило, в состав автоматизированных систем обучения, а также могут являться компонентами авторских систем. Различают следующие типы А.я.п.: рамочные, дескрипторные, декларативные и представления педагогических знаний. Рамочные А.я.п. (*Курсрайтер*, *Язык описания курсов* (ЯОК), *Тьютор*, *ABC-1*) содержат низкоуровневые проблемно-ориентированные средства, в основе которых лежит модель процесса обучения вида "вопрос — текстовый ответ — проверка правильности ответа по заранее подготовленным образцам", реализованная в осн. конструкции языка — фрагменте (ЯОК, *ABC-1*), кадре или единице (Тьютор). Программа на рамочном языке представляется последовательностью фрагментов, связь между которыми осуществляется с помощью операторов перехода. Дескрипторные А.я.п. (*Дидикт* и др.) включают средства для классификации учебного материала (кадра, темы) с педагогической точки зрения (вводный, основной, дополнительный учебный материал; понятие, упражнение, подкрепление, образец ответа), а также по уровню сложности учебного материала и предполагаемым классам пользователей. Программа на дескрипторном языке представляет собой сценарий (схему) прохождения кадров и тем в зависимости от ответов пользователя. Декларативные А.я.п. (*ЛЯОК*) содержат средства для построения ПУН заданного класса посредством параметризации встроенной модели ПУН и ввода необходимого учебного материала. Программа на декларативном языке представляет собой набор информационных кадров и параметризованных проблемно-ориентированных конструкций типа "курс", "задание", "вопрос", "реакция", "анализ" и т.п., задающих значения параметров встроенной стратегии исполнения ПУН. А.я.п. для представления знаний (*ABC-2*) содержит средства описания изучаемой предметной области, решаемых задач (вопросов), типичных ошибок пользователей при решении задач, методов диагностики знаний обучаемых и стратегий обучения. Программа на языке представления знаний состоит из совокупности баз знаний и методов их обработки. Использование высокоуровневых конструкций в А.я.п. существенно упрощает разработку ПУН, что делает доступным создание ПУН и облегчает адаптацию к учебному процессу силами преподавателей, методистов и психологов, не являющихся профессиональными программистами.

В.А. Петрушин.

АДА — язык программирования для задач вычислительного характера, системного программирования, параллельных вычислений; ориентирован на использование во встроенных системах с *управляющими электронными вычислительными машинами*. Назван в честь первой в мире программистки Ады Лавлейс, составившей программы для механической вычислительной машины англ. ученого Ч.Бэббиджа (1792-1871). Разработан в 1975-80 группой специалистов из разных стран под руководством Ж. Ишбиа по инициативе министерства обороны США.

В языке воплощены многие современные концепции *программирования*: ориентация на четкую дисциплину использования конструкций языка, иерархичность процессов разработки и последовательная детализация программ. Использован ряд конструкций, содержащихся в языках *АЛГОЛ-68*, *Паскаль*. Специфические возможности языка предназначены для программирования процессов, протекающих в реальном времени, в т.ч. для программирования нескольких параллельно происходящих процессов. Имеются средства высокого уровня для описания взаимодействия задач, синхронизации их выполнения, параллельного выполнения операторов одной задачи, планирования обработки задач в зависимости от их приоритетов. Эти средства сочетаются с возможностью отдельной компиляции модулей, что облегчает разработку больших программ, содержащих параллельно выполняемые модули. Как единое целое с языком определяются средства разработки программ. На основе А. создан ряд систем программирования, обеспечивающих проектирование, разработку, документирование и сопровождение программ.

В.А.Бардадым.

АДАРТ — универсальная многоуровневая адаптивная авторская система с промежуточным языком программирования, предназначенная для автоматизации этапов проектирования и реализации демонстрационных, контролируемых, обучающе-контролируемых и информационно-справочных программ учебного назначения. Для авторов ПУН, имеющих различный уровень подготовки, А. предоставляет четыре режима работы: интервью, меню двух уровней сложности и программирование на специализированном авторском языке программирования.

В режиме интервью система управляет процессом разработки учебной программы, задавая автору серию вопросов, на которые автор отвечает "да" или "нет" либо вводит оператор авторского языка. Если автор больше не нуждается в таком пошаговом управлении, он может перейти в режим меню первого уровня, в котором та же серия вопросов выдается на экран одновременно в виде простых меню.

Первый уровень меню включает проверку *ответа*, графику, видео, звук и ограничение времени ответа *обучаемого*. Допускается наложение *текста* и графики на видеофрагменты. Меню второго уровня являются расширением меню первого уровня и позволяют создавать достаточно сложные курсы. На втором уровне подключаются возможности мультимпликации, выбора цвета, *анализа ответов обучаемого*, задания переменных, зависящих от среды на этапе выполнения. Режим из меню второго уровня можно вызвать на первом уровне. Автор выбирает режим касанием *светового пера* или вводом первых трех букв на нижней строке экрана. Последний метод позволяет использовать те режимы, которые в данный момент не присутствуют на экране, или вводить цепочки режимов, пропуская вывод на экран промежуточных меню.

Наиболее мощное средство системы — авторский ЯП. На этом уровне может работать программа, пишущий автор. Некоторые режимы меню могут быть доступными на этом уровне. *Команды* авторского языка часто имеют те же наименования, что и режимы в меню.

Преимуществом данной системы является то, что *файл*, созданный на одном уровне, можно использовать на другом уровне, поскольку на любом уровне фактически создается программный код на авторском языке. Но лишь на третьем уровне автор имеет доступ к этому коду непосредственно. Система дает возможность выбирать встроенную *стратегию* обучения, соответствующую целям обучения (напр., обучение классификации понятий, применению правил, линейным процедурам, моделированию и т.д.). Автору с недостаточным опытом разработки ПУН можно перечислить обязательные и необязательные компоненты для каждой стратегии (напр., цель, обобщение, практическое упражнение, помощь и т.д.) и требования для каждого компонента (напр., проверка ответа для практического упражнения). Следовательно, при создании ПУН автор выбирает нужное ему средство из матрицы, в которой на одной оси находятся режимы работы, соответствующие различным уровням подготовки автора, а на второй — стратегии обучения (встроенные или отсутствие встроенных стратегий), соответствующие различным целям обучения. Проблема подготовки автора решается самим многоуровневым характером этой системы. Подсказки по текущим режимам доступны с помощью клавиши "ADVICE". Каждый уровень формата редактора предъявляет свои требования к подготовке автора. Система может пресечь попытку автора перейти на уровень, не соответствующий его подготовке. Большой объем материала может ввести персонал, не имеющий опыта в программировании.

Т.А.Лисюк.

АДАПТАЦИИ ЗАДАЧА — задача индивидуализации обучения, при решении которой обеспечивается сохранение показателя каче-

ства процесса обучения в некоторой заданной области его значений. В компьютерной технологии обучения показателем качества обучения обычно является некоторая функция от оценки качества знаний, достигнутых обучаемым между пре- и пост-тестом, и от времени обучения. Частным примером А.з. может служить минимизация времени обучения при условии, что оценка его качества не меньше требуемого значения оценки. Для решения этой задачи адаптивная автоматизированная обучающая система (АОС), экспертная обучающая система должны вырабатывать такие регулирующие воздействия (способы представления учебного материала, подсказки, примеры), при прохождении которых повышаются или стабилизируются текущие оценки усвоения содержания материала обучаемым. Различают два типа адаптивных АОС поисковые и аналитические. Поисковые АОС осуществляют целенаправленное изменение регулирующего воздействия для достижения оптимального значения показателя качества. Аналитические АОС в ходе процесса обучения накапливают информацию, необходимую для вычисления эффективного регулирующего воздействия, после чего способ изложения учебного материала приспособляется к индивидуальным особенностям обучаемого без поиска. В ранних адаптивных АОС для достижения эффекта адаптации автор программы учебного назначения должен был предусматривать несколько способов изложения одного и того же учебного материала, что резко увеличивало трудоемкость создания этих программ и препятствовало распространению адаптивных АОС. В связи с развитием персональных электронных вычислительных машин, методов синтеза текстов, генерирующих автоматизированных обучающих систем вновь возник интерес к решению А.з. в экспертных обучающих системах.

А.М.Довгялло.

АДАПТАЦИЯ (от лат. *adapto* — приспособляю) — способность устройств либо систем приспособляться к изменяющимся условиям окружающей среды или/и к своим внутренним изменениям, что приводит к повышению эффективности их функционирования. Понятие А. связано с биологическими системами, в ходе эволюции которых сформировались механизмы А. Для человека большое значение имеет физиологическая А. анализаторов к действию специфических раздражителей, напр., зрительного анализатора — к свету или темноте, слухового — к звуку, кожного — к механическим и температурным раздражителям, вестибулярного аппарата — к вращательному движению. См. также *Адаптации задача, Адаптивная обучающая программа, Адаптивность авторизированных учебных курсов, Адаптивность системы обучения.*

АДАПТИВНАЯ ОБУЧАЮЩАЯ ПРОГРАММА — обучающая программа, в которой последовательность прохождения учебного материала и характер его изложения зависят от истории обучения и индивидуальных особенностей обучаемого. Адаптацию к обучаемому можно реализовать, изменяя управление обучением, методику преподавания, а также стиль общения. В последнем случае в зависимости от индивидуальных психологических свойств обучаемого выбираются стиль и тон сообщений, временные характеристики взаимодействия (время, отводимое на выполнение задания, частота оказания помощи обучаемому, допустимое время реакции и т.п.) с целью создания "дружественной среды" и повышения комфортности общения. Управление обучением в А.о.п. основано на оценке эффективности использования дидактико-методических средств и анализе результатов познавательной деятельности обучаемого. Гибкость управления обучением в А.о.п. выражается в изменении соотношения инициативы обучаемого и системы, формулировке различных по сложности заданий, варьировании степени подробности сообщений об ошибках и их причинах и т.п. Проблему адаптации к психологическим особенностям частично можно решить статистически настройкой параметров обучающей программы по результатам предварительного тестирования обучаемого. Реализация качественной А.о.п. требует решения задач диагностики и оптимизации, в частности, диагностики состояния знаний обучаемого и оптимизации выбора очередного обучающего воздействия. Выбор того или иного подхода остается за автором А.о.п., который исходит, как правило, из возможностей сбора и обработки данных о процессе обучения в используемом средстве компьютерной технологии обучения. В автоматизированных обучающих системах обычно накапливаются данные об ошибках обучаемого, времени выполнения заданий и истории обучения. Экспертные обучающие системы поддерживают модель обучаемого, более полно отражающую его характеристики.

Э.Н.Кунат, Е.М.Синица.

АДАПТИВНОСТЬ АВТОМАТИЗИРОВАННЫХ УЧЕБНЫХ КУРСОВ — свойство автоматизированных учебных курсов приспособляться к некоторым психологическим особенностям обучаемого в интересах оптимальной организации процесса обучения. Заключается в том, что на основе анализа ответа обучаемого выдаваемая в диалоге информация строится индивидуально для обеспечения оптимального усвоения. А.а.у.к. реализуется посредством строгого управления в обучающей программе, различием в стиле и тоне замечаний, указанием на ошибки и их причины, формулировкой различных по трудности или по содержанию заданий, изменением временных границ, отводимых на выполнение задания, и т.п. Выбор возможностей опр. делается автором обучающей программы, причём

свобода выбора ограничена дидактическим качеством имеющейся *автоматизированной обучающей системы*. Реализация высокой А.а.у.к. требует решения проблем диагностики и оптимизации. В научных исследованиях на первый план выступает проблема диагностики, т.к., во-первых, высокий уровень диагностики является осн. предпосылкой для создания *адаптивных обучающих программ*, а во-вторых, автоматизированные учебные курсы позволяют быстро и с относительно небольшими затратами получать данные, характеризующие психологические свойства обучаемого. Поскольку данные извлекаются в процессе *автоматизированного обучения*, необходимо анализировать прежде всего ошибки, время работы и историю обучения. Эти данные можно дополнить таким образом, чтобы обеспечить индивидуальное управление прохождением учебного материала.

Э.Н.Кунат.

АДАПТИВНОСТЬ СИСТЕМЫ ОБУЧЕНИЯ — автоматическое приспособление системы к изменению внешних условий и сохранение способности эффективного достижения дидактических целей при изменении характеристик *обучаемого*. Адаптивным системам свойственны: малый объем априорной *информации*, неопределенных или измененных-условий; малая чувствительность к изменению свойств объекта и среды; получение информации об условиях в процессе работы; использование информации для изменения своего поведения; направленность программы на достижение определенной цели.

Задача управления обучением относится к числу нестационарных, т.к. значения неопределенных параметров изменяются во времени. Цель обучения предполагает оптимальное достижение определенного уровня знаний при некоторых временных ограничениях.

По типу механизма выделяют параметрическую, алгоритмическую и структурную *адаптацию*. При параметрической адаптации приспособление системы заключается в настройке параметров *алгоритма* управления. При алгоритмической адаптации настраивается алгоритм управления путем изменения порядка проверки условий выполнения операций в рамках заданного набора. Структурная адаптация заключается в структурных преобразованиях системы управления: изменению подвергается набор выполняемых операций и правил их упорядочивания.

*Ю.И.Лобанов.

АДМИНИСТРАТОР БАЗЫ ДАННЫХ — лицо (или группа лиц), ответственное за функционирование *базы данных* и ее развитие. Знаком с теорией обработки данных и со спецификой *предметной области* по данной *информационной системе*, реализующей централизованное управление БД посредством *системы управления*

базами данных. При работе с БД прикладному программисту часто приходится менять, добавлять, модифицировать старую запись, в этом случае он должен обратиться к А.б.д. за разрешением, т.к. только А.б.д., ответственный за систему, может иметь дело со структурой данных в целом.

Осн. функция А.б.д. - обеспечение независимости структур данных взаимосвязей между ними, эффективных для обслуживания всего коллектива *пользователей.* Развитие БД происходит по мере разработки и интеграции прикладных программ расширения БД, поэтому функции А.б.д. долгосрочны и направлены на координацию всех этапов *проектирования, реализации и* поддержания БД.

В персональных *банках данных,* когда БД является самостоятельной единицей, обслуживаемой одним пользователем, пользователь выступает также в роли администратора базы данных.

А.П.Ильяшенко, М.Е.Козлов.

АДОНИС — адаптивная диалоговая *информационная система,* предназначенная для разработки средств *информационного обеспечения* учебного процесса в рамках новых информационных технологий и организации *автоматизированного обучения и контроля знаний* по различным дисциплинам. Относится к классу *автоматизированных обучающих систем* с возможностью построения разветвленных *автоматизированных учебных курсов* со сложной адаптивной структурой. А. включает: средства обеспечения самообучения, *группового обучения* и контроля знаний с развитым *управлением процессом обучения;* средства интерактивной *машинной графики,* обеспечивающие создание и редактирование черно-белых и цветных графических изображений, разработку и интерпретацию динамических изображений (мультипликацию); средства обеспечения расчетов в режиме калькулятора (с памятью) для *авторов курсов и обучаемых;* средства языкового управления логикой работы учебной программы; средства конвертирования курсов из систем АСТРА / МИКРО (см. АСТРА) и РАКУРС. Реализована для диалоговых вычислительных комплексов ДВК-2, ДВК-3, ДВК-3.2, ДВК-4, персональной *микро-ЭВМ "Электроника-85", персональных электронных вычислительных машин* ЕС-1840, ЕС-1841, ЕС-1842, Искра-1030, Повец-16, Мазовия, IBM PC/XT, IBM PC/AT, PS/2.

А. используется как инструментальная *авторская система* для разработки АУК и создания *информационно-справочных систем* учебного процесса на базе персональных компьютеров, для работы в режиме обучения и контроля знаний в составе классов на базе персональных компьютеров и в составе *локальной вычислительной сети.* Осн. функции А.: организация библиотеки курсов на диске; резервирование курса в библиотеке; формирование и корректировка параметров курса в режиме *диалога;* копирование фрагментов курса;

распечатка содержимого курса на *печатающем устройстве*; удаление и переименование курса; просмотр оглавления библиотеки; интерпретация курса в режиме обучения и контроля с обеспечением *анализа ответа обучаемого* с помощью развитого анализатора ответов; создание проблемно-ориентированных справочно-информационных подсистем обеспечения учебного процесса в рамках информационных технологий обучения.

В.И.Васильев, А.Е.Пасхин, А.Н.Филиппов.

АДРЕС — символическое (цифровое, буквенно-цифровое) либо кодовое обозначение элемента *памяти ЭВМ* или *порта* ввода—вывода. В *мини-ЭВМ* и *микро-ЭВМ* код А. устанавливается на адресной шине в каждом цикле обращения *процессора* к памяти или системе ввода—вывода. Ячейка памяти и программно доступный порт ввода—вывода обладают своим уникальным А., код которого должен быть установлен на шине процессором (микропроцессором) или др. активным устройством для идентификации обращения к конкретной ячейке памяти либо порту ввода—вывода.

В.И.Сигалов.

АЙДА (Архитектура И Диалоговый Ассемблер) — *многофункциональный автоматизированный учебный курс*, предназначенный для применения на основных этапах процесса обучения *программированию ЕС ЭВМ на ассемблера языке* - от разработки алгоритмов и изучения языка до конструирования и *отладки программ*. Обеспечивает и поддерживает *обучение*, контроль, тренировку и самостоятельную работу *пользователей* в следующих режимах: изучения архитектуры и принципов функционирования ЕС ЭВМ; обучения конструированию и описанию алгоритмов; *диалогового программирования* на ассемблере; отладки *программ* в среде *операционной системы*.

В режиме изучения *архитектуры ЭВМ* словесное описание учебного материала дополняется моделированием осн. объектов, доступ к которым имеет программист, имитацией на экране динамики работы *процессора* при выполнении *команд* программы, механизма смены слов состояния программы при обработке прерываний и т.д. Управление обучением осуществляется по двухуровневой схеме: первый уровень — управление разделами, второй — управление обучающими модулями. На первом уровне используется линейный тип *обучающей программы* с возможностью произвольного доступа к разделам, на втором — разветвленный тип обучающей программы. В функции обучающего модуля входят обучение и текущий контроль *знаний* по элементарным *понятиям* (и их связям с др. понятиями), включенным в раздел предметного содержания. В режиме применена типовая структура обучающего модуля, включающая несколько

однотипных, случайно выбираемых упражнений, количество которых не фиксируется и для разных *обучаемых* может быть различным. Управление выбором заданий осуществляется единой для всех модулей *подпрограммой* оценки успеваемости.

Режим обучения конструированию и описанию алгоритмов представляет собой *автоматизированную контрольную работу* с элементами самообучения. Язык описания алгоритмов, изучаемый в данном режиме, позволяет описывать произвольные алгоритмы, заданные в виде блок-схем и ориентированные на последующую реализацию средствами ассемблера. В состав языка включены три группы *операторов*: декларативные — определение памяти и определение констант; императивные — пересылки и преобразования *данных*, арифм. *операций*, рекурсивного присваивания, ввода и вывода *данных*; предикатные — сравнения *данных*. Операторы переходов в состав языка не входят, т.к. все управляющие связи программы задаются в процессе описания блок-схемы.

Режим диалогового программирования реализует методику алгоритмического подхода и на начальных этапах обучения играет роль "стартового ускорителя", обеспечивая немедленное получение результатов счета по программе в условиях миним. знакомства с языком ассемблера и отсутствия навыков работы с операционной системой ЭВМ. Режим обеспечивает диалоговое конструирование (генерацию) программ на основе операторного описания алгоритма решаемой *задачи* и уточняющих *данных*, запрашиваемых у *обучаемого* в процессе *диалога*. Набор решаемых задач является произвольным и определяется *преподавателем*. В этот набор можно включить задачи, использующие средства целочисленной и десятичной арифметики, задачи обработки символьных *данных*, логические задачи и др. Обучающий эффект режима диалогового программирования проявляется на этапе анализа распечатки программы путем сопоставления *исходной информации* об алгоритме и тех операторах языка ассемблера, которыми реализуется алгоритм.

В режиме отладки программ *обучаемому* предоставляются возможности ввода и корректировки *исходных текстов* программ, запуска заданий на выполнение в пакетном режиме, просмотра результатов выполнения заданий, просмотра системных очередей и активных заданий и др. Перечисленные функции реализуются средствами интегрированного пакета прикладных программ АОС-ВУЗ/ПРИМУС.

Управление работой осуществляется с помощью команд: ПОМОГИ — предоставление вспомогательной информации по изучаемому разделу или выполняемому заданию, РЕЖИМ — предоставление возможности смены режима работы, ВЕРНУТЬ — возврат в предыдущий режим работы, ЭКРАН — включение (отключение) вывода учебного материала на экран, БИБЛИОТЕКА — предоставление произвольного доступа к библиотеке учебного обеспечения

курса. В курсе применена экранная реализация языка команд обучаемого. Экран делится на две зоны: зона ведения диалога - верхняя большая часть экрана и зона команд - нижняя часть экрана. Зона команд постоянно находится на экране в процессе работы с курсом и в ней отображены все используемые команды. Выбор требуемой команды и восстановление зоны команд и текстов в зоне диалога при повторных подключениях обучаемого либо после к.-л. его действий, нарушивших формат экрана, выполняется с помощью функциональной клавиатуры. В состав учебно-методического обеспечения входят руководства преподавателя и обучаемого по работе с курсом и два учебных пособия: "Основы архитектуры и принципов функционирования ЕС ЭВМ" и "Конструирование и описание алгоритмов". Все документы можно распечатать с магнитного носителя информации в требуемом количестве экземпляров. Однако обучаемый может обращаться к учебным пособиям и не имея перед собой печатного варианта. Воспользовавшись командой БИБЛИОТЕКА, он получает оперативный доступ к текстам различных учебных пособий, имеющихся в библиотеке курса. С помощью простых подкоманд обучаемый просматривает содержание выбранного пособия, знакомится с текстами любого раздела и главы.

А.Ф.Лысиков.

АККОМОДАЦИЯ (лат. *accomodatio* — приспособление) — способность глаза приспособляться к чёткому видению предметов, находящихся от него на различном расстоянии. Действие механизма А. основывается на способности хрусталика изменять свою кривизну с помощью аккомодационной (цилиарной) мышцы и цинновой связки. А. может осуществляться лишь в определенном диапазоне - между ближайшей и дальнейшей точками ясного зрения. Определяется объемом, силой, амплитудой или шириной А., которые выражаются в *диоптриях*. При длительной работе с видеотерминалами у пользователей могут возникать нарушения А., сопровождающиеся астенопическими жалобами (см. *Астенопия*).

А.С.Коваленко, В.Г.Мартиросова.

АЛГОЛ-60 (от англ. *ALGO*rithmic Language — язык алгоритмов) — алгоритмический язык, ориентированный на описание алгоритмов решения задач численного анализа. Разработан в 1960 международной группой специалистов под руководством амер. ученого Дж. Эккуса. При создании А.-60 впервые были использованы понятия и идеи, оказавшие влияние на многие более поздние языки программирования (*ПЛ/1*, *Симула-67*, *АЛГОЛ-68*, *Паскаль*, *Ада*, *Модула-2*): блочная структура программ, развитые управляющие структуры (полный условный оператор, цикл с предусловием и др.), вложенные структуры, области действия описаний, возможность

динамического распределения памяти, развитые средства использования *процедур*, различные механизмы передачи параметров в процедуры. В то же время при разработке А.-60 не рассматривались многие вопросы практической реализации (напр., стандартные средства ввода-вывода), что ограничило использование языка для решения практических *задач*. А.-60 сыграл значительную роль в обучении специалистов в области *программирования*, в частности для иллюстрации осн. понятий и методов программирования, а также как средство решения *учебных задач*.

В.А.Бардадым.

АЛГОЛ-68 — многоцелевой универсальный язык *программирования*. Разработан в 1968 международной группой специалистов под руководством А.ван Вейнгардена. В А.-68 определены набор базовых объектов и *понятий* и гибкий набор правил создания из них различных языковых версий, в частности проблемно-ориентированных. К базовым объектам относятся пять примитивных типов *данных* (логические значения — истина и ложь, целые числа, вещественные числа, литеры, строки) и четыре *оператора* — присваивание, условный оператор, цикл и вызов *процедуры*. Из значений примитивных типов можно образовывать прямоугольные массивы произвольной размерности с динамически изменяемыми границами, иерархические структуры данных произвольной глубины. Операторы могут выполняться последовательно или параллельно. Средства расширения языка позволяют конструировать новые типы данных из примитивных, определять над данными новых типов произвольные операции на основе базовых примитивных операций. Предусмотрены также средства, облегчающие переносимость *программ*, их настройку на различные типы машин и устройств ввода-вывода. При разработке А.-68 появилось много новых концепций, представляющих большой теор. интерес и нашедших развитие в более поздних ЯП.

В.А.Бардадым.

АЛГОРИТМ (лат. *algorithmis*, от арабского имени узбекского математика IX в. аль-Хорезми) — предписание о выполнении строго определённой последовательности *операций*. Точнее — предписание о выполнении активной системой Q *процедуры*, состоящей из эффективных (для системы Q) операций (таких, что каждая из них обеспечивает получение определённого результата) и не содержащей разветвлений с неоднозначным исходом. Как правило, по меньшей мере одна из предусматриваемых А. операций является *родовой*; иными словами, совокупность исходных *данных* А. (имеющихся к началу его реализации *операндов* указанных операций) может принимать различные значения в некотором их диапазоне. А. издавна

разрабатывались в математике как *средства решения массовых задач* (простейшими примерами могут служить А. извлечения квадратного корня, нахождения наименьшего общего кратного двух чисел и др.). Для *решения задач с помощью компьютера* соответствующие А. воплощаются в компьютерных программах.

Алгоритмом решения некоторой массовой задачи наз. А., обеспечивающий её решение при любых допустимых значениях исходных данных. Понятие об А. решения задачи следует отличать от общего понятия А. Смещение рассматриваемых *понятий* привело в свое время к неясностям в понимании сущности так наз. эвристических программ; с помощью этих программ часто удается решить на компьютере задачи, А. решения которых могут быть вообще неизвестны. Эвристическая, как и всякая компьютерная, программа реализуется какой-то А., однако он не является А. решения любой задачи того класса, на который рассчитана программа, и поэтому не для всех задач этого класса обеспечивает получение правильного результата решения. Термин "А." часто употребляется для обозначения предписаний, используемых в управлении человеческой деятельностью и лишь приближающихся по своим свойствам к описанным выше А. (см. *Квазиалгоритм*).

Г.А.Балл.

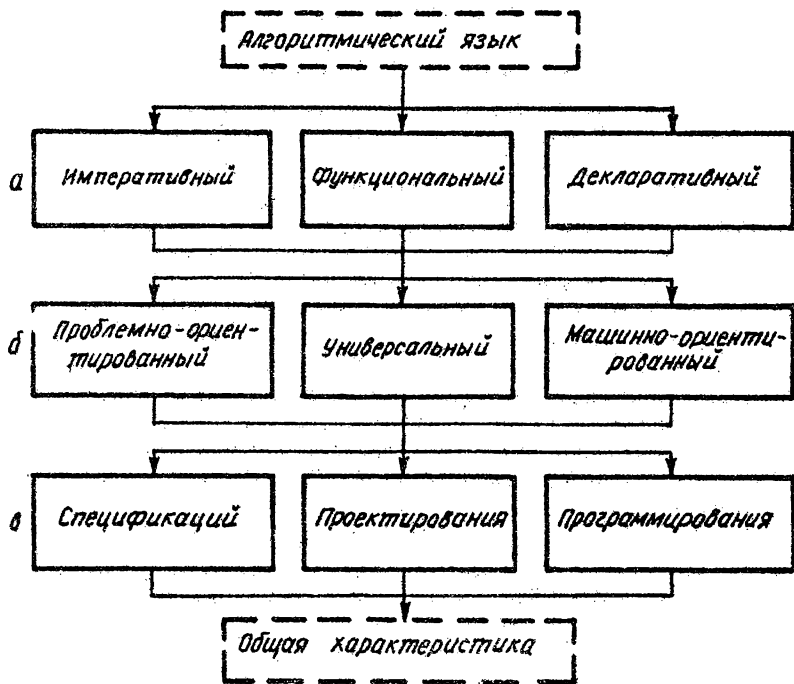
АЛГОРИТМИЧЕСКАЯ КУЛЬТУРА — часть математической культуры и общей культуры мышления, предполагающая формирование умений, связанных с пониманием сущности *алгоритма* и его свойств. В широком смысле слова под А.к. обычно понимают совокупность представлений, умений и навыков, связанных с решением *задач* и осуществлением целенаправленной деятельности. Иногда в этом значении употребляют и понятие "алгоритмическое мышление", которое в узкопсихологическом контексте предполагает формирование умственных действий *обучаемых* на уровне алгоритмов, т.е. такой тип *обучения*, при котором прямым *продуктом деятельности* выступает исполнительная часть способа действий по решению задач (а не ориентированные и контрольные части способа действий). Этот тип обучения не может рассматриваться как эффективный, за исключением особых случаев, когда требуется формирование конкретных навыков, а не развитие обучаемого.

Е.Д.Маргулис.

АЛГОРИТМИЧЕСКИЙ ЯЗЫК — язык или система обозначений, используемые для однозначного описания *алгоритма*. Определяется заданием алфавита (*словаря исходных символов*), синтаксиса (грамматики) и семантики. Описание алгоритма представляет собой совокупность правильных предложений А. я., наз. предписаниями. Набор предписаний, из которого можно построить любые алгоритмы,

наз. алгоритмически полным. На компьютерах различные аспекты А.я. реализуются в виде языковых процессоров.

А. я. различают: по способу представления и обработки данных — императивные языки, функциональные языки и декларативные языки; по ориентированности на предметную область — машинно-ориентированные языки, проблемно-ориентированные языки и универсальные языки; по технологической направленности — языки спецификаций, языки проектирования, языки программирования (ЯП). Приведенная схема (рис.) является классификатором (конструктором для характеристик) А.я. На входе схемы



выбирается некоторый А.я. При движении сверху вниз вдоль стрелок на каждом уровне выбирается характеристика, соответствующая критерию классификации и выбранному А.я. Объединение отдельных характеристик на выходе схемы дает общую характеристику языка. Например, для языка Паскаль схема такова: алгоритмический язык — Паскаль, критерий а — императивный, критерий б - универсальный,

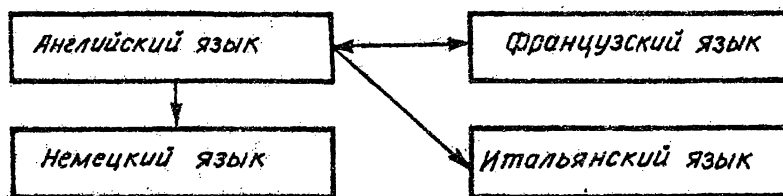
критерий *в* — программирования, общая характеристика — алгоритмический императивный универсальный язык программирования Паскаль. Эта классификация относительна, т.к. реальные А.я. могут сочетать в себе свойства, характерные для языков из разных классов.

При изучении *программирования* для иллюстрации элементарных понятий и методов, а также для решения *учебных задач* используются специально созданные А.я.. Типичный пример А. я. учебного назначения — императивный универсальный язык *проектирования*, разработанный в Вычислительном центре Сибирского отделения РАН в 1985 под руководством А.П.Ершова для школьного курса основ информатики и вычислительной техники. Этот язык позволяет описывать алгоритмы обработки числовых и символьных данных, скалярных величин и массивов, использовать полный условный *оператор*, операторы цикла с предусловием и со счетчиком, оператор выбора, *процедуры* и функции (в т.ч. рекурсивные), создавать алгоритмы управления программируемыми исполнителями. При создании языка использовался опыт работы с учебным ЯП Рапира, разработанным и реализованным в Вычислительном центре Сибирского отделения РАН в 1977-83 под руководством А.П.Ершова и Г.Н.Звенигородского. Среда для поддержки работы со школьным А.я. ("Е-практикум") была разработана в МГУ в 1986-87 под руководством А.Г.Кушниренко. Е-практикум содержит: *интерпретатор*; синтаксически-ориентированный редактор, имеющий средства конструирования алгоритмов в соответствии с принципами структурного программирования; средства *отладки программ*; библиотеки стандартных *подпрограмм* и исполнителей.

В.А.Бардадым, В.А.Третьяк.

АЛГОРИТМИЧЕСКОЕ МЫШЛЕНИЕ — см. *Алгоритмическая культура*.

АЛПС (англ. Automated Language Processing Systems — система автоматической обработки языка) — система *машинного перевода*,



используемая в качестве *автоматизированного рабочего места* переводчика. В ней предусмотрены различные возможности обработки

входного *текста* в зависимости от степени участия компьютера в процессе автоматического перевода текста, что определяется самим переводчиком. Переводчик может использовать АЛПС как вспомогательное средство на уровне редактирования перевода, в качестве автоматизированного *словаря* для выборочного поиска слов, для автоматизации поиска терминов, для интерактивного машинного перевода. АЛПС функционирует с несколькими языковыми парами (рис.). Система разработана в университете Бригема Янга (США).

К.Р.Пиотровская.

АНАЛИЗ ОТВЕТА ОБУЧАЕМОГО — определение качества ответа *обучаемого* согласно установленным критериям. В результате А.о.о. определяются признаки (симптомы), исходя из которых осуществляется *диагностика знаний* обучаемого. Применяется в контролирующих, обучающе-контролирующих, обучающих (в зависимости от *метода обучения*) и тренирующих системах.

Распространенным методом А.о.о. является его сопоставление с заранее подготовленными эталонами *ответов*, каждый из которых относится к определенному типу: правильный, неправильный, предполагаемый, неопределенный и т.д. Автор *обучающей программы* предусматривает типичные признаки ответов, ошибки, а также выдает бессмысленные ответы. В результате совпадения хотя бы с одним из эталонов вырабатывается управляющее воздействие (переход к следующему фрагменту программы, предыдущему фрагменту, запрос нового ответа и т.д.), которое автор обучающей программы может “встроить” в соответствующую систему или предусмотреть. Если ответы представляются на формальном языке, можно использовать методы *синтаксического анализа*. Анализатор контролирует синтаксическую правильность ответа и преобразует его так, чтобы синонимические по смыслу ответы получили одно и то же представление (образ ответа).

При решении *задач*, для которых трудно составить эталоны, используется программный А.о.о. *Авторские языки программирования* обычно имеют средства обращения к специальным *процедурам* анализа ответов на конкретные *вопросы*. Эти процедуры могут выполнять собственно анализ с выяснением ошибок либо преобразование ответа для последующего сопоставления.

В обучающих системах с *моделями предметных знаний* ответы интерпретируются на данной модели. Правильность ответа определяется на основе сопоставления ответа обучаемого и фактов, выведенных *решателем задач* в модели *предметной области*.

Ф.Анакер, Р.А.Ковалюнас, В.А.Петрушин, В.А.Третьяк.

АНИЗОМЕТРОПИЯ (от греч. *ανισοζ.* — неодинаковый, *μετρου* — мера, *οψ* — зрение) — разница в *рефракции* обоих глаз. При высокой степени А. нарушается бинокулярный характер зрения.

В.Г.Мартиросова.

АНТИВИРУСЫ — семейство программ, разработанных для борьбы с *компьютерными вирусами* (КВ). Осн.проблема для пользователей *персональных электронных вычислительных машин*, возникающая в связи с появлением КВ, — как защитить свои программы и *данные*, не прибегая к самоизоляции. Для борьбы с КВ можно применять целый комплекс мер — технических, организационных, юридических. Поэтому наблюдаемая иногда тенденция к самоизоляции совершенно неоправдана.

Простейшим А. является детектор (напр., LD) — программа, позволяющая составить список заражённых программ. В качестве детектора можно использовать имеющиеся программы, способные выполнять поиск строки в одном или нескольких *файлах* на заданном диске либо каталоге. Детектор может быть и резидентным. В этом случае после загрузки программы он проверяет её на заражённость и передаёт ей управление только тогда, когда КВ не обнаружен. Наиболее распространенные А. — так наз. фаги (напр., AJDSTEST, DOCTOR) — программы, “выкусывающие” КВ из зараженной программы и тем самым восстанавливающие ее в виде, близком к первоначальному. Операция “выкусывания” не всегда бывает успешной. Фаги также могут быть резидентными, однако из-за значительного объема резидентные фаги встречаются редко. Третий вид А. — так наз. сторожа (напр., ANTY-KOR) — резидентные программы, контролирующие подозрительные действия запускаемых программ и блокирующие их либо “молча”, либо выдавая сообщение *пользователю*, который может разрешить действие или запретить (в последнем случае программа, предпринявшая опасное действие, может закончиться аварийно). При этом дисковые драйверы, обеспечивающие возможность сегментации *винчестерского накопителя* и присваивания отдельным разделам статуса “только чтение”, можно рассматривать как специальную разновидность сторожей. Четвертый вид — так наз. ревизоры (напр., СНЕС) — программы, подсчитывающие контрольные суммы и другие параметры файлов и сравнивающие их с эталонными (обычно хранятся в отдельном файле). Этот вид контроля наиболее надежен, т.к. при отсутствии в оперативной *памяти ЭВМ* резидентного КВ позволяет выявить все измененные программы независимо от причины, вызвавшей эти изменения. Резидентные ревизоры загружают в память программу, подсчитывают ее контрольную сумму и передают ей управление только тогда, когда контрольная сумма совпадает с записанной в специальном *псле* файла или элемента каталога данного файла. В

противном случае выдается предупреждающее сообщение, и выполнение программы блокируется. Если записать зараженную программу в файловую систему, все остальные файлы которой систематически контролируются ревизором, то наличие КВ можно выявить по заражению других программ для большинства, но не для всех типов КВ. Поэтому очень важно, чтобы в момент запуска ревизора было достоверно известно, что в оперативной памяти нет резидентного вируса, чего можно достичь, загрузившись с эталонной, защищенной от записи, *дискеты* или разместив все компоненты *операционной системы* (ОС) в разделе *винчестера*, имеющего статус "только чтение". И, наконец, наиболее изощренный вид А. — так наз. вакцины (напр., UNJIVAC) — программы, изменяющие среду функционирования КВ так, что он теряет способность к размножению. Различают пассивные и активные вакцины. Пассивная вакцина — пакетная программа, которая за один вызов специальным образом обрабатывает один или все файлы на диске либо в каталоге. Обычно при такой обработке проставляется признак, который КВ использует для того, чтобы отличить зараженные программы от незараженных. Обработанная программа является вакцинированной против данного конкретного КВ. Активные вакцины — резидентные программы, действие которых обычно основано на имитации присутствия КВ в оперативной памяти. Поэтому их обычно применяют против резидентных КВ. Если такая вакцина находится в памяти, то, когда при запуске зараженной программы КВ проверяет, находится ли уже в оперативной памяти его копия, вакцина имитирует наличие к.л.ии. В этом случае КВ просто передает управление программе-хозяину, и его инсталляция не происходит. Простейшие вакцины представляют собой выделенный и лишенный способности к размножению КВ. Поэтому их можно изготовить оперативно, быстрее, чем фаги. Более сложные вакцины (поли-вакцины) имитируют наличие в оперативной памяти нескольких КВ. Приведенный список не исчерпывает всего многообразия А., хотя и охватывает их осн. разновидности. Каждый из А. подобен узкому специалисту в определенной области, поэтому оптимальной тактикой является комплексное применение нескольких видов А. Появление новых КВ будет вызывать появление и новых средств борьбы с ними. Проблема разработки "универсального" А. неразрешима.

Для защиты от КВ применяют различные методы. Важнейшим из них является архивирование: копирование таблицы FAT, ежедневное ведение архивов измененных байтов. Остальные методы не могут заменить ежедневное архивирование, хотя и повышают общий уровень защиты. При входном контроле проверяют поступающие программы детекторами, а также соответствие длин и контрольных сумм в сертификате (*данные*, предоставляемые изготовителем данного *программного обеспечения*) длинам и контрольным суммам полученных программ; для уничтожения *бутовых вирусов*

систематически обнуляют первые три *байта* бут-сектора либо применяют специальную буттовую вакцину, (VITAMINB). При профилактике работают с дискетами, защищенными от записи, минимизируют периоды доступности дискеты для записи, разделяют “общие” дискеты между конкретными пользователями, проверяют детекторами передаваемые и поступающие дискеты, отдельно хранят вновь полученные программы и эксплуатировавшиеся ранее, хранят программы на винчестере в архивированном виде. При ревизии анализируют вновь полученные программы специальными средствами, контролируют целостность с помощью регулярного подсчета контрольных сумм и проверки бут-сектора перед считыванием информации или загрузкой с дискеты, а также содержимое системных файлов (прежде всего COMMAND.COM). Каждая новая программа, полученная без контрольных сумм, должна проходить карантин, т.е. тщательно проверяться компетентными специалистами на известные виды КВ, и в течение определенного времени за ней должно быть организовано наблюдение. Используют также специальное имя пользователя в ADM (Advanced Disk Manager — передовой управляющий диск) при работе со вновь поступившими программами, причем для этого пользователя все остальные разделы должны быть либо невидимы, либо иметь статус READ-ONLY. При сегментации используют ADM для разбивки диска на “непотопляемые отсеки” — зоны с установленным атрибутом READ-ONLY. Для хранения ценной информации используют разделы, отличные от C или D, и не указываемые в PATH. Отдельно хранят исполняемые программы и базы данных. При фильтрации применяют программы-сторожа для обнаружения попыток выполнить несанкционированные действия. Для вакцинирования специально обрабатывают файлы, диски, каталоги, запускают резидентные программы-вакцины. Для автоматического контроля целостности применяют резидентные программы подсчета контрольных сумм перед запуском программ, используют в программе специальные алгоритмы, позволяющие после запуска программы определить, были внесены изменения в файл, из которого загружена программа, или нет. При терапии деактивируют конкретный вирус в зараженных программах специальной программой-антибиотиком или восстанавливают первоначальные состояния программ, “выкусывая” все экземпляры КВ из каждого зараженного файла либо диска с помощью программы-фага.

Для борьбы с КВ необходимо применять многоуровневую защиту, т.к. наличие нескольких уровней позволяет компенсировать недостатки, присущие тому или иному средству защиты. Если КВ обойдет один вид защиты, то он может “споткнуться” на другом. Хорошие результаты дает схема, включающая следующие этапы: входной контроль новых программных средств; сегментацию информации на винчестере; защиту ОС и системных программ от заражения; систематический контроль целостности информации; архивирование.

На этапе входного контроля можно выявить большинство *бутовых* и *файловых вирусов*. Эта процедура отнимает всего несколько минут, сохраняя десятки часов, которые необходимо было бы потратить на дезинфекцию или восстановление уничтоженной информации. Для этого целесообразно использовать специально подобранную батарею детекторов и фагов. При интерпретации результатов следует учитывать возможность ложного срабатывания одного из детекторов. Сегментация винчестера основана на применении специального драйвера, обеспечивающего присвоение отдельным логическим дискам (разделам винчестера) атрибута READ-ONLY, а также простейшей схемы парольного доступа. При этом в "непотопляемые отсеки" (разделы с установленным атрибутом READ-ONLY) можно записать значительную часть исполняемых программ, включая большинство *трансляторов* и *утилит* (вспомогательных программ). Логический диск с ОС не должен быть слишком большим. Туда следует включать только саму ОС и некоторые "стабильные" утилиты (утилиты Нортон, PC Shell и т.д.).

Важно не только иметь последние версии А., но и уметь правильно их применять. Наиболее распространенная ошибка при применении А. — запуск его на компьютере, зараженном резидентным КВ. Вторая типичная ошибка — перезагрузка системы при наличии "зашелкнутой" дискеты в дисковом А. Третья распространенная ошибка — запуск "батареи" фагов до определения, хотя бы ориентировочного, с каким видом КВ пришлось столкнуться. Четвертая типичная ошибка — отсутствие надежного архива программ. И, наконец, пятая типичная ошибка, частая в учебных заведениях, — передача другому пользователю незащищенных от записи дискет для считывания той или иной программы.

Н.Н. Безруков.

АНТРОПОМЕТРИЯ (от греч. *ανθρωπος* — человек, *μετρέω* — измеряю) — совокупность методов и приёмов измерения тела человека и его частей. Важнейшими антропометрическими признаками, связанными с использованием *компьютерной технологии обучения*, являются размеры человеческого тела в положении сидя и границы движений рук, глаз и головы. Учет этих факторов обеспечивает правильную *рабочую позу* и оптимальность движений *обучаемых*. Антропометрические данные обучаемых должны учитываться при определении размерных параметров учебной мебели и *технических средств обучения*, а также при компоновке учебных *автоматизированных рабочих мест*. Для антропометрических измерений используют специальные инструменты: антропометры, толстотные и скользящие циркули, гониометры, ленты и другие. Собранные в результате проведения антропометрических измерений данные

подвергают биометрической обработке и оформляют в виде таблиц, схем и графиков.

Осн. принципы учёта антропометрических данных при проектировании учебных АРМ: 1) необходимо использовать значение антропометрических признаков, соответствующих возрастной группе обучаемых; 2) габариты рабочего места, размеры и взаимное расположение его элементов должны соответствовать антропометрическим характеристикам 90% обучаемых (исключаются 5% обучаемых с наименьшими значениями признаков и 5% — с наибольшими); 3) технические средства необходимо располагать в зонах оптимальной досягаемости и зрительного наблюдения, определяемых на основе антропометрических данных обучаемых с наименьшими размерами тела; 4) требуемый минимум свободного пространства для размещения тела и его перемещения определяются на основе антропометрических данных обучаемых с наибольшими размерами тела.

В.М.Бондаровская, Г.П.Дамянова.

АОС — то же, что и автоматизированная обучающая система.

АОС-ВУЗ — специализированный пакет прикладных программ учебного назначения, являющийся базовым пакетом семейства автоматизированных обучающих систем для вузов. Предназначен для разработки автоматизированных учебных курсов по различным предметным областям и организации автоматизированного обучения на базе ЕС ЭВМ под управлением операционной системы.

В состав технических средств, необходимых для функционирования АОС-ВУЗ, входят: ЕС ЭВМ моделей от ЕС-1022 и выше с объемом оперативной памяти ЭВМ не менее 256 К; дисководы на сменных *дискетах* (можно использовать *накопители* на магнитных лентах); устройства группового управления ЕС-7906 с терминалами ЕС-7066; устройства группового управления ЕС-7920 с терминалами ЕС-7927-1 (локальные) или для одной из версий ЕС-7927-2 (удаленные); штатный набор *периферийных устройств* ЕС ЭВМ — алфавитно-цифровое печатающее устройство (АЦПУ), консоль, устройства подготовки данных и т.п. В одной из версий АОС-ВУЗ в качестве терминалов допускается подключение специализированных пультов системы АТОС Белорусского государственного университета. Для некоторых применений можно использовать составной терминал, состоящий из двух штатных устройств: АЦПУ и устройство перфокарточного ввода. Объем оперативной памяти, необходимой для функционирования базового пакета АОС-ВУЗ, составляет 120 К.

Программные средства АОС-ВУЗ включают системное программное обеспечение ОС ЕС от версии 4.1 и специализированные

прикладные программы, обеспечивающие выполнение АУК и директив пользователей. Пакет АОС-ВУЗ выполняется как обычное задание в одном из разделов ОС ЕС и может совмещаться с другими (фоновыми) работами. Специализированные программные модули подразделяются на транзитные, управляющие, прикладные, вспомогательные, сервисные. К транзитным относятся модули инициализации и завершения работы. Модуль инициализации вызывается при запуске АОС-ВУЗ и выполняет следующие функции: загружает все резидентные модули, открывает рабочие файлы, передает управление модулю управления задачами. Модуль завершения закрывает файлы, выводит статистические данные, передает управление ОС ЕС. К управляющим модулям относятся: модуль управления задачами, модуль управления терминалами, модуль управления файлами. Модуль управления задачами производит синхронизацию выполнения заданий, сбор статистики о ходе и результатах процесса обучения, передачу команд др. управляющим модулям для выполнения операций ввода—вывода, вызов модуля завершения работы системы. Модуль управления терминалами выполняет ввод—вывод данных, команды прикладных модулей и работу с буферными устройствами. Модуль управления файлами выполняет запросы прикладных модулей на доступ к файлам, операции ввода—вывода для файлов подпрограмм и АУК, осуществляет сбор статистики по работе с файлами. Прикладные модули производят подключение и отключение пользователей, обрабатывают директивы всех категорий пользователей, интерпретацию АУК и их редактирование. Вспомогательные модули осуществляют синтаксический анализ директив пользователей и операторов языка описания курсов (ЯОК). Сервисные модули выполняют форматизацию файлов АУК и подпрограмм, загрузку и исключение подпрограмм, печать оглавления файла подпрограмм, распечатку программ АУК, обработку и вывод статистических данных.

В АОС-ВУЗ имеются программные средства конвертирования (переноса) АУК, разработанных в системе СПОК-ВУЗ. Включены средства моделирования; моделирующие программы можно написать на языках ПЛ/1, Фортран, ассемблера языке. Возможен обмен данными между АУК и моделирующими программами. Имеется банк данных сетевого типа, используемый для подготовки ведомостей, сбора и обработки статистики. АОС-ВУЗ поддерживает работу четырех категорий пользователей: диспетчера, автора АУК, преподавателя, обучаемого. К процедурам, выполняемым диспетчером, относятся процедуры, связанные с работой обучаемых; процедуры, относящиеся к работе авторов и обслуживанию курсов; процедуры управления работой системы. Первая группа процедур предназначена для регистрации обучаемых в системе и удаления их после изучения курса, просмотра и изменения данных обучаемых. Процедуры, связанные с работой авторов и библиотекой курсов: регистрация,

копирование, перекомпоновка и удаление курсов, получение распечатки курсов на АЦПУ и др. Процедуры управления работой системы связаны с ежедневной инициализацией АОС-ВУЗ, планированием работы пользователей, получением статистических данных о работе обучаемых с курсами. В задачи автора курса входит разработка сценария АУК, подбор и структуризация учебного материала и заданий, написание программы курса на ЯОК, ввод, редактирование и отладка программы курса. Задача преподавателя - помогать обучаемым во время занятий в дисплейном классе. Обучаемые в процессе работы с курсом могут управлять последовательностью изучения курса, посылать при необходимости сообщения автору курса (эти сообщения помещаются в архив, откуда диспетчер может вывести их на АЦПУ и передать автору).

АОС-ВУЗ позволяет обучаемым работать как индивидуально, так и в условиях класса при групповых занятиях. В любом случае система обеспечивает независимую друг от друга самостоятельную работу студентов в индивидуальном темпе с любым из имеющихся в библиотеке АУК. После завершения сеанса работы с АУК система запоминает информацию о ходе обучения, что дает возможность при следующем подключении к курсу начать работу с той точки, к которой он дошел на предыдущем занятии. Ди.лог обучаемого с системой осуществляется в реальном масштабе времени (см. *Ре: 'м разделения времени*). Время реакции системы не превышает нескольких секунд.

Кроме базового пакета, в семейство АОС-ВУЗ входят АОС-ВУЗ/ОСКАР, АОС-ВУЗ/ПРИМ.УС, АОС-ВУЗ/ФОКУС, АОС-ВУЗ/АТОС, АОС-ВУЗ/АП, АОС-ШКОЛА, АОС-РРВ и др.

О.П.Платонова.

АОС-ВУЗ/АП — автоматизированная обучающая система для вузов, поддерживающая взаимодействие с пользователем посредством удалённых терминалов типа АП-64. Разработана в 1983—84 в Техническом университете Дрездена в сотрудничестве с НИИ высшей школы (Москва) и Московским институтом инженеров гражданской авиации, прошла техническое и педагогическое опробование в университете в 1985—87. АОС-ВУЗ/АП функционирует на ЭВМ семейства ЕС типа ЕС-1022 и последующих. Количество и тип используемых терминалов можно зафиксировать при генерации системы.

Разработчиками АОС-ВУЗ/АП были успешно решены следующие задачи: изменение организации ввода—вывода, связанное с низкой скоростью передачи информации (2000 бит/с) и др. особенно ями удалённых терминалов; адаптация программ вывода информации к особенностям АП-64 (коды и набор знаков, форматизация экрана, управление курсором и т.п.); разработка принципов выбора и выдачи

системных сообщений на языке, выбранном при запуске системы; обеспечение совместимости учебных и программных средств АОС-ВУЗ/АП и др. АОС-ВУЗ в обоих направлениях.

При педагогическом опробовании авторами в Техническом университете Дрездена и Высшей школе архитектуры и строительства Веймара были созданы программы учебного назначения по математике, языкам программирования, разработке программного обеспечения и русскому языку. Авторы, уже владеющие некоторым проблемно-ориентированным языком, легко осваивали язык описания курсов, в т.ч. его концепцию приоритетов различных операторов. Недостатки языка: небольшие размеры экрана (12 x 80 символов), отсутствие графических и псевдографических возможностей, а также недостаточная поддержка "оконной" системы взаимодействия (см. Многооконный интерфейс). Другая отрицательная черта языка - смешение обучающей и управляющей информации в исходных текстах, что уменьшает наглядность и усложняет внесение изменений. В целом авторы оценили АОС-ВУЗ/АП как средство пригодное для разработки оригинальных обучающих программ, реализующих различные дидактические методические цели. Работа студентов различных специальностей с такими обучающими программами показала значительное увеличение интенсивности самообучения благодаря индивидуализации педагогического руководства и постоянной активности обучаемого. Наиболее существенные недостатки АОС-ВУЗ/АП — длительное время реакции и малая надежность АП-64, не позволяющие поддерживать одновременно работу с более чем пятью терминалами такого типа. Однако опыт использования АОС-ВУЗ/АП в качестве исследовательского средства повлиял на развитие концепций автоматизированных обучающих и авторских систем для персональных компьютеров типа ЕС-1834.

Н.М.Когдов, В.Х.Хартвиг.

АОС-ВУЗ/АТОС — пакет прикладных программ, предназначенный для поддержки автоматизированных обучающих систем на базе компьютеров серии ЕС и специального дисплейного комплекса, совместимого с системой учебного телевидения. В качестве монитора терминалов преподавателя и обучаемых используется телевизор типа "Юность". Разработан в Научно-исследовательском институте проблем высшей школы и Научно-исследовательском институте прикладных физических проблем Белорусского государственного университета.

Ю.И.Лобанов.

АОС-ВУЗ/МИКРО — автоматизированная обучающая система для вузов на базе микро-ЭВМ. Предназначена для разработки и использования автоматизированных учебных курсов, разработанных

на языке описания курсов. Система обеспечивает перенос курсов, разработанных на ЕС ЭВМ, в среду микро-ЭВМ ДВМ-3 или компьютеров, совместимых с IBM PC. ЯОК, реализованный в системе, соответствует синтаксису, принятому в базовой системе АОС-ВУЗ для ЕС ЭВМ. Технология переноса курса состоит из следующих этапов: выгрузки курса в символьный файл на ЕС ЭВМ; обработки курса специальной утилитой (вспомогательная системная обрабатывающая программа *операционной системы ЕС ЭВМ, обеспечивающая пользователя различными средствами обращения с наборами данных*) для перевода в формат микро-ЭВМ; обработки исходного файла курса транслятором ЯОК и получения внутренней формы; интерпретации внутренней формы в режиме обучения/контроля с помощью интерпретатора ЯОК.

АОС-ВУЗ/микро может использоваться для разработки курсов на персональной электронной вычислительной машине; при этом программа на ЯОК готовится в виде обычного символьного файла с помощью любого текстового редактора, после чего обрабатывается транслятором ЯОК, обеспечивающим диагностику синтаксических ошибок и получение в результате корректной внутренней формы курса. Интерпретатор ЯОК включает средства интерпретации внутренней формы с обеспечением логики выборки операторов, функциональный отладчик ЯОК-программ, средства сбора статистики о работе системы. Технология разработки мобильного курсового обеспечения (см. *Мобильность программного обеспечения*), поддерживаемого системой, предполагает выполнение ряда требований и ограничений для обеспечения переносимости курса. Некоторые конструкции АОС-ВУЗ для ЕС ЭВМ непереносимы в среду персональных компьютеров, поэтому пакет АОС-ВУЗ/микро можно использовать только для курсов на мобильном варианте ЯОК.

Ю.И.Лобанов, А.Е.Пасхин, А.Н.Филиппов.

АОС-ВУЗ/ОКА — автоматизированная обучающая система для вузов, предназначенная для обучения и тренировки пользователей базой данных ОКА. Объединяет возможности АОС и системы управления базами данных. Включает базовый пакет прикладных программ АОС-ВУЗ с широким набором макросредств и программный интерфейс с СУБД ОКА, который служит для обмена сообщениями между подсистемами. В системе максимально используются возможности базы данных.

АОС-ВУЗ/ОКА реализована как прикладная задача СУБД ОКА и работает под её управлением. Обучение осуществляется с терминалов БД, выполняющей необходимые обеспечивающие функции по управлению системой и передаче сообщений от терминалов СУБД ОКА к АОС-ВУЗ и обратно. Система обеспечивает обмен сообщениями между АОС и СУБД, эффективное использование

процессора, исключение дублирования средств телеобработки. Интерфейс, управляющий потоком информации между подсистемами, размещается в разделе АОС-ВУЗ и дополнительно включает контроль раздела СУБД ОКА. Программы интерфейса написаны на ассемблера языке.

А.В.Брановицкий, Ю.И.Лобанов, Е.П.Седов.

АОС-ВУЗ/ОСКАР — система, предназначенная для разработки обучающих программ и управления их выполнением на базе ЕС ЭВМ в операционных средах ОС ЕС и системы виртуальных машин (АОС-СВМ). Обеспечивает обслуживание в реальном масштабе времени пользователей четырех типов (*автора курса, обучаемого, преподавателя, диспетчера*); разработку тренирующих, обучающих и справочных программ; хранение и выдачу учебной информации обучаемому со скоростью и в объеме, которые зависят от его индивидуальных характеристик; одновременную работу большого количества пользователей; накопление и использование статистической информации о процессе обучения; включение анализатора ответов. Версия АОС-СВМ предназначена для функционирования в операционной среде СВМ. Каждый тип пользователя имеет в СВМ свою виртуальную машину. АОС-ВУЗ/ОСКАР представляет собой многоцелевую систему широкого назначения. Область ее применения - автоматизация обучения в различных областях знаний независимо от предметной области. Языком программирования для разработки учебного материала является язык описания курсов. Система совместима со СПОК-ВУЗ и АОС-ВУЗ.

АОС ВУЗ/ОСКАР позволяет разрабатывать и использовать в учебном процессе все виды обучающих программ. Средства системы выдают учебный материал и вопросы по алгоритму, заложенному в курсе, управляют работой пользователей. Предусмотрены два уровня управления: мультидоступом и файлами, что обеспечивает работу пакета АОС-ВУЗ/ОСКАР как интерактивной системы. В АОС ВУЗ/ОСКАР можно выделить три осн. типа программных компонент: модули управления, управляющие таблицы, модули реализации работы с пользователями. Модули управления предназначены для организации множественного доступа пользователей к программным ресурсам и да им системы, а также обмена данными между дисками и лентами. Управляющие таблицы представляют собой область обмена данными и управляющей информацией между модулями управления и модулями реализации работы с пользователями. Модули управления используют такие средства операционных систем ЕС, как макрокоманды супервизора и управления данными, а модули реализации работы с пользователями — макрокоманды системы. Модуль управления и управляющие таблицы постоянно находятся в оперативной памяти ЭВМ.

А.В.Брановицкий, Ю.И.Лобанов, С.В.Трояновский.

АОС-ВУЗ/ПРИМУС — интегрированный пакет прикладных программ, объединяющий возможности базового пакета АОС-ВУЗ и диалоговой системы коллективного пользования ПРИМУС. Основным назначением пакета прикладных программ (ППП) АОС-ВУЗ/ПРИМУС является создание и сопровождение автоматизированных учебных курсов (АУК) по дисциплинам программирования и др. курсов, в процессе функционирования которых возникает необходимость в выполнении программ моделирования различных изучаемых процессов и явлений, ППП АОС ВУЗ/ПРИМУС обеспечивает обучаемому возможности ввода в режиме диалога учебной программы и данных, синтаксического анализа и корректировки программ, формирование заданий для выполнения в режиме пакетной обработки; предоставляет авторам АУК широкие возможности по управлению работой обучаемого в режиме отладки и счета учебных программ посредством авторских макрокоманд и выполнению в процессе функционирования курса, заранее подготовленных и помещенных в библиотеки программ различного назначения. Пакет построен по модульному принципу, что обеспечивает возможность внесения необходимых изменений и включения новых функциональных возможностей. Подключение функциональных программ, реализующих дополнительные возможности, осуществляется посредством модулей сопряжения, создающих среду ПРИМУС в обучающем курсе. Непосредственная связь модулей сопряжения с АУК обеспечивается подпрограммой-функцией, вызов которой включается в последовательность операторов языка описания курсов либо осуществляется неявно посредством записи в курсе соответствующей макрокоманды. В распоряжение автора предоставляются две макрокоманды: DEBUG — для генерации в составе обучающего курса модуля, имитирующего среду ПРИМУС; CALL — для перевода обучаемого в среду ПРИМУС, является вызывающей по отношению к макрокоманде DEBUG. Эту команду можно записать в тех местах курса, где требуется перевод в режим отладки столько раз, сколько это необходимо. Режим работы обучаемого в среде ПРИМУС задается одним из девяти аргументов макрокоманды CALL и может принимать значения: F — полный режим, обеспечивает обучаемому доступ ко всем командам системы; S — ограниченный режим позволяет осуществить только ввод и корректировку программы; V — произвольный режим, дает возможность автору задать произвольно выбранный набор команд; P — программируемый режим, позволяет автору описать алгоритм процесса работы обучаемого в среде ПРИМУС путем задания фиксированного набора команд для каждого шага работы обучаемого по вводу и отладке программ. Использование макрокоманд пакета АОС-ВУЗ/ПРИМУС иллюстрирует фрагмент АУК, в котором обучаемому предлагается составить программу решения кубического уравнения, ввести ее командой корректировки CORR, запустить на выполнение командой EXEC, сохранить в библиотеке по команде SAVE и

просмотреть результат выполнения командой SOUT. Контроль за выполнением задания осуществляется командами просмотра активных заданий и очередей DA и DN.

QU СОСТАВЬТЕ ПРОГРАММУ РЕШЕНИЯ УРАВНЕНИЯ:

$$X^{**3}-3*X^{**2}-6*X+8=0$$

НА ОТРЕЗКЕ [0,2], ИСПОЛЬЗУЯ МЕТОД КАСАТЕЛЬНЫХ.

СА ГОТОВ

ТУ ВВЕДИТЕ ПРОГРАММУ В ПАМЯТЬ, ЗАПУСТИТЕ НА ВЫПОЛНЕНИЕ,

ПРЕДВАРИТЕЛЬНО СОХРАНИВ ЕЁ В БИБЛИОТЕКЕ, И ПОЛУЧИТЕ НА

ЭКРАНЕ РЕЗУЛЬТАТ.

LB M0

CM /CALL/1/CORR,SAVE,EXEC,П(DA, DN),SOUT/

LB M1

QU НАПЕЧАТАЙТЕ РЕЗУЛЬТАТ СЧЁТА ПО ПРОГРАММЕ.

СА(W) 1 ОДИН ЕДИНИЦА

ТУ СОВЕРШЕННО ВЕРНО.

ПЕРЕХОДИМ К ИЗУЧЕНИЮ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ.

BR СИСТЕМ

СА(W) 4-2

ТУ ЭТО ЧИСЛО ТАКЖЕ ЯВЛЯЕТСЯ КОРНЕМ УРАВНЕНИЯ, НО НЕ ВХОДИТ В ЗАДАННЫЙ ИНТЕРВАЛ.

ВНЕСИТЕ НЕОБХОДИМЫЕ ИЗМЕНЕНИЯ В ПРОГРАММУ И ПРОСЧИТАЙТЕ ЗАДАЧУ ЕЩЁ РАЗ

CM /CALL/1/CORR,SAVE,EXEC,П(DA, DN),SOUT/

BR M1

NX В ВАШЕЙ ПРОГРАММЕ СОДЕРЖАТСЯ ОШИБКИ.

НЕОБХОДИМО ИХ ИСПРАВИТЬ И ПОВТОРНО ВЫПОЛНИТЬ ЗАДАНИЕ.

BR M0

Дополнительно в состав пакета включены средства, упрощающие и ускоряющие работу автора по вводу в систему больших фрагментов курсов, подготовленных во внешних по отношению к АОС-ВУЗ/ПРИМУС наборах данных. АОС-ВУЗ/ПРИМУС представляет собой естественную среду для реализации многофункциональных автоматизированных учебных курсов по дисциплинам программирования, типичными представителями которых являются МФ АУК АФРОДИТА и МФ АУК АИДА.

А.Ф.Лысков.

АОС-ВУЗ/СМ — инструментальная диалоговая программная система автоматизации решения задач, использующая компьютеры системы малых ЭВМ. Является системой широкого применения;

может быть использована для создания подсистем САПР, АСНИ, АСУ. Осн. назначение — автоматизация обучения.

В системе выделяют два уровня базовый и диалоговый. Базовый уровень составляет *операционная система* ДИАМС. Этот уровень предназначен для организации управления ресурсами системы. Диалоговый уровень является надстройкой над базовым и организован в виде диалоговой подсистемы, предназначенной для организации взаимодействия с *пользователями*, осуществляемого с помощью специальных диалоговых программ. Пользователи АОС-ВУЗ/СМ делятся на три категории: *диспетчер*, *автор* абонент — пользователь, решающий совместно с системой к.-л. задачу по определенному курсу (при *автоматизированном обучении* абонентами являются *обучаемые*).

АОС-ВУЗ/СМ обеспечивает: одновременную работу с системой пользователей всех категорий; одновременное выполнение нескольких курсов параллельно с их составлением; составление одного курса несколькими авторами; составление и отладку курсов (см. *Отладка программы*) на специализированном языке описания *диалога*; составление и отладку прикладных программ на *алгоритмическом языке* высокого уровня (см. *Уровень алгоритмического языка*); организацию библиотек курсов и библиотек прикладных программ; организацию и ведение на дисках иерархических *баз данных* (не более 200 М); простое обращение к набору *периферийных устройств*, входящих в номенклатуру технических средств СМ ЭВМ; исполнение курсов, разработанных для базовой системы АОС-ВУЗ и других типовых систем этого семейства. АОС-ВУЗ/СМ дает возможность авторам написать на *языке описания курсов* программу-курс, который может быть контролирующим курсом, обучающим курсом, справочным текстом, *тренажером*, имитирующим взаимодействие абонента и объекта изучения. Для анализа *ответов* абонента в ЯОК используется сравнение с эталонами правильных и предполагаемых неправильных ответов, назначение частичного анализа ответа абонента, предварительное редактирование ответа для сокращения количества анализируемых вариантов, накопленная автором статистика об абоненте. Средства ЯОК позволяют генерировать *тексты* сообщений абоненту и эталоны ответов в процессе исполнения курса.

А.В.Брановицкий, Ю.И.Лобанов.

АОС-ВУЗ/ФОКУС — *автоматизированная обучающая система* для вузов, обеспечивающая связь с диалоговой *системой коллективного пользования* ФОКУС. В командные языки *диспетчера*, *преподавателя* и *автора* системы АОС-ВУЗ введена команда FOCUS, при исполнении которой *терминалы* АОС-ВУЗ этих групп *пользователей* передаются под управление системы ФОКУС. Терминал *обучаемого* переходит под управление системы ФОКУС при выпол-

нении команды FN FOCUS, включенной в язык описания курсов. Выход обучаемого из системы ФОКУС и возвращение в АОС-ВУЗ осуществляется с помощью команды АОС. Оба задания (АОС-ВУЗ и ФОКУС) выполняются параллельно и независимо, связь между ними осуществляется с помощью программы SVC. Пакет прикладных программ АОС ВУЗ/ФОКУС позволяет создавать гибкие системы для обучения программированию ЭВМ. При этом АОС-ВУЗ берет на себя функции обучения, справочного обеспечения и помощи, а система ФОКУС — функции подготовки и корректировки текстов программ и запуска заданий.

А.В.Брановицкий, Ю.И.Лобанов.

АОС-ИСКРА-226 — комплекс программных компонентов специального назначения, используемый для создания, сопровождения и эксплуатации в диалоговом режиме автоматизированных учебных курсов, написанных на языке описания курсов. Реализует функции обслуживания двух категорий пользователей: автора АУК и обучаемого, работающего с программой курса; предоставляет в распоряжение автора АУК средства: первоначальной регистрации курса; закрепления за автором некоторого конкретного курса; корректировки программы курса; получения твердой копии программы курса; перемещения материала курса; удаления курса. Каждое из перечисленных средств реализуется соответствующей процедурой программы АОС-ИСКРА-226.

Процедура первоначальной регистрации курса предназначена для создания на магнитном носителе файла для размещения материала курса. Здесь же формируется информация об авторе курса, объёме курса, необходимом объеме вспомогательной памяти ЭВМ. Процедура закрепления за автором некоторого конкретного курса или выбора курса ведет поиск по имени соответствующего файла на внешнем носителе информации, определяет право пользователя на работу с данным курсом в качестве автора, осуществляет загрузку специальных параметров курса в память, подготавливая таким образом выбранный курс для работы. Процедура корректировки программы курса представляет собой специализированный синтаксически-ориентированный текстовый редактор, с помощью которого автор имеет возможность производить всевозможные изменения в тексте программы курса. Процедура получения твердой копии программы курса позволяет вывести на печатающее устройство как весь текст программы, так и ее фрагмент от метки до метки. Вывод можно осуществить листами, задав требуемое количество строк на листе, а также получить нелистованный текст. Процедура позволяет задать длину строки при выводе. Процедура перемещения материала курса предназначена для копирования курса при необходимости изменения его местоположения. Копирование осуществля-

ется из выбранного процедурой выбора курса во вновь зарегистрированный процедурой регистрации. Копирование можно выполнять и в курсе, в котором имеются *операторы*, отличающиеся от головной и хвостовой меток. При этом материал курса-приемника уничтожается. Копирование позволяет освободить свободные зоны в файле курса, образовавшиеся в результате редактирования. Процедура удаления курса позволяет удалить файл курса с магнитного носителя. При этом можно либо удалить файл из каталога, либо физически удалить его с диска. В первом случае файлу курса в каталоге присваивается статус удаляемого. Во втором случае имя файла удаляется из указателя, информация на диске сжимается и производится настройка указателя каталога.

АОС-ИСКРА-226 предоставляет обучаемому возможность работы с программой обучающего курса в двух режимах: в режиме разового пользователя и в режиме постоянного пользователя. Разовый пользователь работает с курсом только один сеанс; им, напр., может быть *преподаватель*, желающий ознакомиться с функционированием курса. Такой пользователь не регистрируется, для него не создаются специальные файлы. Если обучаемый предполагает работу с курсом в течение нескольких сеансов, он идентифицируется системой как постоянный. Такой пользователь должен назвать имя курса, с которым желает работать, свое имя и шифр, необходимый для создания специального файла, хранящего параметры курса.

С целью обеспечения максимально возможной простоты и гибкости для разработки программы в АОС-ИСКРА-226 принят модульный принцип построения. Это позволяет легко модифицировать ее для включения новых функций. В состав программы входят следующие модули: главный управляющий; управляющий подсистемы автора; регистрации курса; редактирования курса; получения твердой копии курса; копирования материала курса; удаления курса; подсистемы обучаемого. Эти модули реализуют осн. логику работы программы АОС-ИСКРА-226.

Главный управляющий модуль получает управление при запуске программы АОС-ИСКРА-226 и включает: блок описания *данных*, содержащий объявления всех системных переменных и массивов, объединенных в общую область; блок системных *подпрограмм*, реализующих обработку функциональной *клавиатуры* и *операции* со структурами данных; блок инициализации АОС, осуществляющий анализ распределения внешних устройств, считывание и запоминание системных таблиц и т.п.; блок переключения подсистем, управляющий режимом работы пользователя. Выполнив процедуру инициализации АОС, главный управляющий модуль остальное время работает в режиме переключения подсистем, ожидая выбора пользователем режима работы. Управляющий модуль подсистемы автора выводит на экран *главное меню*, содержащее авторские процедуры. При выборе необходимой процедуры управляющий модуль

осуществляет загрузку модуля, реализующего выбранную процедуру, и передает ему управление. Модуль регистрации курса подключается управляющим модулем подсистемы автора. Диалоговая часть модуля запрашивает параметры: количество секторов, отводимых на подматериал курса; имя курса (в формате метки ЯОК); пароль для защиты от несанкционированного доступа (1-8 символы); коэффициент таблицы меток (по умолчанию принимается 23 метки на 50 секторов курса). Диалоговая часть анализирует эти параметры на допустимость и требует, если необходимо, повторного ввода параметров. Исполнительная часть модуля производит расчет требуемого количества дисковой памяти, осуществляет размещение файла курса, заполняет и записывает таблицы системных параметров курса, формирует головную и хвостовую метки курса. Модуль получения твёрдой копии курса реализует процедуру получения *распечатки* материала курса. Диалоговая часть модуля запрашивает параметры: тип распечатки (лист/рулон); начальную метку (по умолчанию принимается головная метка); конечную метку (по умолчанию принимается хвостовая метка); количество символов в строке (по умолчанию 82); количество строк на листе (по умолчанию 60); номер первого листа (по умолчанию 1). Введенные параметры анализируются на допустимость. В процессе печати на экран выводится динамика процесса (количество распечатанных строк, а также сами печатаемые строки), по которой автор может управлять прохождением процесса печати. Модуль копирования материала курса позволяет тиражировать готовые АУК, а также перекомпоновывать курсы в процессе их разработки. Диалоговая часть модуля запрашивает имя и пароль курса приемника. Исполнительная часть осуществляет копирование, контролируемое автором с экрана. В процессе копирования контролируется корректность заполнения таблицы меток и переполнение файла курса-приемника. Особенности механизма обработки файлов данных в режиме каталога в языке *Бэйсик* таковы: память, занимаемая удаленным файлом, физически не освобождается, и имя удаленного файла сохраняется в указателе каталога и не может быть повторно использовано. Удаление курса в АОС-ИСКРА-226 реализуется в двух режимах - стандартными средствами языка *Бэйсик*; физическое удаление и освобождение памяти, занимаемой курсом, и удаление имени файла из указателя каталога. Диалоговая часть модуля запрашивает режим удаления, и после получения подтверждения исполнительная часть выполняет требуемые действия. Модуль подсистемы обучаемого реализует функции по интерпретации АУК и обеспечивает работу обучаемого с требуемым курсом. Включает блоки: регистрации и идентификации обучаемого; создания и ведения файла обучаемого; интерпретации; ввода и обработки сообщений обучаемого.

Программа АОС-ИСКРА-226 применяется в среде технических и программных средств ПЭВМ ИСКРА-226. Условия ее функциони-

рования определяются как областью применения, так и наличием технических средств.

М.А.Краснов.

АОС-М — *автоматизированная обучающая система*, предназначенная для *мини-ЭВМ*, при реализации которой используют методы достижения *мобильности программных систем*. Разработана в начале 80-х гг. в Институте кибернетики АН Украины. Система позволяет применить мини-ЭВМ для *обучения и контроля знаний* обучаемых, работающих на машине одновременно по индивидуальным *программам*; использовать ЭВМ для подготовки, тренировки и консультации операторов и *пользователей системного программного обеспечения* мини-ЭВМ и *пакетов прикладных программ*; сократить время и повысить качество подготовки *обучаемых* различных категорий (учащихся, студентов, курсантов, операторов различных машин и систем, пользователей ЭВМ); обеспечить *авторов* учебных курсов и *администрацию* необходимой *информацией* о ходе учебного процесса. В АОС-М обеспечен комплексный подход к управлению подготовкой специалистов за счет реализации режимов: *обучаемого* (для индивидуальной работы обучаемого с *автоматизированным учебным курсом*); *автора* (для работы автора над АУК — написания, редактирования и отладки их); *диспетчера* (для поддержания функций функционирования системы, контроля доступа пользователей к учебным курсам, ведения библиотек учебных курсов, сбора и обработки статистических данных о ходе процесса обучения).

Основу АОС-М составляют: *язык описания курсов* и его *интерпретатор*; библиотеки АУК в текстовом и компилированном виде; библиотеки функций, расширяющих возможности ЯОК (функции могут быть написаны как на *ассемблера языке*, так и на языках высокого уровня); *база данных*, хранящая сведения о пользователях системы и статистическую информацию о ходе обучения. АОС-М предназначена для организации процесса *автоматизированного обучения* на базе *системы малых ЭВМ* в *операционных системах* ДООС КП, РАФООС и ОС РВ.

Для организации работы АОС-М необходимы следующие технические средства: ЭВМ СМ-4 или СМ-1420 с объемом *памяти ЭВМ* не менее 64К, а для работы в оптимальном режиме (8 *терминалов*) — 256 К; не менее одного съемного дисковод; не менее одного накопителя на магнитной ленте; устройство вывода на печать; *терминальные устройства* — *дисплеи* для работы с СМ ЭВМ. По сравнению с другими АОС АОС-М обладает *мобильной реализацией* (путем незначительных усилий систему можно адаптировать на любую другую мини-ЭВМ и *микро-ЭВМ*); простым, наглядным и доступным языком описания учебных курсов (который совместим с языком систем *СПОК-ВУЗ* и *АОС-ВУЗ*, что дает возможность

использовать АУК, разработанные в этих системах); развитыми сервисными и справочными средствами, позволяющими авторам легко и эффективно создавать и отлаживать учебные курсы; надежной защитой системы от неправомерного доступа - средствами, обеспечивающими секретность, целостность и непротиворечивость БД и библиотек учебных курсов системы.

В.А.Петрушин.

АОС-М/ПРАВЕЦ — комплекс программных средств, обеспечивающих создание и эксплуатацию программ учебного назначения на базе персональных электронных вычислительных машин Правец-16 и других, совместимых с ПЭВМ типа IBM PC. Использование системы как инструмента в рамках компьютерной технологии обучения дает возможность разрабатывать высококачественные программы учебного назначения (ПУН) по различным учебным дисциплинам; отлаживать разрабатываемые программы; осуществлять сопровождение уже созданных программ; эффективно выполнять программы в ходе занятий. Программы для системы АОСМ —М/Правец разрабатываются на входном языке, использование которого не требует высокой квалификации в области программирования. Данный язык позволяет автору ПУН использовать все возможности системы, в т.ч. графический многооконный интерфейс для воспроизведения текстовой, графической и смешанной (текстово-графической) информации; развитую подсистему определения и воспроизведения меню, позволяющую осуществлять многопозиционную селекцию; подсистему анализа ответов по меню. Осн. компоненты системы АОС-М/Правец—компилятор входного языка и абстрактная обучающая машина (АОМ).

Компилятор входного языка преобразует исходный текст ПУН в мобильный промежуточный код, являющийся входным языком АОМ и получивший название *i*-кода. ПУН, представленная в *i*-коде, может быть немедленно интерпретирована. Однако при этом возможно определенное снижение скорости исполнения, возникают ограничения на размеры программы, в связи с чем режим непосредственной интерпретации *i*-кода используется как отладочный. Для повышения скорости исполнения ПУН и снятия ограничений на их размеры предусмотрен режим интерпретации подпрограммного шитого кода. В этом случае представление отлаженной программы в *i*-коде преобразуется в ее представление в подпрограммном шитом коде.

В.И.Отенко.

АОС-РРВ — автоматизированная обучающая система для режима разделения времени — пакет прикладных программ (ППП), предназначенный для создания, сопровождения и эксплуатации обучающих, контролирующих, справочных и обслуживающих при-

ладных диалоговых программ (ПДП), в которых допускается применение внешних (по отношению к АОС) программ произвольного объема и сложности. Позволяет организовать не только обучение, но и тренировку навыков работы с различными ППП и средствами операционной системы, а также может применяться для создания диалоговых интерфейсов, обеспечивающих для неквалифицированных пользователей доступ и взаимодействие с различными программными средствами. АОС-РРВ разработана на базе пакетов АОС-ВУЗ, АОС-ВУЗ/ОСКАР и системы коллективного пользования СРВ. Поддерживает технологию создания и эксплуатации ПДП, характерную для АОС семейства СПОК (см. СПОК-ВУЗ). Обеспечивает функционирование в режиме диалога трех осн. категорий пользователей — диспетчера, автора, обучаемого. Сохранены с миним. изменениями и дополнениями доступные им языки директив. Осн. инструментальным средством программирования ПДП является язык описания курсов. Таким образом обеспечена преемственность с пакетами типа АОС-ВУЗ, что позволяет разработчикам ПДП переходить к работе в среде АОС-РРВ в кратчайшие сроки.

Набор команд СРВ дополнен командами вызова АОС. С их помощью пользователи АОС-РРВ, работающие в среде СРВ, могут при необходимости воспользоваться имеющимися ПДП. Реализовано совмещение самостоятельной работы обучаемого с получением справки или помощи, что не удается достичь в большинстве АОС. Важнейшие отличительные особенности АОС-РРВ, повышающие эффективность обслуживания пользователей, состоят в следующем. Во-первых, все функции по обеспечению мультидоступа выполняются системными программами СРВ. Для разделения времени процессора применяется алгоритм квантования, а не циклический алгоритм обработки запросов, характерный для АОС-ВУЗ. Такое изменение дисциплины обслуживания позволяет поддерживать высокую реактивность системы в диалоге независимо от сложности и длительности обработки запросов других пользователей, работающих в это же время. Разделение наборов данных и прочих ресурсов системы также осуществляется средствами СРВ. Во-вторых, программы, обеспечивающие функционирование пользователей каждой категории, выделены в автономные подсистемы, находящиеся под управлением монитора СРВ наравне с др. обработчиками команд системы разделения времени. Такое расчленение структуры программных модулей АОС целесообразно, поскольку в СРВ программы, обслуживающие каждый терминал, трактуются как отдельные задания. При сохранении общей структуры программ каждый пользователь получил бы в свое распоряжение все средства АОС, тогда как сам он использует всегда лишь их часть, связанную с обслуживанием пользователей данного типа. Если, напр., за терминалом работает обучаемый, то программы, обслуживающие автора и диспетчера, не применяются, хотя и занимают оперативную память ЭВМ. В результате выделения

автономных функциональных подсистем в АОС-РРВ была повышена взаимная независимость пользователей. Общим ресурсом являются лишь наборы данных, содержащие ПДП, в остальном пользователи не влияют друг на друга. В-третьих, расширены системные средства АОС, что позволяет осуществлять двухстороннюю передачу управления и данных между ПДП и внешними программами или ПДП. Это достигнуто прежде всего за счет создания специальной системной программы связи, обеспечивающей взаимодействие между ПДП и языком команд СРВ, а также путем устранения осн. ограничений на программы-функции, характерных для АОС-ВУЗ. Удобным сервисным средством в АОСРРВ является протоколирование сеансов работы или их частей. Запуск процесса протоколирования и его прекращение выполняются либо по команде пользователя, либо автоматически в начале и конце сеанса соответственно.

АОС-РРВ разработан в Институте кибернетики им.В.М.Глушкова АН Украины и рекомендуется к использованию на моделях ЕС ЭВМ с объемом оперативной памяти не менее 1М в среде операционной системы ОС ЕС с режимом разделения времени.

М.В.Легкий.

АОС-ШКОЛА — автоматизированная обучающая система для средней школы, предназначенная для обучения основам программирования на одном из языков программирования и работе на устройствах ЕС ЭВМ в качестве оператора. Обучение проводится в две стадии. На первой стадии осуществляется *профорентация с помощью компьютера*. С этой целью в среде системы необходимо выявить для каждого обучаемого наиболее адекватные типы деятельности, напр.: “человек — знак”, “человек — машина”, “человек — природа” и т.п. В связи с этим в АОС-Ш. реализованы *процедуры тестирования обучаемых, демонстрационные модели объектов и понятий предметной области*, избираемой для деятельности. На второй стадии осуществляется *автоматизированное обучение* в процессе решения содержательно понятных учебных задач. Для профорientации обучаемых в области вычислительной техники разработана тестовая профорientационная система ПРОФОР, реализованная средствами системы управления базами данных ИНЕС. Профорientационная среда имеет иерархическую структуру и представляется в виде дерева (рис.).

Поддерево с вершиной “человек” определяет анкету обучаемого, заполняемую при регистрации (первом подключении к системе). Поддерево “классификация” определяет структуру избираемой деятельности. Классификация представляет собой композицию значений вершин: типы, классы, отделы, группы (табл.1).

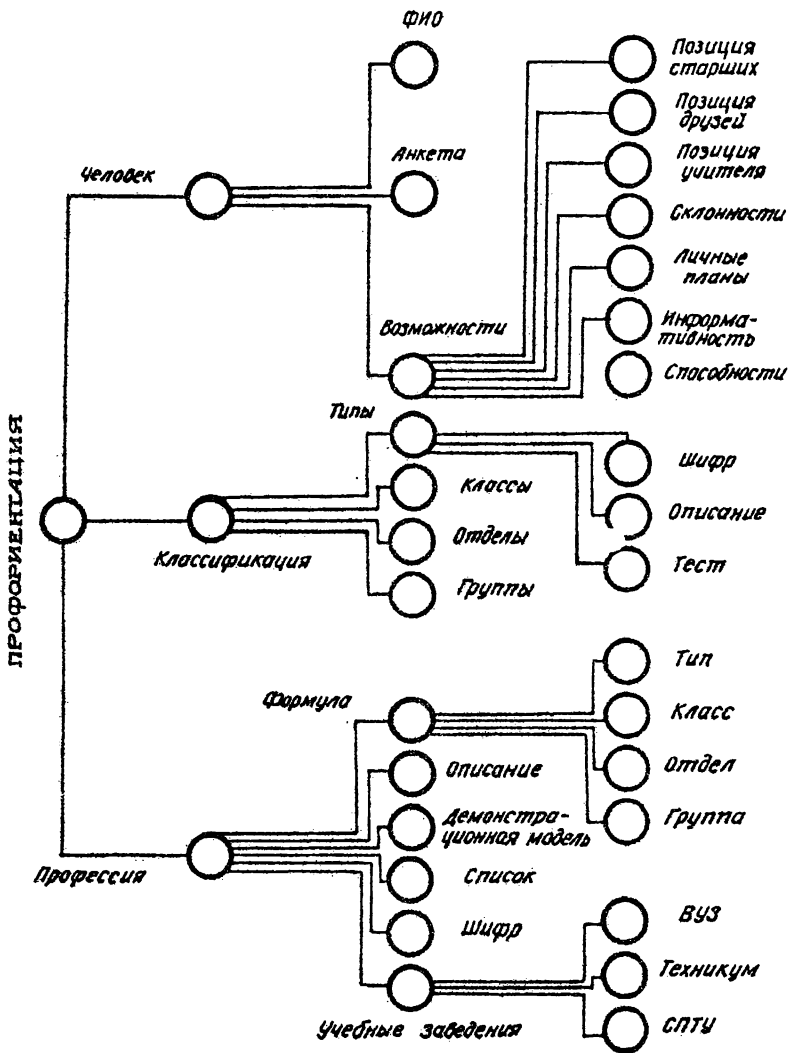


Таблица 1.

Вершины

Тип	Класс	Отдел	Группа
Природа	Гностические	Ручной труд	Бытовой микроклимат
Техника Человек	Преобразующие	Машинно-ручной труд	Открытый воздух
Знак	Изыскательные	Использование автоматизированных систем	Необычные условия
Художественный образ		С преобладанием функциональных орудий труда	Моральная ответственность

Профотбор по специализации “Программирование и вычислительная техника” определяется следующими значениями композиции вершин:

Тип	Класс	Отдел	Группа
<техника> <знак>	<гностические> <преобразующие>	Использование автоматизированных систем	<необычные условия> <моральная ответственность>

Демонстрация и автоматизированное обучение организуются в интегрированной среде ПРОЛОГ/АОС-ШКОЛА. *Автоматизированный учебный курс* подкрепляется учебным пособием с более широким перечнем учебных задач. В среде обучающего диалога обучаемый знакомится с базовыми *понятиями* курса для использования этих понятий при решении учебных задач. При недостаточном четком понимании данного понятия обучаемым *диалоговая система* (ДС) отсылает его к учебному пособию, где приводятся соответствующие примеры. Интегрированная среда ПРОЛОГ/АОС-ШКОЛА позволяет унифицировать лингвистическое обеспечение диалога. Для этого на базе системы ПРОЛОГ-ЕС был реализован специальный препроцессор, учитывающий лексическую окраску решаемых в среде ДС учебных задач по программированию. Это дает возможность сместить акцент обучения на выработку у обучаемых определенного типа мышления при овладении ими навыками программирования. Структурно препроцессор можно разделить на две части — словарную и управляющую. Словарная реализована в виде базисного *тезауруса*, отражающего семантику фраз из рассуждений каждого обучаемого на начальном этапе автоматизированного обучения. Для этого были

учтены типы *данных*, составляющие содержание учебных задач, после чего выделены соответствующие фразы, используемые обучаемыми при описании рассуждений о *способах решения* задачи. Эти фразы были занесены в словарную часть препроцессора. В его среде реализованы также процедуры пополнения словарной базы. Этим целям служит таблица синонимов, которую обучаемый может заполнить на начальной стадии взаимодействия. Управляющая часть выполняет две функции. Во-первых, в ней фиксируются логические связи объектов решаемых учебных задач. Во-вторых, вырабатывается семантический код, устанавливающий соответствие между описаниями рассуждений пользователя в начальной и изучаемой им алгоритмической нотациях. В качестве семантического кода при автоматизированном обучении программированию был выбран язык линейного описания *алгоритмов*.

Взаимодействие обучаемого с АОС-III. основывается на специальных режимах, каждый из которых обеспечивает определенный вид деятельности. Ядро системы составляет АОС *Афродита*. Ответы и сообщения обучаемого делятся на три группы. К первой группе относятся фразы, занесенные в словарную базу препроцессора, ко второй — слова ДА, НЕТ, НЕ ЗНАЮ, к третьей — конструкции операторного языка и конструкции изучаемого ЯГ (табл.2).

Таблица 2

Фортран	Элементы операторного языка
READ FORMAT	R
PRINT FORMAT	P
IF GO TO	L
IF <метка>, <метка>, <метка>	A
ПРИСВАИВАНИЕ	V, W, F
STOP ... END	S

Схема компьютер—книга позволила расширить информационную структуру автоматизированного обучения, поскольку АУК подкрепляется учебным пособием с более широким перечнем учебных задач. В среде обучающего диалога обучаемый знакомится с базовыми понятиями курса (*множеством элементов опорных понятий*) и учится использовать эти понятия при решении конкретных задач. Однако при недостаточно четком понимании пользователем данного понятия система отсылает его к учебному пособию, где приводятся соответствующие примеры и аналогичные задачи. При таком подходе схема компьютер—книга носит ярко выраженный учебный характер. В среде АОС реализуется обучение правилам описания конкретных типов данных языка *Фортран*, а в учебном пособии приводится детальное изложение учебного материала с примерами. Фиксируя

действия, которые обучаемый применяет к объектам учебной задачи, компьютерная учебная среда осуществляет контроль за правильностью их использования. При необходимости эта среда отсылает обучаемого к пособию.

Э.Л.Ивахненко, А.Е.Стрижак.

ARITY/PROLOG — интерпретатор языка *Пролог*, содержащий примитивные предикаты для обработки информации в текстовом и графическом режимах, имеющихся в микрокомпьютерах типа IBM PC; разработан фирмой Arity (отделение фирмы Lotus Advanced Development Group). Графическое расширение ARITY/PROLOG (APG) — одна из версий языка Пролог.

Возможности APG можно разделить на 9 групп предикатов, выполняющих следующие действия: 1) переход от одного режима к другому (текстовой, графической, причем разрешается работать с двумя текстовыми и двумя графическими режимами); 2) действия над текстовыми экранами: определение и использование активизированных экранных страниц, контроль над формой и положением курсора, разработка цветов *символов*; 3) определение цветов на графических экранах: фон, фронт, палитра и цвета на рисунках; 4) стандартное рисование: определение точки отсчета, рисование точек, прямых линий (непрерывные и штриховые), прямоугольники, окружности и эллипсы (с заполнением или без него), рисование ломаных линий, чтение атрибута любой точки на экране; 5) запись и чтение рисунков: возможности импорта страниц, записанных с помощью других языков или *графических редакторов*, различные графические эффекты, в т.ч. движение; 6) использование различных наборов символов (шрифты): имеется 14 шрифтов, реализуемых одновременно (стандартные, подчеркнутые, маленькие, готические, греческие, матем. символы и др.); есть возможность определить новые шрифты; 7) применение окон и изменения масштаба; 8) рисование графиков матем. функций (в определенном месте экрана, в неопределенной области и др.); 9) завершение работы с APG восстановлением в регистрах стандартных значений работы для возврата в определенную систему.

APG предназначен главным образом для реализации графического интерфейса в экспертных обучающих системах, используемых, в частности, для обучения высшей математике и программированию.

Ф.Мартинес, М.Прието.

АРХИТЕКТУРА ЭВМ — абстрактный комплекс внутренних программ, микропрограмм и технических средств *электронной вычислительной машины*. Определяет принципы организации машины и функции процессора, не отражая способов управления и передачи данных внутри процессора, конструктивных и техно-

логических особенностей логических элементов и т.д. Включает наиболее общую структуру ЭВМ (процессор, память ЭВМ, порты устройства ввода—вывода информации) и структуру внутреннего программного обеспечения ЭВМ. Важнейший принцип, определяющий А. ЭВМ, состоит в гибком, автоматическом управлении машиной с помощью программы, хранимой в памяти ЭВМ. Из этого принципа вытекают важные следствия: вычислительная машина является универсальным устройством, которое позволяет решать широкий класс задач и обрабатывать разнообразные данные; переход от решения одной задачи к выполнению другой осуществляется путем смены программ и соответствующих им данных; программы и данные хранятся в запоминающем устройстве; процессор может обращаться к хранимым в памяти ЭВМ различным объектам, не делая между ними различия, и таким способом может перерабатывать программу, модифицировать ее. Осн. цель модификации — повысить производительность ЭВМ, т.е. в конечном счете сократить время решения задач. Быстродействие всякой системы определяется скоростью работы самого медленного ее элемента. Чтобы повысить скорость, стремятся заставить медленные устройства ЭВМ работать быстрее и найти такие конфигурации устройств ЭВМ, которые уменьшают влияние медленных устройств на быстродействие машины в целом. Одно из архитектурных решений в этом направлении — распараллеливание процессов в ЭВМ. Возможности архитектурных решений неисчерпаемы, и с каждым новым поколением ЭВМ они постоянно расширяются.

В.П.Козлова, В.Л.Леонтьев, З.Ш.Примакова.

АСЕМБЛЕРА ЯЗЫК (от англ. assemble — собирать, монтировать) — машинно-ориентированный язык программирования, структура и взаимосвязи операторов которого близки к уровню языка машинных команд конкретной ЭВМ. Позволяет пользоваться mnemonicскими обозначениями команд, присваивать символические имена регистрам и элементам памяти ЭВМ, использовать различные способы адресации, различные системы счисления. Содержит операторы, соответствующие машинным командам, и др. средства, предназначенные для распределения памяти, сегментации программ, работы с макрокомандами (макроассемблер). А. я. используется для создания системных программ, а также прикладных программ с повышенными требованиями к быстродействию или эффективности использования ресурсов машины.

Для изучения общих принципов работы с различными языками ассемблера, а также для обучения конкретным А.я. созданы автоматизированные обучающие системы и курсы. Напр., многофункциональный автоматизированный учебный курс АИДА для начального обучения программированию на А.я. единой системы ЕС

ЭВМ, разработанный в Кишиневском государственном университете, обеспечивает и поддерживает обучение, контроль, тренировку и самостоятельную работу пользователей в режимах изучения принципов функционирования и архитектуры ЭВМ, обучения конструированию алгоритмов, диалогового программирования на А.я., отладки программ в среде операционной системы ЕС ЭВМ.

В.А.Бардадым.

АССОЦИАТИВНОЕ КОДИРОВАНИЕ — кодирование, при котором каждому элементу данных присваивается уникальный код. Примером ассоциативного кода может служить код телефонного набора устройства, где каждому номеру соответствует один абонент. А.к. и трансформационное кодирование — два осн. принципа получения кодов из полного описания данных. Ассоциативный код непосредственно не связан с полным описанием элемента данных и выбирается произвольно. Недостатком А.к. является плохая (по сравнению с трансформационным кодированием) читабельность кодов (см. *Читабельность текста*). Код необходимо либо выучить, либо каждый раз справляться о нем по сводной таблице. А.к. целесообразно пользоваться для представления значительных массивов информации, когда экономия памяти компьютера и стоимость коммуникаций могут быть большими, чем потери, вызванные увеличением времени распознавания и количества ошибок.

И.Ю.Никонова.

АСТЕНОПИЯ (от греч *ασθενής* — слабый и *ὄψ* — зрение) — зрительное расстройство, возникающее в результате длительной напряженной работы. Сопровождается характерными жалобами на зрительный дискомфорт. Различают А. аккомодативную, мышечную, ретиальную, цветовую и другие. В основе аккомодативной А. лежит утомление цилиарной мышцы (см. *Аккомодация*), участвующей в изменении кривизны хрусталика и при длительной работе на близком расстоянии испытывающей чрезмерное напряжение. Аккомодативная А. выражается расплывчатостью читаемых или рассматриваемых знаков, давлением в глазницах, болью в надбровной области, головной болью. Мышечная А. чаще развивается при нек. рригированных аметропиях, высокой степени миопии, врожденной слабости внутренней прямой мышцы и др. Проявляется раздвоением изображения, болями в глазах и голове. Ретиальная А. характеризуется расплывчатостью изображения и отражает процессы утомления, возникающие в сетчатой (светочувствительной) оболочке глаза. Цветовая А., выражающаяся в наличии хроматопсии (неестественное окрашивание рассматриваемых предметов), является следствием утомления цветоощущающего аппарата глаза. При работе с видеотерминалом чаще всего проявляются мышечная

и ретинальная А. В соответствии с классификацией Всемирной организации здравоохранения мышечную А. принято называть окулярной, а ретинальную А. — визуальной.

Меры борьбы с А: оптимальный режим работы с видеотерминалом; правильная организация рабочего места и помещения, обеспечивающая оптимальные условия работы зрительного анализатора; соблюдение регламентированного времени труда и отдыха. Режим работы за экраном устанавливается в зависимости от целей учебного занятия. Для предотвращения развития А. необходимо проводить профосмотры с целью выявления и своевременной коррекции миопии.

А.С.Коваленко, В.Г.Мартирсова.

АСТИГМАТИЗМ (от греч. α. — префикс, означающий отрицание и *στίγμα*(*στίγματος*) — точка), асферическая рефракция глаза — сочетание в одном глазу разных видов рефракции или различных степеней одного вида рефракции. При работе с видеотерминалом особенно важна коррекция А. с учетом расстояния наблюдения.

В.Г.Мартирсова.

ATLAS II — двуязычная система машинного перевода, использующая в качестве лингвистической стратегии “язык-посредник”. Предназначена для обработки японо-английской политехнической документации. Лексическая база системы представляет собой автоматические словари до 50 тыс. словарных статей для каждой языковой пары. ATLAS II ориентирована на семантическое представление текста и тезаурусы — словари значений слов, содержащихся в автоматических словарях. Семантическое представление текста базируется на семантической сети — концептуальной структуре исходного предложения. Система разработана японской фирмой FIDZUCU и реализована на компьютере FACOMM.

К.П.Пиотровская.

AUTOOL (англ. Authoring Tool — инструмент автора) — универсальная открытая авторская система, автоматизирующая этапы детализированного проектирования и программирования программ учебного назначения. Позволяет создавать линейные и разветвленные ПУН демонстрационного, информационно-справочного, вопросно-ответного, обучающе-контролирующего и др. типов. Разработана на основе системы PLATO фирмы Control Data специально для использования системы в национальной сети видеотекстов ВТХ и реализована в филиале фирмы Control Data во Франкфурте. Работает на MUPID ВТХ-компьютерах, на IBM PC и совместимых с ними компьютерах с помощью MUPID-карт, а также

EGA-карт. С помощью системы А. можно создавать ПУН, называемые в системе уроками, по любой дисциплине на уровне меню. Курсы могут храниться на *дискетах*, в локальной или национальной системе видеотекстов.

Компоненты системы: две версии *интерпретатора* — дисковая и видеотекстовая — для вывода материала на *дисплей*, *анализа ответов обучаемого* и управления ходом *обучения*; редактор уроков — для создания и редактирования уроков; интерпретатор тестов — версия интерпретатора для *автора*; *программа сжатия* — для подготовки урока к хранению его в сети видеотекстов; программа сортировки — для реорганизации и создания оптимального порядка фрагментов урока; программа анализа *ответов* — для быстрого тестирования сложных *моделей* ответов; конвертор из первой во вторую версию системы; средства печати содержимого урока, оценки размера урока и т. д. Урок состоит из множества фрагментов — логических единиц обучения, которые хранятся как отдельные страницы на диске со своим именем. В уроке может быть до 99 фрагментов. Фрагмент состоит из множества объектов (треугольник, линия, блок текста, пауза по времени, анализ ответа и др.) и/или структур (группа объектов с групповым именем). Фрагмент может содержать информационный материал (*текст*, графические структуры, *мультипликацию*) и *вопросы*.

В системе имеются такие типы фрагментов: нормальный обучающий (содержит осн. часть обучающего материала); итоговый (последний фрагмент в уроке, где подводятся итоги его выполнения); индексный (обеспечивает возможность ветвления к различным фрагментам урока; при выполнении урока *обучаемый* выбирает одну из определенных здесь ветвей); фрагмент вопроса (нормальный обучающий фрагмент, содержащий вопрос); графический (содержит графическую и текстовую *информацию*, выводится на экран интерпретатором перед нормальным фрагментом, содержащим ссылку на него); фрагмент помощи (дополнительные объяснения обучаемому, которые выдаются на экран по его *запросу*; ссылки на этот фрагмент обычно содержат фрагменты вопросов); внешние фрагменты (подключают в урок информацию, которая была создана не в системе; в качестве внешнего фрагмента можно вызвать ВТХ-страницу, созданную с помощью СЕРТ редактора MUPID, динамически переопределяемые множества *символов*, Бейсик-программы, фрагменты структур созданные в др. уроке). Порядок представления фрагментов обучаемому в ходе выполнения урока устанавливается автором с помощью указателей. Нормальные фрагменты всегда содержат указатели на следующий и предыдущий фрагменты. Графические фрагменты не содержат указателей, т.к. всегда связаны с другими фрагментами. Фрагмент помощи может содержать указатель на предыдущий фрагмент, указатель на следующий фрагмент в нем не определяется,

поскольку управление после выполнения фрагмента помощи всегда передается вызывающему его фрагменту.

Редактор системы основан на меню. Он позволяет создавать новые и редактировать старые фрагменты, объекты и структуры, копировать их, удалять, сохранять на диске. Из редактора автор может послать команды системе MUPID и вернуться в редактор, если при этом не вызывается другая программа. *Операции* редактирования выполняются на четырех уровнях, соответствующих уровням структуры данных системы — объектов, структур, фрагментов и уроков. В системе определяются следующие объекты: цвет (переднего плана — текста и графики, фона текста и фона всего экрана), текст, ряд графических объектов (прямоугольник, окружность, сектор, сегмент, дуга и т.д.), стирание экрана, пауза (по времени и по нажатию клавиши), индекс (определение альтернатив выбора следующего фрагмента), тон (звуковое предупреждение), повтор (определение числа повторений и повторяющейся группы объектов), динамически переопределяемые множества символов, вызов ВТХ-страницы, объект движения, объект анализа ответа, объект вызова Бейсик-программ. Объект движения позволяет мультиплицировать текстовые или графические объекты. Движение может быть линейным, по определенному пути и вращением (вокруг своей собственной либо внешней точки). Возможны комбинации этих видов движения. Между шагами движения может устанавливаться задержка, объект при движении может изменяться в размерах, зеркально отображаться, оставлять или не оставлять след; каждый шаг движения при необходимости сопровождается звуком.

Система А. дает возможность обрабатывать следующие типы *ответов*: множественный выбор (до 16 вариантов); текстовый ответ в виде слова, фразы или предложения; группа текстовых ответов (до 16) в виде заполняемых форм; числовые ответы. Автор может определять такие типы контроля: специальные ответы; ответ верен (неверен) с первой попытки; ответ верен (неверен) с повторных попыток; ответ частично верен с первой попытки; ответ частично верен с повторных попыток. Для каждого случая автор по запросу системы определяет ветвление и текст сообщения, если оно требуется. При обработке ответа обучаемого можно выделять необязательные и *ключевые слова*, группу слов произвольного порядка, запрещенные слова, слово или группу слов, допускающих синонимы, границы правильного числового ответа. По желанию автора могут разрешаться или запрещаться орфографические ошибки в ответе обучаемого.

В системе есть два типа курсора — текстовый, передвигающийся по одному символу, и графический, который может передвигаться по *символу* и по одной координатной позиции экрана. Видимый объект помещается в то место экрана, где находится курсор. При редактировании фрагментов реализованы следующие операции над объектами: **ВСТАВИТЬ**, **УДАЛИТЬ**, **МОДИФИЦИРОВАТЬ**, **КОПИРОВАТЬ**. Видимые объекты ищут по их местоположению на экране.

Структура может храниться как отдельный внешний фрагмент на диске и вызываться по имени из к.-л. фрагмента любого урока. Это позволяет создавать библиотеки конструкций общего пользования. В системе есть два типа структур — внешние и внутренние. Объекты внешней структуры выводятся на экран в точке вызова, однако недоступны для оперирования с ними, т.к. физически не располагаются в вызывающем их фрагменте. При вызове во фрагмент внутренней структуры его объекты остаются в точке вызова и над ними можно выполнять операции.

Выполнение урока начинается с первого фрагмента, не являющегося графическим или фрагментом помощи. В конце выполнения урока автор заканчивает процесс обучения, либо повторяет выполнение того же урока или переходит к изучению другого. После нормального обучающего фрагмента можно перейти к следующему фрагменту, вернуться к предыдущему или вызвать фрагмент помощи. Из индексного фрагмента возможен переход к одному из указанных в таблице фрагментов, в конец урока или возвращение к предыдущему фрагменту. Если включена пауза, то продолжить выполнение урока можно нажатием специальных клавиш или по истечению указанного автором времени. Нажатием клавиш также прерывается вывод повторяющейся последовательности объектов. После фрагмента вопроса реакция системы зависит от типа вопроса и ответа обучаемого. Обучаемый имеет определенные возможности управления ходом выполнения урока. Напр., он может отмечать некоторый фрагмент, а затем вызывать выполнение этого или другого фрагмента, номер которого он может указать.

Интерпретатор тестов расширяет возможности автора при выполнении уроков по сравнению с возможностями обучаемого в режиме интерпретации. Так, автор может выполнить часть урока, указывая номера первого и последнего выполняемого фрагментов, игнорировать паузы, выполнять или не выполнять функции графического заполнения, выводить на экран все модели ответов и текущие значения счетчиков и переключателей состояния.

Программа анализа ответов дает возможность автору быстро тестировать определяемые им модели ответов без переходов из редактора в интерпретатор и обратно. Здесь автор, работая в синтаксисе редактора, может выполнять тот же анализ, что и в интерпретаторе. С помощью этой программы автор задает модель ответа, проверяет ее синтаксис, вводит различные варианты ответов, анализирует их относительно заданной модели и получает сообщения системы.

Т.А.Лисюк.

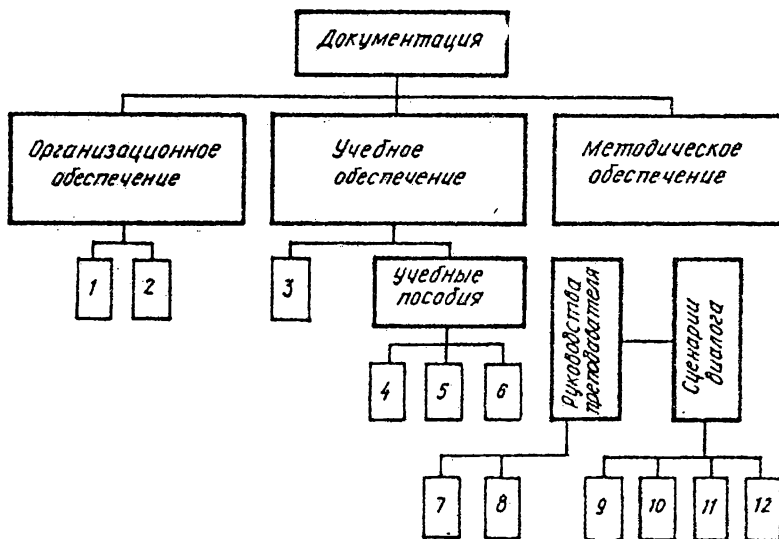
АФРОДИТА (Автоматизированный Фортран — Режим Обучающего Диалога с Терминалом Автора) — *многофункциональный автоматизированный учебный курс*, предназначенный для обучения программированию на основе языка *Фортран*. При обучении в курсе выделяются следующие разделы предметного содержания и связанные с ними дидактические цели: изучение осн. принципов обработки информации на компьютере; усвоение способов конструирования алгоритмов для решения различных классов задач; изучение языка программирования, формирование умений и навыков по описанию алгоритмов на этом языке; формирование навыков обнаружения и исправления ошибок в исходной программе — освоение процесса отладки программ. АФРОДИТА представляет возможность использования гибкой технологии обучения, основанной на сочетании традиционных методов обучения, применении, по усмотрению преподавателя, языкового и алгоритмического подхода для усвоения предметного содержания обучения. Программное обеспечение организовано в виде самостоятельных подсистем, взаимодействующих между собой через управляющую программу. Каждый структурный компонент реализует определённую логическую часть предметного содержания и определённый метод обучения.

В структуру системы входят режимы: АЛГОРИТМ — обучение приёмам составления и способам описания алгоритмов решения задач с помощью компьютера; ДИПРОФОР — диалоговое программирование задач на языке Фортран под управлением компьютера если обучаемый недостаточно знаком с языком программирования; ПРОГРАММА — диалоговое программирование задач на Фортране, если обучаемый хорошо знаком с языком программирования; ФОРТРАН — автоматизированное обучение языку Фортран; ОТЛАДКА — диалоговая отладка программ на языке операционной системы. Режимы работы курса полностью охватывают предметное содержание и методы обучения программированию: АЛГОРИТМ, ДИПРОФОР и ПРОГРАММА обеспечивают алгоритмический метод обучения, ФОРТРАН — языковый метод, а ОТЛАДКА обеспечивает развитие практических навыков программирования на языке Фортран. Управляющая программа организует взаимодействие обучающих режимов и осуществляет: приём и анализ команд языка взаимодействия при выборе режима работы; переключение на режим работы, соответствующий введённой директиве; наблюдение состояния обучаемого при смене режимов работы; оказание помощи при выборе режима работы. Управляющая программа получает управление всякий раз, когда обучаемый либо подключается к системе, либо завершает работу в одном из режимов, либо вводит специальную команду для перехода в др. режим работы. Режим АЛГОРИТМ представляет собой обучающе-контролирующую программу, реализует один из компонентов алгоритмического подхода к обучению и не

ориентирован на конкретный язык программирования. Режим ДИПРОФОР по заданной обучаемым блоксхеме алгоритма совместно с ЭВМ конструирует программу решения задачи. Этот режим используется на начальном этапе подготовки, когда психологически важно показать новичкам широкие возможности компьютера. Режим ПРОГРАММА также реализует алгоритмический подход, однако по сравнению с режимом ДИПРОФОР имеет ограниченные обучающие функции и используется на начальных этапах обучения, когда уже освоено миним. подмножество Фортрана. Режим ФОРТРАН реализует языковой подход и предназначен для детального изучения конструкций языка программирования Фортран-IV. Режим ОТЛАДКА предоставляет в распоряжение *пользователя* набор отладочных средств. Обучающие функции этого режима обеспечиваются возможностью запроса помощи на любом этапе отладки и управлением действиями пользователя со стороны компьютера.

Основу языка взаимодействия составляют команды: смены режима — РЕЖИМ, ВЕРНУТЬ; выбора режима — АЛГОРИТМ, ДИПРОФОР, ПРОГРАММА, ФОРТРАН и ОТЛАДКА; запроса консультации — ПОМОГИ; команды режима — ОТЛАДКА. Кроме этих команд, обучаемый использует язык операторного описания алгоритмов и свободно конструируемые ответы на Фортране. Группа команд смены режима осуществляет изменение последовательности прохождения учебного материала. Команда РЕЖИМ позволяет сменить режим работы с курсом; введя эту команду, обучаемый может начать работу в любом другом режиме. Команда ВЕРНУТЬ используется для возврата в режим, который был предыдущим по отношению к работающему. Возврат осуществляется в точку прерванного *диалога* предыдущего режима. Группа команд выбора режима работы предназначена для подключения обучаемого к соответствующему режиму. Команды этой группы могут вводиться либо при первоначальном выборе режима, либо при желании сменить режим работы с помощью команды РЕЖИМ. Команда запроса консультации интерпретируется трехуровневой выдачей информации: первый уровень — разъяснение смысла *вопроса* или указания, выданного курсом; второй уровень — элементарная подсказка или наводящий вопрос; третий уровень — подробное разъяснение ответа, который необходимо дать для продолжения работы. Уровень выдачи *информации* зависит от того, сколько раз обучаемый вводил команду ПОМОГИ на некоторые сообщения курса. Кроме программного обеспечения, составной частью курса является развитая система документации (рис.) — организационное, учебное и методическое обеспечение.

Курс АФРОДИТА можно применять в учебно-производственных комбинатах общеобразовательных средних школ, профессионально-



технических училищах, высших учебных заведениях, организациях и на факультетах повышения квалификации.

М.А.Краснов.

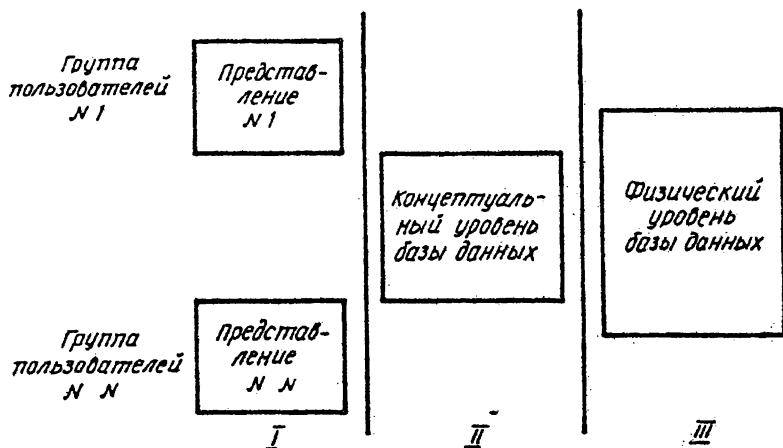
BUGGY — экспертная обучающая система для диагностики ошибок обучаемых при выполнении арифметических операций (сложение и вычитание) многозначных чисел. Разработана в 1976-77 в США. Центральное место в системе занимает модель обучаемого, включающая как правильные, так и неправильные элементарные действия, производимые при выполнении арифм. операции. Каждая арифм. операция представляется в виде процедурной сети, состоящей из элементарных действий (напр., перенос единицы старшего разряда, сложение или вычитание однозначных чисел и др.). Если все действия в сети правильные, то операция выполняется правильно. Если некоторые из правильных действий заменить на ошибочные, сеть будет представлять собой модель ошибочного поведения обучаемого. При диагностике ошибок обучаемого система модифицирует процедурную сеть, заменяя правильные действия ошибочными, и тем самым пытается построить модель обучаемого, объясняющую все его ответы. Для вычитания имеется 70 элементарных ошибочных действий. Если замена одного действия в модели

не приводит к успеху, делается попытка построить модель с двумя ошибочными действиями. В основу анализа ошибок в BUGGY положено представление о том, что систематические ошибки при выполнении арифм. действий являются результатом правильного исполнения неправильных процедур. Следовательно, в качестве модели обучаемого не может служить подмножество знаний эксперта, т.к. ни одно такое подмножество не содержит ошибочных процедур, применяемых обучаемым.

Осн. назначение системы BUGGY — дать преподавателям опыт формирования гипотез о соотношениях между симптомами ошибок обучаемого и теми неправильными представлениями, которые их вызывают. В этом случае BUGGY играет роль обучаемого, модель поведения которого содержит ошибки, а преподаватель по примерам выполнения арифм. операций должен диагностировать ошибки обучаемого. Кроме того, систем. может выступать в роли преподавателя, предлагая обучаемому различные задания и устанавливая причину ошибок в их выполнении. Задания подбираются так, чтобы в кратчайший срок однозначно установить причину ошибки.

В.А.Петрушин, Е.М.Синица.

БАЗА ДАННЫХ (БД; от греч. *βασις* — основание) — именованная совокупность данных, отображающая состояние объектов и их отношений в рассматриваемой предметной области. БД можно рассматривать на различных уровнях абстрагирования, выбираемых с определенными целевыми назначениями. Напр., для группы пользователей N% 1, не являющихся специалистами в области данных, выбирается уровень абстрагирования N% 1 (рис.). Для



выполнения качественного проектирования структур данных в БД — другой уровень — концептуальный. Для решения задач рациональной организации БД на *запоминающих устройствах* — третий — физический. БД обеспечивает централизованное создание, обновление и наращивание данных. Одной из черт БД является независимость данных от прикладных программ, которые их используют, а также возможность создания этих программ в такой форме, что изменение особенностей хранения, логической структуры или значений данных не требует изменения программ их обработки.

В соответствии с предложениями исследовательской группы по *системам управления базами данных* Национального института стандартов (США) архитектура БД разделена на три осн.уровня: внутренний, внешний и концептуальный. Внутренний уровень определяет способы хранения данных на физическом носителе информации. Внешний уровень отражает взгляд того или иного конкретного пользователя на информационное содержание БД. Концептуальный уровень является представлением полного информационного содержания БД в абстрактной форме. Такое представление данных часто наз. *концептуальной моделью базы данных*. Во многих случаях в понятие *концептуальной модели* включают не только общее описание БД, но и схем модификации данных во времени, проверку паролей защиты, контроль данных и т.п. Концептуальная схема должна быть разработана так, чтобы, сохраняя стабильность структур данных, обеспечить возможность расширения БД для других приложений. Составление концептуальной схемы и контроль за обслуживанием структур БД возложено на *администратора базы данных*.

Создание БД — новый шаг в развитии средств обработки данных. Их функционирование будет содействовать дальнейшему расширению области применений компьютеров, более эффективному использованию данных. Для этого БД должна отвечать определенным требованиям к организации данных и обладать свойствами, обеспечивающими ряд преимуществ по сравнению с применением обычных *файлов* данных. Требования к организации БД : физическая независимость данных (способы физического хранения могут изменяться без изменения прикладных программ) ; логическая независимость данных (добавление новых элементов данных или расширение логических структур не требует изменения прикладных программ) ; миним. избыточность данных ; многократное использование данных ; простота использования ; наличие защиты от несанкционированного доступа ; контроль за целостностью данных; обеспечение необходимой скорости доступа к данным ; наличие *интерфейса* высокого уровня для связи с программистом ; наличие средств восстановления данных и средств протоколирования работы с БД (системный журнал) ; наличие средств отображения данных между различными уровнями представления данных; наличие высо-

коуровневых процедурных средств манипулирования и обработки данных, ориентированных на функционального пользователя системы. Осн. теоретическим аппаратом создания и использования БД является аппарат *моделей данных*.

А.П.Ильяшенко, М.Е.Козлов.

БА́ЗА ЗНА́НИЙ (БЗ) — организованная часть *систем, основанных на представлении знаний*, в которой содержатся утверждения и эвристические правила планирования и решения задач. Утверждения в БЗ представляют *информацию*, которая может изменяться, напр., в ходе консультации. Правила представляют информацию о том, как выводить новые утверждения из того, что сейчас известно (см. *Метазнания*). БЗ в отличие от *базы данных* (БД) обладает большей содержательностью и динамичностью. Факты в БД обычно пассивны: они либо есть, либо их нет. БЗ активно пытается пополнить недостающую информацию. Однако *знания* эти не воплощены в какую-то программу, а представляют собой *данные* для подсистемы вывода системы, основанной на представлении знаний. Знания в БЗ можно представить как в декларативном, так и в процедурном виде (см. *Декларативное представление знаний, Процедурное представление знаний*). Во многих случаях возникает необходимость в обоих типах представления информации (см. Турбопролог).

Использование знаний и БЗ в *обучении* приобретает особую роль в силу нескольких причин. Во-первых, для большинства проблем из области *искусственного интеллекта* (напр., для организации процесса обучения в интеллектуальных *обучающих системах*) нет четких алгоритмических решений, поскольку многие важные *задачи* возникают в сложных контекстах явлений, не поддающихся точному и строгому описанию. Во-вторых, специалисты в области обучения благодаря своим познаниям достигают исключительно высоких результатов, и если компьютерные *экспертные обучающие системы* смогут аккумулировать и применять знания экспертов, они также достигнут достаточно высокого уровня в работе. Экспертные знания позволяют на раннем этапе решения задачи выделить необходимые данные и помогают заранее избежать неоправданных попыток, как можно раньше отсекая тупиковые ветви. В-третьих, в реальном мире специалисты имеют дело с неполными или зашумленными данными. Чтобы реализовать потенциальную возможности системы, основанной на представлении знаний, приходиться к верным решениям, опираясь в своей работе на такого рода данные, в БЗ систем вводят избыточность знаний, что дает возможность системе прийти к правильному заключению. В-четвертых, знания — дефицитный ресурс общества. Извлечение знаний у специалистов и придание им формы, позволяющей использовать их в компьютерных системах,

может существенно удешевить воспроизводство знаний и их применение.

Появление обучающих систем с БЗ было обусловлено рядом трудностей, связанных с широким внедрением *автоматизированного обучения*: недостаточным уровнем интеллектуализации обучающих систем; трудоемкостью разработки *моделей* деятельности *обучаемого* и *преподавателя* в процессе обучения; сложностью реализации анализа *ответов* и сообщений *обучаемого*; необходимостью детального описания *алгоритмов* решаемых задач обучения; сложностью автоматизации трудоемкого процесса подготовки и ввода в систему учебного материала; ограниченными возможностями традиционных обучающих систем по *адаптации* к уровню подготовки и личностным характеристикам *обучаемого* и т.д. *Индивидуализация обучения* путем ввода *модели обучаемого*, различного рода *учебных задач*, набора *стратегий* и *методик* обучения и т.д. в значительной степени способствовала интеллектуализации обучающих систем, практическая реализация которой привела к необходимости использования БЗ в интеллектуальных обучающих системах, таких, напр., как САКИО, BETS, интеллектуальные *диалоговые системы*.

Самое узкое место в развитии и применении БЗ — усвоение знаний компьютерной системой. Остро ощущается необходимость автоматизации процесса *приобретения знаний*. Это привело к появлению нового поколения обучающих систем, первым представителем которого можно считать систему EURISCO. См. также *База знаний инструментальной системы, База знаний учебного назначения*.

А.А.Сахно.

БАЗА ЗНАНИЙ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ — совокупность *знаний* о системе (ее назначении, режимах функционирования, типах решаемых задач и методах их решения). Должна содержать: знания инструментального средства, разделяющиеся на синтаксические (синтаксис языка *программирования*) и семантические (семантика *операторов* языка программирования); системные знания, обеспечивающие реализацию режимов функционирования инструментальной системы и включающие словарь *процедур* (управляющих директив) инструментального языка, их синтаксис и семантику; лингвистические знания, включающие *словарь* и *грамматику* лингвистического процессора, предназначенного для обеспечения функционирования естественноязыковой информационно-справочной подсистемы и диалогового *интерфейса*, синтаксис языка директив инструментальной системы; учебные знания — *модель обучаемого* и *модель процесса обучения*, учебный материал (задачи, *тексты*, в т.ч. и знания инструментального средства); знания прикладной задачи

зависящие от конкретной предметной области, включают модель предметной области (ПО), шаблоны представления знаний для заполнения по прикладной задаче; знания диалогового конструктора программ включающие знания инструментального средства, модель пользователя, модели диалогового конструирования программ для определенных классов задач (напр., представленных в терминах объектов, состояний и операций).

Для функционирования Б.з.и.с. в рамках компьютерной технологии обучения можно использовать информационно-справочные системы с элементами обучения, имеющие средства представления и обработки знаний различного уровня: автоматизированные обучающие системы (АОС) с развитыми средствами представления и обработки знаний; экспертные обучающие системы, имеющие знания по предмету, методике обучения этим знаниям и диагностике ошибок; лингвистические процессоры. Средства представления знаний будут реализованы при наличии соответствующих инструментальных средств. Напр., при реализации знаний АОС можно использовать пакет прикладных программ АОС-ВУЗ, АВС-1, АВС-2; информационно-справочных систем — ПРОЛОГ-АДАБАС, естественно-языкового интерфейса — ЛИПС. В зависимости от того, какие средства реализации знаний используются в инструментальной системе, выделяют осн. методы манипулирования знаниями: выбор релевантной информации, обеспечиваемый модульной структурой базы знаний; поиск по образцу, встроенный в язык логического программирования; доказательство теорем в логике предикатов первого порядка. Средства расширения библиотеки встроенных процедур позволяют использовать методы процедурного представления знаний. Прагматический подход к разработке Б.з.и.с. должен предусматривать возможность эффективного использования как отдельных компонентов системы, так и любого их сочетания.

В.В.Колос, С.П.Кудрявцева.

БАЗА ЗНАНИЙ УЧЕБНОГО НАЗНАЧЕНИЯ — знания об учебных задачах, обучаемом, процессе обучения, предмете изучения. Предназначена для реализации педагогических функций (учение, обучение, контроль знаний в решателе); сложна по своей структуре и содержит базу знаний (БЗ) по предметной области, БЗ управления обучением, БЗ об обучаемом (модель обучаемого), БЗ ошибок, БЗ учебных задач.

БЗ по ПО содержит знания о конкретных объектах, отношениях между объектами изучаемого предмета (см. также Модель предметной области). При построении БЗ по ПО при решении задачи обучения необходимо учитывать динамику ПО. Структура БЗ управления обучением и ее наполнение определяются целями, которые ставятся перед процессом обучения. Напр., при обучении

языкам программирования можно выделить следующие осн. цели: обучение методам построения программ, обучение синтаксису и обучению семантике ЯП. Достижение каждой из этих целей требует разработки прежде всего модели процесса обучения, которая должна отражать тип диалога, структуру задачи, стратегии и методики обучения. Использование различных стратегий и методик в зависимости от модели обучаемого дает возможность реализовать индивидуальность в обучении. Основой индивидуального обучения является соответствие содержания обучения возможностям обучаемого, уровню его профессионального и психологического развития. Можно предложить следующие классы стратегий использования в обучающих системах: 1) отражающие уровень профессионального мастерства (способный, посредственный, новичок, квалифицированный); 2) исследовательское обучение (обучение в ходе познания своей деятельности); 3) открытое обучение (обучение через контакт с действительностью); 4) индуктивное обучение; 5) обучение до получения результата. Эти классы стратегий можно реализовать через такую методику: 1) соответствие метода изложения возрасту обучаемых, их профессиональному мастерству, особенностям учебного материала; 2) определение содержательного центра урока и представление системы производных знаний и умений; 3) различное представление обучаемому учебного материала; 4) организация работы обучаемого; 5) использование обучающих воздействий, создающих благоприятные психоэмоциональные условия для обучения; 6) различный контроль усвоения знаний. Модель обучаемого должна удовлетворять следующим требованиям: валидности (учитывать индивидуальные особенности обучаемого); адекватности (обеспечивать соответствие модели обучаемого его оригиналу с точки зрения индивидуальных особенностей); динамичности (обеспечивать корректировку модели обучаемого в процессе обучения). Можно использовать, напр., четырехуровневую модель обучаемого, как это сделано в Сакио. БЗ ошибок содержит таксономию ошибок. Ошибки обучаемого можно использовать для диагностирования недостающих понятий и процессов в БЗ. В БЗ ошибок содержатся возможные отклонения от правильного результата, которые может допустить обучаемый по сравнению со специалистом. БЗ учебных задач содержит типы учебных задач и сам учебный материал по предмету обучения. Типы учебных задач соответствуют распределению задач по трудностям, по времени выполнения учебных заданий, по уровню индивидуализации, по целям, на достижение которых рассчитано решение задачи. Выбор учебных задач должен обеспечивать реализацию учебной деятельности. Учебная задача должна не просто демонстрировать деятельность, а воспроизводить ситуацию обучения при раскрытии содержания учебного материала. В учебной задаче необходимо сфокусировать содержание, метод обучения, теоретическое представление об учебной и обучающей деятельности. При

построении БЗ учебных задач нужно рассматривать не одну отдельную задачу, а систему задач, которая используется для достижения не только ближайших, но и отдаленных учебных целей. Учебные задачи должны обеспечить усвоение системы средств, необходимой и достаточной для успешного осуществления учебной деятельности, причем задачи необходимо подбирать так, чтобы соответствующие средства для достижения цели выступали как прямой продукт обучения (см. *Продукты деятельности*).

С.П.Кудряцева.

БАЙТ (англ. byte) — единица количества информации в двоичном представлении, равная восьми битам. Размер Б. сложился исторически как результат технического компромисса между гибкостью и эффективностью реализации оперативного доступа к участкам памяти ЭВМ произвольной длины в связи с необходимостью обработки двоичных, восьмиричных, шестнадцатиричных, литерных и числовых данных при разной зависимости от особенностей прикладных задач, точности представления чисел. Различают Б. информационный, содержащий данные; контрольный, используемый для контроля сохранности блока данных; Б. состояния, содержащий информацию о состоянии устройства ввода—вывода для программы обслуживания устройств и Б. блока.

В.Н.Семихов.

БАНК ДАННЫХ — система, предоставляющая услуги по хранению и поиску данных для определенной группы пользователей и по определенной тематике (цены на товары, социальная статистика, биологические виды и т.п). Не обязательно открыта для широкого использования, хотя предполагается, что пользователи такой системы есть по всей стране. Доступ к информации может осуществляться через к.-л. сетевую службу (напр., через службу видеотекста, предоставляющую пользователю текстовую и графическую информацию на терминал компьютера, установленного дома) или даже по почте. Сами данные могут быть организованы как база данных или в виде одного или нескольких файлов данных, подключаемых сетевой службой к конкретной сети.

А.П. Ильяшенко, М.Е. Козлов.

БЕЙСИК (от англ. B(eginners) A(II-purpose) S(ymbolic) I(nstruction) C(ode) — многоцелевой символический код для начинающих) — язык программирования высокого уровня (см. *Уровень алгоритмического языка*), ориентированный на диалоговый режим разработки программ. Создан в 1963 в США как средство, позволяющее обучаемым с миним. навыками составлять и отлаживать программы. Б. широко распространен в связи с появлением персо-

нальных электронных вычислительных машин (ПЭВМ); практически все фирмы-изготовители ПЭВМ стали включать его в состав программного обеспечения как стандартный компонент, внося тем не менее свои модификации. Это обусловило появление целого спектра диалектов Б. (многие из которых взаимно несовместимы), отличающихся друг от друга как синтаксисом отдельных конструкций, так и их семантикой. В различных диалектах Б. имеются векторные и матричные операции, средства обработки строк символов, работы с файлами, графического вывода, воспроизведения звука, прямого доступа к памяти ЭВМ по адресам. Как правило, Б. реализуется в виде среды, содержащей интерпретатор, средства редактирования и отладки программ, однако распространены и компиляторы этого языка. В настоящее время Б. широко используется при изучении программирования для выполнения практических упражнений, а также при написании различных прикладных программ, в т.ч. обучающих (автоматизированных учебных курсов, учебных сред, обучающих игр) и при решении небольших прикладных задач (в качестве мощного калькулятора). Для изучения Б. созданы многочисленные обучающие программы.

Осн. преимущества Б. — простота изучения и применения операторов языка, удобство работы с интерпретатором, однако применение его для изучения методов программирования и разработки сложных программ ограничивается тем, что многие распространенные диалекты Б. не имеют развитых средств структурирования программ и данных, не позволяют создавать сложные структуры данных, использовать локальные переменные и параметры процедур.

В.А.Бардадым, В.И.Отенко.

BESS (англ. Bayes Expert Systems Shell — байесовская инструментальная экспертная система) — инструментальная экспертная система (ЭС), основанная на байесовском методе принятия статистических решений. Предназначена для создания диагностических и прогнозирующих ЭС. Разработана в 1988-89 в Институте кибернетики имени В.М.Глушкова АН Украины. В системе принята следующая организация базы знаний: каждой гипотезе ставится в соответствие априорная (допытная) вероятность наличия данной гипотезы, число симптомов, которые могут подтвердить либо опровергнуть данную гипотезу, текст диагностики, а также симптомы, относящиеся к этой гипотезе. Каждому симптому в некоторой гипотезе ставится в соответствие имя симптома, вероятность подтверждения данным симптомом гипотезы и вероятность того, что симптом отвергает гипотезу. Поскольку информация о значениях симптома запрашивается у пользователя, каждому симптому сопоставляется вопрос и диапазон возможных ответов. Допускаются три типа ответов: альтернативные (да,нет,не знаю);

лингвистические (напр., очень низкий, низкий, средний, высокий, очень высокий); целочисленные, предполагающие выбор ответа из интервала [-n, ..., +n]. Процесс взаимодействия системы с пользователем протоколируется. Протокол является основой для функционирования подсистемы обоснования.

В машине вывода системы BESS применяется стратегия, сущность которой заключается в приписывании каждому симптому цены, отражающей его роль в процессе вывода, и задании в первую очередь того вопроса, для которого цена оказывается наибольшей. В зависимости от ответа пользователя вероятности гипотез пересчитываются, а сами гипотезы классифицируются. Вывод продолжается до тех пор, пока не будет найдена гипотеза, апостериорная (последопытная) вероятность которой удовлетворяет некоторым граничным условиям, или пока не будут рассмотрены все симптомы. Пользователь системы имеет возможность в процессе вывода либо после его завершения узнать, как система пришла к тому или иному решению, обратившись к подсистеме обоснования с помощью одной из директив “как” или “почему” При выборе директивы “как” пользователю выдается весь протокол от начала и до того шага вывода, после которого произошло обращение к подсистеме обоснования. При выборе директивы “почему” пользователю объясняется предыдущий шаг вывода.

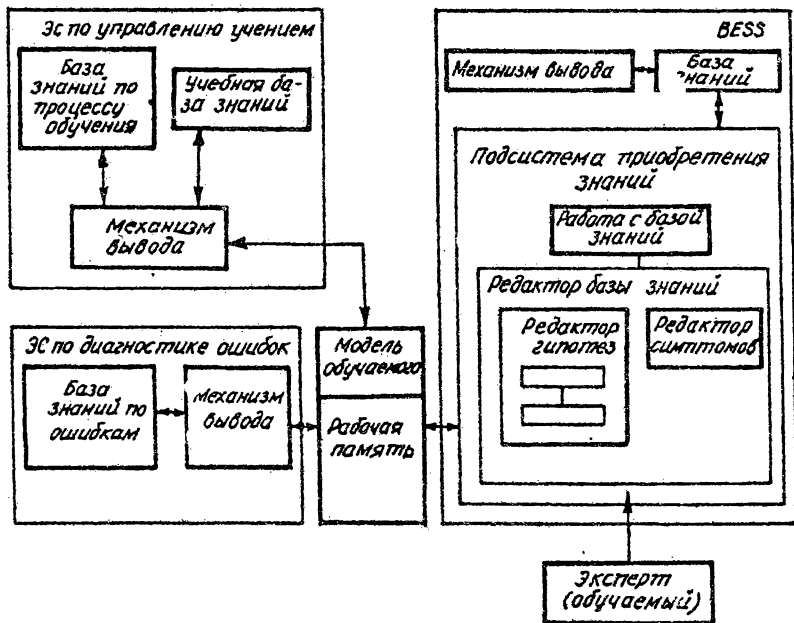
Взаимодействие с экспертом происходит посредством подсистемы *приобретения знаний*. Эта подсистема представляет собой иерархическую совокупность экранных редакторов, позволяющих эксперту создавать, хранить, удалять, редактировать базу знаний на уровне гипотез, симптомов и значений симптомов гипотезы. Диалог с пользователем ведется посредством меню, форм, шаблонов. Необходимость *программирования* в системе BESS отсутствует.

В обучении систему BESS применяют при создании *адаптивных обучающих программ* для тестирования и *диагностики знаний* обучаемых по различным дисциплинам. Кроме того, она является *удобной учебной средой* для освоения осн. понятий, связанных с построением диагностических ЭС. Для пользователей BESS разработана экспериментальная экспертная обучающая система *BETS*, позволяющая сделать более эффективным процесс взаимодействия слабо подготовленного пользователя с системой.

О.В.Ибрагимов, В.А.Петрушин.

BETS (от англ. Expert Tutoring System on Bayes method's application - экспертная обучающая система с применением байесовского метода) — экспериментальная *экспертная обучающая система*, разработанная в 1989 в Институте кибернетики имени В.М.Глушкова АН Украины. Предназначена для *обучения* эксперта формализации своих знаний байесовским методом, а также использования системы

BESS (рис.). Процесс проектирования экспертной системы (ЭС) с помощью BESS рассматривается как процесс совместного человеко-машинного решения задач и состоит в создании и отладке базы знаний (БЗ). При создании БЗ эксперту приходится решать две



задачи: формирование/уточнение гипотез и формирование/уточнение симптомов. Первая задача распадается на две подзадачи: формирование/уточнение гипотезы и устранение избыточности и противоречий между гипотезами. Вторая задача предполагает выбор полного избыточного набора симптомов. Деятельность эксперта по построению ЭС основана на наличии у него определенных теоретических и практических знаний и умений, причем задачи, находящиеся на более низком уровне в иерархии решаемых задач, требуют более глубоких знаний и умений, чем задачи, находящиеся на более высоком уровне.

В системе поддерживается метод педагогического наблюдения. Эксперту предоставляется возможность выбрать и решить произвольную задачу из иерархии решаемых задач. Вмешательство системы в деятельность эксперта происходит лишь тогда, когда при решении

задачи у него обнаруживается недостаток определенных знаний и умений. В этом случае эксперт рассматривается системой как *обучаемый* и происходит перевычисление его модели. *Модель обучаемого* определяет уровень его знаний и умений. Знания обучаемого подразделяются на знания, которыми он должен обладать к началу решения задачи, и знания, которые он может получить в процессе решения задачи. Навыки, или умения, обучаемого в решении задачи определяются типом допускаемых ошибок и числом их повторений. Априорные (доопытные) оценки знаний и умений обучаемого извлекаются из истории взаимодействия обучаемого с системой. Если таких оценок нет, т.е. обучаемый не решал соответствующей задачи, априорно предполагается, что этими оценками являются “знает” и “умеет”.

Планирование процесса обучения производится соответствующей ЭС на основании текущей модели обучаемого, истории его взаимодействия с системой и заключается в выборе задачи, дидактической цели и способа ее достижения. В основе процесса планирования лежит следующий *алгоритм*: производится анализ текущей модели обучаемого, т.е. определяется решаемая задача и тип затруднения; выдвигается гипотеза о причине затруднения (забыл, не знает, не понял, ошибся случайно), на основании которой происходит выбор задачи, дидактической цели и способа ее достижения. Учебная БЗ представляет собой *сеть фреймов*, каждый из которых содержит опр деление *понятия*, набор примеров, раскрывающих его смысл, связь с другими понятиями, типичные ошибки обучаемых, примеры, демонстрирующие значимость этих ошибок. Учебная *информация* выбирается из учебной БЗ в соответствии с историей взаимодействия обучаемого с системой, выбранной дидактической целью и способом ее достижения. БЗ об ошибках обучаемого состоит из описания классов ошибок, возможных при формализации *экспертных знаний* байесовским методом. Каждой решаемой задаче соответствует свой класс ошибок. Все ошибки выявлены и классифицированы заранее.

О.В.Ибрагимов.

БИТ [от англ. bi(nary digit) — двоичная цифра] — единица количества *информации*, соответствующая информации, содержащейся в одном из двух возможных равновероятных сообщений, принимающих значение “0” или “1”; один разряд двоичного числа. Последовательность из 8Б. образует 1 *байт*.

В.Н.Семихов.

БЛЁСКОСТЬ — свойство чрезмерно интенсивных световых раздражений отрицательно сказываться на зрении. Если в поле зрения на фоне относительно равномерной *яркости*, к которой адаптирован глаз, появляется светящаяся поверхность с яркостью, значительно

превышающей фоновую, то установившийся режим работы глаза нарушается. Б., блики и отражения, возникающие при работе с видеотерминалом, зависят прежде всего от фактуры поверхности экрана. Желательно, чтобы поверхность экрана была матовой (безотражательной), а коэффициент отражения поверхности экрана не превышал 0,3. Для устранения бликов рекомендуется использовать специальные покрытия и ограждения экрана. Вопрос борьбы с бликами и отражениями в значительной степени связан с рациональной организацией рабочего места, оснащенного видеотерминалом. В поле зрения пользователя не должно быть избыточных источников естественного (окон) и искусственного освещения.

В.М.Бондаровская.

БРИГАДНАЯ ОРГАНИЗАЦИЯ РАБОТ в программировании. Профессиональная разработка программных средств требует, как правило, коллективной деятельности многих специалистов, организации бригад программистов, а иногда и групп, объединяющих такие бригады. Можно выделить три осн. причины, обуславливающие необходимость Б.о.р. 1) Некоторые задачи весьма трудоемки и для их решения необходимо решить множество подзадач. Время, которое требуется для этого одному человеку, несравнимо с реальными сроками работы (речь может идти о нескольких десятилетиях). 2) Поставленную задачу можно разбить на ряд сравнительно простых подзадач и решать их параллельно — кооперативным способом. Это также позволяет резко (в число раз, сравнимое с количеством параллельно ведущихся разработок) сократить время решения задачи. 3) Решение некоторых задач требует кооперации участников, обладающих специальными знаниями. Эта причина становится все более весомой по мере того, как программирование получает признание в качестве метода решения различных задач; специализация вытекает из этого аспекта программирования, поскольку на приобретение специальных знаний программисту может потребоваться слишком много времени (возможно, больше времени всей его жизни).

Каждый этап решения задач с помощью компьютеров порождает особые проблемы и выдвигает особые требования к организации коллективной деятельности в бригадах программистов. Так, при постановке задач на первый план выступает взаимодействие между постановщиками и исполнителями: прежде чем техническое задание (ТЗ) приобретает окончательный вид, в процессе его уточнения может осуществляться несколько итерационных циклов "постановщик-программист". Здесь осн. задачи связаны с выработкой языка общения, взаимопонимания. Для этого предлагается, в частности, всегда давать четкую письменную формулировку ТЗ (даже если заказчик и исполнитель работают в одной организации). Письменно сформулированное ТЗ уменьшает вероятность переопределения задач

(см. *Доопределение задачи*), облегчает обсуждение, оценку и согласованную с заказчиком коррекцию отдельных требований в ходе разработок. На этапах *проектирования* и собственно программирования организация коллективной деятельности опосредствована формальными отношениями в том подразделении, которое выполняет разработки, а также особенностями поставленной задачи (прежде всего возможностью ее разбивки на отдельные модули).

В процессе *отладки программ* особое значение приобретают различные способы организации совместного тестирования. Как правило, совместная отладка является более эффективной, т.к. индивидуальные программисты владеют различными методами отладки. Поскольку выявление ошибок наиболее экономично на ранних стадиях работы, совместные проверки нередко осуществляются на каждой стадии разработки программного продукта — при формулировании целей (уточнении ТЗ), проектировании (архитектура программы, спецификации) и непосредственно при *кодировании*. На практике для этого организуются специальные бригады проверяющих, проводятся структурные просмотры проектов и *программ* по заранее намеченным критериям, осуществляется формальное техническое рецензирование, а также используются различные специальные методы для сравнения альтернатив при выборе аппаратного и *программного обеспечения*.

Кроме кратковременных совместных работ, при выполнении конкретных заданий практикуются и долговременные *стратегии* организации бригад программистов для решения сложных задач. Определяющими факторами здесь являются имеющиеся материально-технические средства, количество исполнителей и их подготовленность, а также особенности поставленной задачи. Существуют три осн. типа таких бригад. Первый определяется иерархизированными формальными отношениями в бригаде. Есть, как правило, старший программист (руководитель) и подчиненные ему младшие программисты. Задания распределяются вышестоящими по иерархии. Второй тип бригады, основанный на функционально-ролевом принципе, получил название "бригады главного программиста". Здесь каждый выполняет определенную роль. Главный программист пишет наиболее важные части программы, в то время как другие ему "ассистируют". Преимущества подобной структуры — более высокая специализация каждого программиста, большие возможности для *обучения* на примере деятельности главного программиста, участие высококвалифицированных специалистов в непосредственной разработке программ. Недостатками такой структуры являются ее авторитарность, полная зависимость от решения главного программиста, сложность обеспечения полной нагрузки и равноправия членов бригады на протяжении всей работы. Но для небольших групп (не более 10 человек) эта структура обычно себя оправдывает, а в отдельных случаях ее производительность может пятикратно

превышать производительность традиционной иерархической бригады. Третий тип бригады программистов характеризуется отсутствием как формальной иерархической структуры, так и ролевого распределения: все равны и работают на паритетных началах. В такой бригаде без персонализации каждый этап решения задачи осуществляется совместно. Опыт коллективной работы приобретает здесь очень большое значение. Наиболее успешно в ней осуществляется разовое задание (отдельный проект, программа) при сравнительно одинаковой компетентности участников. Сохранить же на длительный срок паритетность отношений затруднительно (отсутствие са. .ций, перспектив формального продвижения по службе, трудности с равномерным распределением работы, оплатой и пр.). Поэтому традиционные иерархическая и функционально-ролевая структуры являются более стабильными.

Е. Д. Маргулис.

БУТОВЫЕ ВИРУСЫ — тип *компьютерных вирусов* (KB), характерным признаком которых является размещение их головы в бут-секторе (секторе загрузки) *дискеты* либо в одном из двух бут-секторов *винчестерского накопителя*. Б.в. — весьма специализированная разновидность резидентных *файловых вирусов*. Упрощенно их можно представить себе как специализированный резидентный KB, загружающий единственный “файл” — загрузочный сектор гибкого или жесткого диска. Четкой границы между резидентными файловыми вирусами и Б.в. нет: существуют гибриды, сочетающие заражение *файлов* с заражением бут-сектора винчестера.

Б.в. распространяются, инфицируя дискеты, причем как заражаемые (т.е. содержащие копию *операционной системы MS DOS*), та и незагружаемые (содержащие любую *информацию*). Заражение незагружаемых дискет связано с определенным психологическим расчетом: при перезагрузке *MS DOS пользователи* обычно забывают проверить, вставлена ли дискета в дисковод А, и часто перезагрузка выполняется со вставленной в указанный дисковод дискетой, с которой вирус и попадает на винчестер. Инфицированная дискета всегда содержит часть *кода* вируса, поэтому Б.в. легко обнаружить. При просмотре бут-сектора дискеты на незараженном компьютере хорошо видны изменения загрузчика. Структура стандартного загрузчика лишь незначительно меняется от версии к версии *MS DOS*, поэтому изменения легко выявить визуально (особенно первых трех *байтов*, содержащих команду перехода на начало загрузчика) либо с помощью подходящей *программы*.

Файловые вирусы обычно состоят из одного сегмента. Первую команду перехода, подставляемую большинством файловых вирусов, дописывающих свое тело в конец *COM-файла*, можно рассматривать как выродившийся сегмент. Б.в. всегда состоит из нескольких

сегментов. Обычно таких сегментов два (голова и хвост). Положение головы на диске определено однозначно: она всегда расположена в бут-секторе и занимает ровно один сектор. На винчестере ситуация несколько сложнее: голова может располагаться в одном из его двух бут-секторов: главном (расположенном по абсолютному дисковому адресу 0/0/1) или бут-секторе логического диска С, обычно занимающего первые сектора соответствующего логического диска. Для наиболее распространенных 40М винчестеров это обычно абсолютный сектор 18 (при 17 секторах на трек его адрес равен 1/0 /1). Хвост может располагаться в разных местах: кластерах, отмеченных на диске как сбойные „ тем самым исключенных из дальнейшего распределения (псевдосбойные кластеры); последних физических секторах дискеты или винчестера, поскольку они обычно свободны (напр., 40/39/8/ для обычных 360К дискет); в неиспользуемых блоках таблицы размещения файлов, главного каталога или одного из подкаталогов; дополнительных дорожках дискеты или винчестера '41 и последующие дорожки для 360К дискет). В любом случае хвост должен содержать копию оригинального бут-сектора, и если она хранится не в закодированном виде, положение хвоста в большинстве случаев можно определить глобальным контекстным поиском по дискете или винчестеру (исключение — использование для хвоста дополнительных дорожек на дискете либо винчестере). В миним. варианте хвост может состоять только из одного сектора с оригинальным бут-сектором.

Для всех Б.в. механизм заражения однотипен. Когда MS DOS загружается с зараженного диска, Б.в. получает управление и сначала копирует себя в старшие адреса *памяти ЭВМ*. Затем он уменьшает размер памяти, заменяя значение вектора прерываний A2, чтобы защитить резидентную часть вируса, и 13, чтобы перехватывать обращения к диску. Следовательно, при любом обращении к диску управление получает обработчик этого прерывания, составляющий осн. часть тела вируса. После этих действий вирус запускает стандартный системный загрузчик, т.е. происходит стандартная загрузка системы. Получив управление по прерыванию по чтению, вирус анализирует, относится оно к дискете или винчестеру. Если это прерывание относится к дискете, то вирус сначала проверяет, заражена уже она или нет. Для этого считывается бут-сектор и проверяется его содержимое. Если дискета еще не заражена, вирус заражает дискету, а затем обрабатывает команду чтения. Если дискета уже заражена, вирус сразу переходит к обработке команды чтения диска; так же он поступает, если дискета защищена от записи. Свой хвост Б.в. чаще всего размещает в свободном кластере (кластерах), который помечается как сбойный. Последнее необходимо для того, чтобы занятый вирусом кластер (кластеры) не был использован MS DOS при создании нового файла. Вместе с тем это неплохой метод маскировки, поскольку пользователи обычно плохо

представляют себе структуру диска и не следят за количеством сбойных кластеров на нем.

После зависания MS DOS часто выполняется попытка ее перезагрузки. Если при этом инфицированная дискета окажется в "готовом" дисководе А, то, естественно, загрузка будет сначала выполняться с нее. В этом случае программа начальной загрузки считывает зараженный бут-сектор и передает ему управление. Вирус загружает себя в оперативную память и имитирует сообщение, выдаваемое в этом в случае стандартным загрузчиком:

Non system disk.

При этом, если пользователь открывает дисковод с дискетой и нажмет клавишу Enter (ввод), то вирус останется в памяти, заразит винчестер и в дальнейшем будет заражать все вставляемые дискеты, к которым производится хотя бы одно обращение (достаточно команды "выдать оглавление"). Все Б.в. перехватывают 13 прерывание и поэтому обычно конфликтуют с драйверами, поддерживающими нестандартные форматы (напр., 720К). В лучшем случае при этом происходит зависание системы или выдача сообщения о делении на нуль. В худшем случае MS DOS остается работоспособной, на дискете что-то записывается, однако прочитать с нее ничего нельзя. Особое внимание следует обратить на то, что некоторые Б.в. перехватывают прерывание с клавиатуры и могут пережить в оперативной памяти мягкую перезагрузку.

Для борьбы с Б.в. нет необходимости всегда просматривать содержимое бут-сектора. Поскольку подавляющее большинство дискет не являются загружаемыми, можно профилактически разрушать содержимое бут-сектора таких дискет. В частности, там можно хранить информацию о времени и источнике получения дискеты либо какие-то указания по ее использованию. Достаточно разрушить первые три байта бут-сектора, чтобы нейтрализовать любой Б.в. Это можно сделать с помощью программ утилит Нортон или любой другой подходящей утилиты. Для вакцинации бут-сектора дискеты используют специальную вакцину (см. *Антивирусы*). При соблюдении этих рекомендаций заражение "чистым" Б.в. можно полностью исключить.

Н.Н.Безруков.

WHY — *экспертная обучающая система*, предназначенная для изучения сложных геофизических процессов. Разработана в конце 70-х гг. в США, является расширением системы SCHOLAR.

База знаний (БЗ) о предметной области (ПО) в WHY представлена семантической сетью, содержащей фактические

сведения о климате и географии отдельных районов, а также описание причинно-следственных и временных связей между метеорологическими явлениями. БЗ о процессе обучения представлена 24 эвристическими продукционными правилами (в форме продукций, т.е. пар вида: условие применения правила — действие), осуществляющими взаимодействие *обучаемого* с системой. Левая часть правила описывает ситуацию, в которой оно применимо, в терминах *задачи обучения*, правая содержит описание *обучающего воздействия*, которое должно осуществляться в данной ситуации. Помимо локального управления *диалогом*, описываемого эвристическими правилами, в системе существует планирование обучающих воздействий, выраженное метаправилами (см. *Метазнания*), определяющими порядок анализа ситуаций и применения допустимых правил. В целом БЗ о процессе обучения связана со структурой БЗ о ПО, а не с ее содержанием, и может быть использована для обучения др. ПО сходной структуры.

Обучающий диалог представляет собой обсуждение утверждений о климате различных районов и выяснение рассуждений обучаемого, используемых для доказательства или опровержения утверждений. *Модель обучаемого* в ВНУ является накрывающей. Следовательно, задача ВНУ заключается в обнаружении различий между БЗ о ПО и представлениями обучаемого и, при необходимости, корректировке этих представлений. Возможные ошибки обучаемого разделены на 6 классов, для каждого из которых предложен свой метод корректировки знаний. Осн. внимание в ВНУ уделяется ошибкам в понимании причинно-следственных связей явлений; в результате формируются правильная *модель* рассуждений и умение использовать известные факты для подтверждения и опровержения гипотез. Ошибки недостаточного учета факторов и ошибки включения излишних факторов при объяснении причин явления корректируются путем предъявления соответствующих контрпримеров и обсуждения их причинно-следственной структуры. Кроме того, в ВНУ предпринята попытка частичной корректировки ошибок в построении рассуждений, а также выявлен класс так наз. функциональных ошибок, обработка которых требует дополнения БЗ о ПО функциональными моделями явлений. Ошибки в фактическом материале и понимании причинно-следственных связей вне пределов изучаемой ПО исправляются путем непосредственного указания верных сведений, без привлечения дополнительного материала, поскольку передача такого рода знаний не является первоочередной целью обучения.

Е.М. Синица.

WISE (англ. Wicat Interactive System for Education — диалоговая система для образования фирмы Wicat) — универсальная адаптивная открытая *авторская система*, предназначенная для поддержки этапов *проектирования* и реализации *программ учебного назначения* (ПУН). Позволяет создавать линейные и разветвленные ПУН демонстрационного, информационно-справочного, вопросно-ответного, обучающе-контролирующего и др. типов. Система разработана американской фирмой Wicat и выполняется на 32- или 16-разрядных *процессорах*, поддерживаемых многопользовательской и многозадачной *операционной системой* на компьютерах фирмы DEC серии PDF-11/VAX- 11, а также IBM PC. WISE поддерживает работу с цветными и монохромными *терминалами*, использует цифровую звукозапись, видеодиски и видеоленты. *Программа*, генерируемая системой WISE наз. уроком, который состоит из фрагментов.

Структура системы включает: экранный редактор; редактор шрифтов; библиотеку графических объектов; редактор функциональных клавиш; средство разработки логики урока, основанное на *меню*; *авторский язык программирования*; словари синонимов и слов, игнорируемых при обработке *ответов*; средство быстрого прототипирования уроков; отладчик логики уроков; интерпретатор уроков и др. В системе WISE *автор* может работать в трех режимах различной степени сложности: 1) *меню* — верхний уровень, на котором не требуется *программирования*; этот уровень достаточен для создания многих уроков; 2) *авторский язык* — уровень, требующий *программирования* на *проблемно-ориентированном языке*; может быть использован для создания как всего курса, так и любой его части; 3) фрагменты внешних программ — уровень, определяющий мощность системы и позволяющий использовать в уроке все возможности операционной системы.

Экранный редактор позволяет создавать и редактировать *текст*, графические изображения и мультипликацию. Он дает возможность объединять группу графических объектов в сегмент, а затем управлять им как одним объектом; при этом не утрачивается возможность оперировать составными частями сегмента. Все объекты можно масштабировать относительно осей X и Y, создавать зеркальные отображения, копировать, вращать и передвигать по определенному пути. Для ввода готовых рисунков поддерживается графический планшет. При выводе на экран WISE используют цифровую звукозапись, видеодиск и видеоленту. Редактор шрифтов позволяет создавать неограниченное число специальных *символов*, которые можно изменять в размерах, передвигать, вращать. Созданный в одном фрагменте графический объект можно использовать в любом другом фрагменте урока. Графические изображения вызывают также из библиотек, специально созданных авторами. Эти изображения можно модифицировать и мультиплицировать. В редакторе функциональных клавиш автор, используя меню, может любой

клавише поставить в соответствие специальную функцию, напр. дать возможность *обучаемому* в любой момент времени делать свои комментарии, которые можно послать в *файл* для дальнейшего изучения или сразу выдать на *печатающее устройство*. Большую часть работы автора можно выполнить с помощью многоуровневого меню. Опытный *пользователь* может делать выбор из любого меню, пропуская промежуточные уровни. В системе WISE урок или его фрагменты создают также с помощью авторского языка, содержащего следующие средства: вычисление арифм. выражений, формирование выводов на экран, условное ветвление, использование системных переменных, принятие и обработка ответов обучаемого, вызов логики и выводов на экран как *подпрограмм* из других фрагментов (допускается вложенность любого уровня), управление *периферийными устройствами* (такими, как плееры и видеодиски), работа с буферами общих переменных, доступными всем пользователям одновременно, и др. WISE поддерживает такие способы ответа обучаемого: 1) печать алфавитно-цифрового ответа, который может обрабатываться как строка, арифм. выражение или число; 2) выбор поля экрана с правильным ответом с использованием функциональных клавиш; 3) касание той части экрана, где находится правильный ответ. Время ответа можно ограничить. Обработка ответа включает возможности: 1) поиск *ключевых слов*, при этом игнорируются остальные слова или слова, перечисленные в *словаре* игнорируемых слов; 2) применение словарей синонимов; 3) учет или игнорирование числа пробелов в ответе; 4) поиск подстроки в длинной строке, удаление ее; 5) запись и выполнение *операций* с позицией подстроки. Логiku обработки ответа любой сложности можно использовать в любом фрагменте курса как подпрограмму. Доступны такие системные переменные, как ответ обучаемого, время ответа, идентификатор фрагмента и др. В системе WISE есть возможность выхода ответа, идентификатор фрагмента и др. В системе WISE есть возможность выхода из урока в среду операционной системы посредством так наз. фрагментов внешних программ. В этих фрагментах можно задать имя любой программы, выполняемой операционной системой, которая поддерживает полные реализации языков *Кобол*, *Фортран*, *Паскаль*, *Си*, *Бесик* и др. После выполнения внешней программы управление передается следующему фрагменту урока. WISE позволяет быстро дублировать логику ветвления, определение функциональных клавиш, словари синонимов обработки ответов и т.д. из аналогичного урока. Новая копия урока содержит (либо не содержит) графику и выводы на экран из прототипного урока. Используя функции глобального поиска и замены *текстового редактора* системы, с помощью нескольких *команд* можно вносить множество изменений в исходный урок. Эти операции выполняют макродрайверы, написанные на АПЛ и Паскале. Для экономии *памяти ЭВМ* на диске вместо копий экранов и логики можно использовать подпрограммы.

Режим отладки позволяет автору выполнить урок в режиме обучаемого. При этом система выдает автору идентификаторы выполняемых фрагментов, генерирует файл трассировки, включающий список всех номеров и типов фрагментов в порядке их выполнения. В режиме отладки можно также выполнить урок с включением в него так наз. фрагмента трассировки. Обычно это фрагмент вывода на экран или вычислительного типа. При выполнении урока в режиме отладки фрагмент трассировки автоматически выполняется после каждого фрагмента урока и выводит на экран состояние переменных, включенных в этот фрагмент. Режим отладки включает и автоматический режим обучаемого, выполняющий все определенные ветви для выполнения несуществующих или неполных ветвей.

Т.А.Лисюк.

ВАРИЕЛ — локальная сеть учебного назначения, позволяющая соединять восьмиразрядные микро-ЭВМ с помощью двужильного кабеля. Передача сигналов последовательная со скоростью 32 К.

В. содействует процессу преподавания путем пересылки из компьютера преподавателя программы обучения, тест-программы или упражнения в компьютеры обучаемых. Осуществляет контроль за ходом работы, записывает результаты на дискету для дальнейшей обработки. С помощью В. проводится также обучение языкам программирования (ЯП). При изучении языка Бейсик, имеющегося в постоянной памяти большинства учебных компьютеров, на дискету компьютера преподавателя можно записывать созданные обучаемыми программы, использовать печатающее устройство для печатания текстов, а также пользоваться другими устройствами, подключенными к сети. При изучении других ЯП на компьютере преподавателя программа-редактор позволяет написать исходную программу на требуемом языке и записать ее на дискету. На дальнейшем этапе эти программы транслируются компьютером преподавателя и могут быть запущены в компьютерах обучаемых. В. применяется также для поддержки автоматизированной системы управления малых школ при помощи DBS-2 с несложным обслуживанием файла и использованием файла для текущей работы при помощи программы-редактора.

Обновленная версия В.-2 дополнена возможностью работы с дистрибутированной системой CP-NET. Можно выбрать работу либо в вышеописанном режиме сети В., либо в рамках этой системы.

П.Горват, В.Шеховцов.

ВЕРИФИКАЦИЯ ПРОГРАММЫ — доказательство правильности программы по отношению к спецификации решаемой задачи. Методы В.п. различаются по способам формализации семантики программ и

способам специфицирования решаемой задачи. Наиболее известен метод Флойда-Хоара, заключающийся в следующем. Программа I рассматривается как соответствие между множеством состояний D , определяемых исходными данными, и множеством состояний R , соответствующих результатам работы программы (для детерминированных программ это соответствие является функцией). Спецификация задачи задается парой формул — предусловием P и постусловием Q . Предусловие задает ограничения на входные данные программы, а постусловие определяет желаемый результат программы. Напр., задача вычисления факториала натурального числа n в диапазоне от 1 до 10 имеет спецификацию: предусловие $P = (1 \leq n \leq 10 \wedge n \in \mathbb{N})$, постусловие, где m — выходная переменная

(результат). Запись вида $\{P\}I\{Q\}$ выражает утверждение о том, что если предусловие P истинно в исходном состоянии $d \in D$, то завершение программы I приведет к состоянию $r \in R$, в котором истинно постусловие Q . Доказать частичную правильность

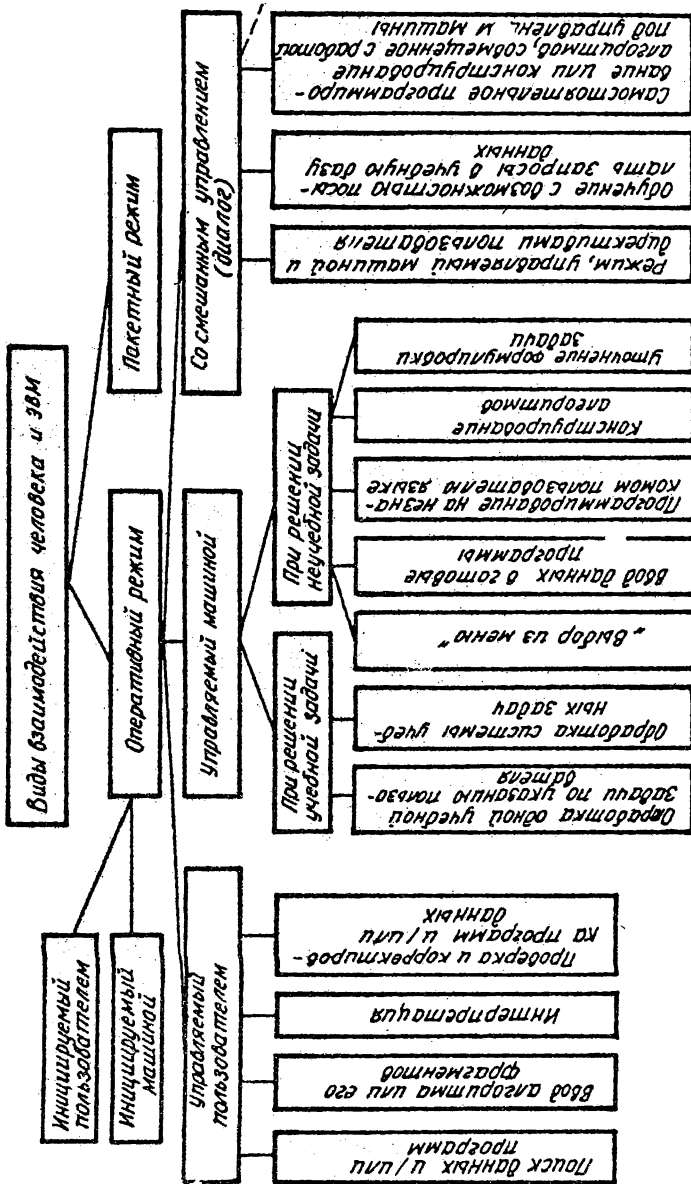
программы I по отношению к спецификации P, Q означает доказать истинность утверждения $\{P\}I\{Q\}$. Для доказательства полной правильности программы I относительно P, Q необходимо помимо

утверждения $\{P\}I\{Q\}$ доказать, что для всех входных данных,

удовлетворяющих предусловию P , программа I завершится результативно (т.е. не заикнется и не выдаст сообщение об ошибке).

В.М. Антимиров.

ВЗАИМОДЕЙСТВИЕ ЧЕЛОВЕКО-МАШИННОЕ — процесс обмена сообщения между человеком и *электронной вычислительной машиной*, обусловленный необходимостью последовательного и/или параллельного выполнения человеком и машиной действий и операций по решению какой-либо задачи. Различают (рис.) два осн. вида В.ч.-м. — пакетный и оперативный (диалоговый) режимы. При пакетной обработке информации значительную часть процесса решения задачи (чаще всего разработку метода или алгоритма решения) человек выполняет, не обмениваясь регистрируемыми сообщениями с ЭВМ. В этом случае условно можно говорить о “неявной” форме взаимодействия: человек производит мысленный обмен сообщениями с моделью ЭВМ, отраженной в его сознании. Пользователь и программист, как правило, не имеют прямой связи с ЭВМ. При оперативном режиме пользователь непосредственно связан с ЭВМ через индивидуальный терминал или абонентский пункт и получает сообщения машины достаточно быстро, обычно через интервал времени, не нарушающий естественный ход его мысли. При оперативном режиме сокращается общее время решения



задач различных классов, а также создаются более благоприятные условия для пользователей низкой квалификации, не овладевших в достаточной мере *языком программирования* (появляется возможность проверки "на месте" правильности текста формируемой программы и/или адекватности исходных данных). Оперативный режим реализуется благодаря развитым системам автоматизации *программирования и операционным системам*, упрощению связи с машиной и др. Подобный режим реализован как на *персональных электронных вычислительных машинах* с относительно невысоким быстродействием, так и на мультипроцессорных быстродействующих машинах, обладающих *режимом разделен'я времени*.

На разных этапах решения задачи В.ч.-м. несет различную функциональную нагрузку. На начальном этапе решения взаимодействие обеспечивает регистрацию пользователя и его задачи, наведение им справок о возможностях данного компьютера и (в ряде случаев) заказ ресурсов, требуемых для решения данной задачи. В дальнейшем при известном человеку алгоритме решения задачи в результате взаимодействия производится *отладка программы*. На завершающем этапе В.ч.-м. должно обеспечить построение неизвестного ранее алгоритма или метода решения задачи. При этом могут быть использованы как творческие возможности человека, так и соответствующие "интеллектуальные" программы, заложенные в машину.

Разновидностью оперативного В.ч.-м. является *речевой диалог* (интерактивный, разговорный режим), при котором пользователь имеет возможность вмешиваться в ход решения задачи на машине и получать необходимые ему справки и помощь, обучение и тренировку. В этом режиме процессом решения задачи может управлять не только пользователь, как в подавляющем большинстве *систем программирования* и автоматизированных *информационных систем*, но и ЭВМ. Диалоговые режимы, управляемые машиной, применяют при *компьютеризированном обучении* (см. *Автоматизированная обучающая система*), подготовке и вводе данных в готовые программы, программировании и конструировании алгоритмов на незнакомом пользователю ЯП и др. Особый интерес вызывает диалог человека и ЭВМ на *естественном языке ограниченном*. Реализация диалогового взаимодействия человека с ЭВМ обеспечивается на основе высокоразвитых технических средств базового и проблемно-ориентированного *программного обеспечения* компьютера, включающего программы *автоматизированного обучения* и консультирования пользователей, автоматического конструирования программ и доказательства их правильности и др.

Развитие диалоговых режимов идет в направлении снижения требований к пользователю. В перспективе от него потребуются только сформулировать свою задачу и иметь хотя бы общие сведения о возможностях данного компьютера. С этой целью реализуется такое

расширение операционной системы ЭВМ, которое “знает” обо всех возможностях машины, а также “умеет” (после опроса пользователя) отнести его задачу к определенному классу для оказания ему необходимой помощи в программировании, конструировании алгоритма решения задачи и т.п. При диалоговом взаимодействии возрастают требования не только к уровню реализации фактора взаимопонимания, но и к фактору быстроты реакции машины — *время ответа компьютера должно быть минимальным*. Важным средством реализации диалогового режима В.ч.-м. является также включение в состав штатного проблемно-ориентированного программного обеспечения АОС, других прикладных программ *учебного назначения* для обеспечения подготовки пользователей данного компьютера к решению задач с помощью этого же компьютера.

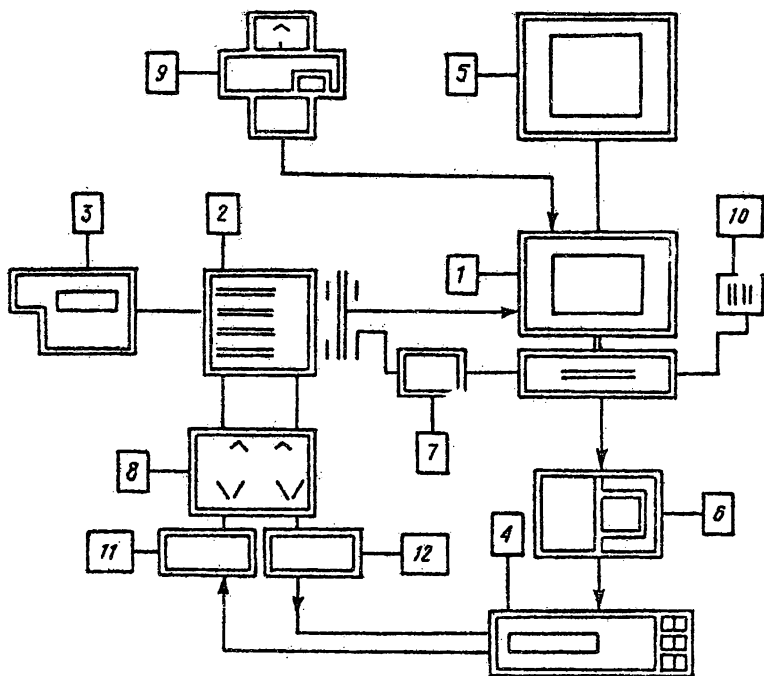
В связи с развитием персональных ЭВМ значительно возросло количество технических средств взаимодействия человека с компьютером. По типу технических средств способы взаимодействия человека с компьютером разделяются на оптические и речевые (см. *Речевая коммуникация*). Самое распространенное интерактивное оборудование *оптической коммуникации* — индикаторное и отображающее оборудование, отображающее с *клавиатурой* и *световым пером*, графические регистрирующие приборы и т.п. Для автоматизации сбора данных применяются счетчики первичных документов. По типу записи и свойствам считывающего устройства счетчики документов делятся на счетчики *знаков*, дающие возможность считывать простые знаки, записанные на точно определенных местах предварительно обозначенного формуляра, счетчики знаков для считывания алфавитно-цифровых знаков и без дополнительного кода (в частности, записываемых на первичные документы с помощью специальных пишущих машинок со стилизованными знаками письма), счетчики рукописного текста для считывания и распознавания написанных от руки алфавитных и цифровых знаков на предварительно обозначенных местах первичного документа; счетчики графических изображений для считывания графических знаков на первичных документах, напр. производственных чертежей. См. также *Общение человека-машинное*.

П.Горват, А.М.Довгялло.

ВИДЕОКОМПЬЮТЕРНАЯ СИСТЕМА (ВКС) — совокупность видео и компьютерной техники, средства связи, прикладного и инструментального программного обеспечения и информационного обеспечения. Объединяет возможности *персональной электронной вычислительной машины* и видеотехники. Необходимость применения ВКС в учебном процессе обусловлена непрерывным возрастанием объема научно-технической информации и числа обучаемых, что требует поиска и внедрения новых методов и средств повышения

производительности педагогического труда, повышения качества обучения, его индивидуализации, интенсификации, а в итоге — повышения эффективности учебного процесса в целом. Внедрение и применение ВКС, как и разработка и создание специализированных дидактических материалов, проводится в учебно-методических *видеокomпьютерных комплексах* (курсах). В комплексном использовании видео- и компьютерной техники заложена идея: в одном и том же зрительном поле последовательно или одновременно показывать развитие процессов или явлений с натуры, которые воспроизводятся видео- или другой визуальной техникой, и абстрактные *модели* — *текст*, формулы, графики, анимации (электронной мультипликации) и т.д., — созданные компьютером. При этом установлено, что в среднем на обучение с помощью *диалоговой системы* на базе видеоустройства расходуется в 3 раза меньше времени, чем при использовании любых других методов. При рассмотрении ВКС как комплекса аппаратно — программных средств, реализующего функциональную направленность и возможность применения в образовании, выделяют две большие группы систем: системы, непосредственно участвующие в учебном процессе, — *видеокomпьютерные рабочие места*; системы, необходимые для создания видео- и видеокomпьютерных программ для систем первого типа.

ВКС, позволяющая обрабатывать и синтезировать изображения и предназначенная для подготовки видеоматериалов с компьютерной граф. иккой и анимацией, наз. авторским видеокomпьютерным рабочим местом (ис.). Такая система позволяет вводить в персональный компьютер (1) и сохранять по кадрам на жестком/гибком диске или в кадровом буфере (7) цветное изображение от источника видео PAL- или RGB-сигнала с разрешением от 256*512 до 2048*2048 элементов изображения и от 8 до 32 битов кодирования цвета на пиксел. Контроллер видеокadров (6) совместно с платой ввода — вывода и обработки изображения (видеографическим адаптером 2) позволяет смешивать введенное изображение с компьютерной графикой и анимацией и выводить на видеомагнитофон по кадрам или в реальном масштабе времени. Как правило, введенная картинка сохраняется в одном из стандартных графических форматов (если сохраняется не в стандартном, то используются конвертировщики из одного формата в другой) и доступна для стандартных *графических редакторов*. Кроме библиотеки ввода — вывода, плату ввода и обработки изображения могут поддерживать библиотека обработки изображений, пакеты по созданию анимационных последовательностей, двух- и трехмерные графические пакеты, картинная *база данных*. Для ввода в компьютер изображения от видеомагнитофона по кадрам необходимо использовать цифровой корректор временных искажений (11), а для вывода на магнитофон — генератор временного кода (12). Некоторые магнитофоны комплектуются этими устройствами стандартно. Если плата ввода — вывода не работает с



композитным видео-сигналом, то для сопряжения ее с видеомагнитофоном необходимо устройство кодирования/декодирования видеосигнала (8). RGB-монитор высокого разрешения (5) используется в качестве контрольного устройства при вводе и обработке изображений.

Особый класс ВКС составляют интерактивные и моделирующие рабочие видеостанции для записи и авторской работы по созданию учебных кино- и видеофильмов. Используя негативные или позитивные киноматериалы, записанные видеокадры или видеофрагменты и звуковые материалы, *пользователь* может создать интерактивный аудиовизуальный курс. В состав установки, в которой используется сигнал NTSC, входят: телекинодатчик/теледиадатчик, кассетный видеомагнитофон стандарта VHS, сценарный видеомонитор, интерактивный видеомонитор, генератор временного кода, синхрогенератор, дисковый аппарат оптической видеозаписи, телевизионная камера, компьютер PC/AT с *клавиатурой, печатающим устройством* и дисководом. Все кино-, диа- и телевизионные изображения, используемые в установке, записываются на аппарате с оптическим видеодиском. При этом записывается также временной код и сигнал

синхронизации, что обеспечивает в дальнейшем возможность быстрого отбора и монтажа. Запись производится на 12 дорожках, что упрощает процедуру монтажа. При разработке данной установки была поставлена цель достигнуть высокой эффективности при хорошем качестве изображения. Применение персональной ЭВМ позволяет при авторской работе над кино- или видеополоской воспроизводить на мониторе, выбирать и включать в фильм компьютерную видеографику, таблицы, тексты.

Г.Ж.Желев, А.П.Ильяшенко, Э.Л.Шестериков.

ВИДЕОКОМПЬЮТЕРНАЯ ТЕХНОЛОГИЯ ОБУЧЕНИЯ — функциональная совокупность психологических, педагогических, кибернетических методов, видео- и компьютерной техники и обеспечивающих систем, направленная на повышение эффективности обучения и его управления.

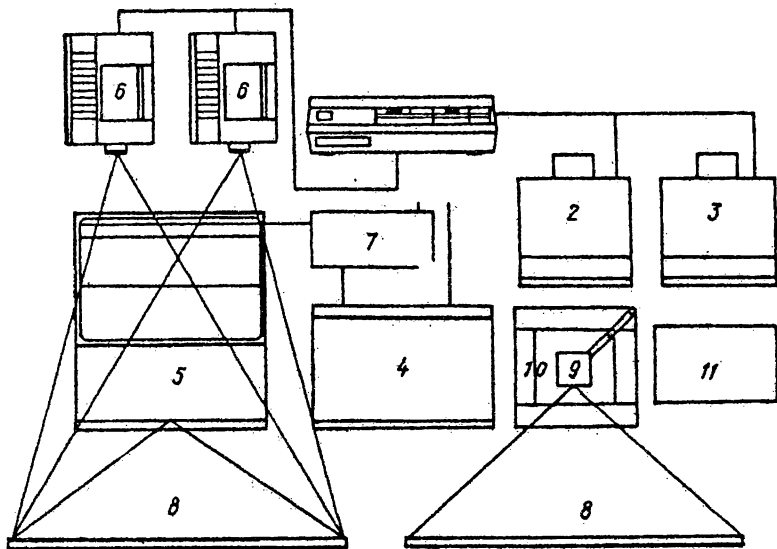
В состав технического обеспечения *видеокомпьютерных систем* входят все известные технические средства, применяемые в учебном процессе (диапроекторы, кодоскопы и др. видеотехника, техника связи и пр.), которые при помощи специальных контроллеров и адаптеров работают под управлением *персональной электронной вычислительной машины*. Она выполняет две функции: генерирующую и управляющую, т.е. является генератором *текста*, графики, моделирует явления и т.д., управляет видеопроекторными средствами, организованными и сфокусированными так, что обеспечивается чередование и смешивание нескольких изображений различных источников в одном и том же зрительном поле. *Информационное обеспечение* В.т.о. состоит из прямых передач или видеозаписей процесса изучаемого явления. Цифровые и аналоговые *файлы*, обеспечивающие поддержку информационного процесса, определены и организованы как *база знаний*.

Чередую, смешивая, проникая одним изображением в другое, можно создавать сложные многопластовые картины, многократное сложное стереоизображение. Создаются условия такого *диалога с обучаемым*, при котором учебная *информация* и связанные с ней вопросы предлагаются как в реальной жизни, а затем осмысливаются, научно обосновываются в абстрактном виде и выдаются на новые качественные уровни. Наличие видеофрагментов и части видеополоски придает системе новые интеллектуальные качества с возможностью более сильного психологического, эстетического и воспитательного воздействия. Все это создает условия для эксперимента и внедрения новых, не применявшихся до сих пор психологических, воспитательных, педагогических и др. методов. См. также *Инструментальные средства видеокомпьютерной технологии обеспечения*.

Г.Ж.Желев.

ВИДЕОКОМПЬЮТЕРНОЕ РАБОЧЕЕ МЕСТО — *автоматизированное рабочее место*, в состав технических средств которого наряду с компьютером входит и аппаратура для визуализации информации. В обучении используют В.р.м. обучаемого (ВРМО) и преподавателя (ВРМП). ВРМО состоит из микро-ЭВМ, цветного видеомонитора или телевизора и видеоустройства, управляемого микро-ЭВМ. Это управление реализуется при помощи специального видеокомпьютерного контроллера для видеомагнитофона, адаптера для видеомонитора и телевизора либо специализированного компьютерного контроллера для видеодисковых устройств. Для минимизации звуковых воздействий на соседнее рабочее место исполняют наушники. Действие ВРМО заключается в чередовании или смешивании видеоизображения микро-ЭВМ и видеоустройства на одном и том же мониторе. Видеоустройство воспроизводит записи развития процессов и явлений в реальных условиях, а микро-ЭВМ синтезирует текст, графику, анимацию и другие абстракции и модели, необходимые для научного обоснования изучаемой области знаний. Действиями обучаемого на ВРМО управляет интеллектуальная интерактивная программа, которая может рассматриваться в двух вариантах: авторском и потребительском. Авторский вариант состоит из графического редактора, текстового редактора, шрифтового, видеокомпьютерного, курсового и др. совместимых редакторов. Потребительский вариант — это система, созданная из вышеперечисленных редакторов, связанная обучающей программой, обеспеченная необходимым драйвером для управления видеоустройством и комутации видеомонитора.

ВРМП (рис.) состоит из видеопроецирующих и видеопроекторных средств, автоматических диапроекторов и кодоскопов, др. устройств, которые обеспечивают адаптивные информационные возможности преподавателя при предъявлении учебного материала. Сложность, абстрактность и углубленность усвоения учебных предметов требуют 16- или 32-битовых микро-ЭВМ. Напр., при предъявлении знаний, связанных с изучением сложных систем, автоматизацией инженерного труда, базой данных, статистическими и моделирующими пакетами и т.д. необходимы большая разрешающая способность дисплея, большая оперативная и внешняя память ЭВМ (гибких и твердых дисков), быстроедействие, работа с графическими периферийными устройствами и др. В зависимости от характерных особенностей учебного предмета некоторые проблемы, связанные с процессами и явлениями из реального мира, можно показать диапозитивами диапроектора или кодоскопа. Тексты, графики, модели и другие абстракции, необходимой теор. части учебной дисциплины генерируются из микро-ЭВМ и демонстрируются через текстовый дисплей и приставку на жидком кристалле. Это изображение смешивается с изображением пленки кодоскопа на экранах. Изображение фотопленки статичное, цветное с самой большой разрешающей



способностью из всех элементов системы, а изображение текстового дисплея статичное или динамичное, с ограниченными цветовыми возможностями. Если необходимо динамичное изображение, используется видеомаягнитофон или видеодисковое устройство под управлением микро-ЭВМ, которая находит один или другой видеофрагмент по предварительно составленной программе. К видеопроекционной системе подключен графический дисплей микро-ЭВМ, имеющий достаточно большие разрешающие возможности и палитры цветов. Существует возможность смешивания изображения текстовой и графической страницы, что невозможно в стандартной системной конфигурации микро-ЭВМ. Если методика и специфические особенности учебного материала обуславливают применение двух экранов, для отображения цветных статических кадров с большой разрешающей способностью можно использовать изображения автоматического диапроектора. Автоматический диапроектор, видеомаягнитофон и видеопроекционная система работают под управлением компьютера при помощи специальных драйверов и контроллеров. Такое управление позволяет составить гибкую управляющую программу, дающую преподавателю возможность выбора ритма и демонстрационного материала в зависимости от компетентности и интеллектуального уровня аудитории.

Конфигурация ВРМП требует видеопроекционной системы с разрешающими возможностями, которые находятся в соответствии с целями обучения. Если демонстрируется компьютерная графика, необходимы видеопроекционные системы с индексом HR (высокая разрешающая способность). Конфигурация системы будет заканчиваться включением одного или двух автоматических диапроекторов либо пишущего проектора, обеспечивающих необходимую разрешающую способность в статическом виде при аналоговом представлении изучаемых объектов. Кодоскоп может быть укомплектован приставкой на жидких кристаллах для демонстрации изображения другим текстовым дисплеем микро-ЭВМ. Целостное включение системы в сеть и ее управление вручную реализуются посредством пульта.

ВРМП предлагает следующие экранные возможности (как в отдельности, так и комбинированно). Первый экран: демонстрация фолиограммы с кодоскопа; демонстрация псевдографики и текста с помощью приставки на жидких кристаллах; комбинированные демонстрации фолиограмм с большой разрешающей способностью и элементарной мультипликацией, псевдографики и текста с помощью приставки на жидких кристаллах. Второй экран: демонстрация диапозитивов, обеспечивающих статистическое изображение с высокой разрешающей способностью с одного или двух автоматических диапроекторов; демонстрация компьютерной графики, мультипликации и текста в режиме HR; демонстрация видео- и видеокомпьютерных фильмов и фрагментов; комбинированная демонстрация диапозитивов и компьютерной графики, мультипликации и текста и других возможных комбинаций. Оба экрана можно использовать как поочередно, так и вместе. В.р. м. не исключает использование учебной доски, наоборот, она является обязательным элементом необходимым для немедленной актуализации преподаваемого учебного материала, более полной *адаптации* его к аудитории, для выяснения любых возникающих вопросов, не предусмотренных в сценарии лекций. Разработка методических указаний для создания видеокомпьютерных лекций тесно связана с *авторской системой* и видеокомпьютерным модулем и является объектом специального исследования.

Г.Ж.Желев.

ВИДЕОКОМПЬЮТЕРНЫЙ КОМПЛЕКС — совокупность технических и программных средств для визуализации определенного вида деятельности. В состав учебно-методического В.к. входят: учебные и лекционные залы, каждый из которых оборудован *видеокомпьютерным рабочим местом* преподавателя; видеочитальни, оборудованные видеокассетофонами, мониторами и наушниками для индивидуального прослушивания и просмотра видеозаписи лекций и новых уроков, учебных уроков, учебных и научных фильмов; видеокомпьютерные

классы для активной самоподготовки и *автоматизированного обучения*, оборудованные видеокомпьютерными рабочими местами *обучаемых*; методические кабинеты для подготовки и просмотра видеокомпьютерных программ, оснащенные перечисленными выше техническими и программными средствами; справочно-информационный и регистрационный блок для обеспечения *преподавателей* и обучаемых необходимыми фильмами или программами; информационные и развлекательно-игровые блоки и системы в фойе и клубах школ и институтов; комплексы, которые должны специализироваться при разработке и создании видеофильмов и фрагментов с помощью компьютерной графики и анимации. Для этого комплексы должны включать необходимые студии и соответствующие аппараты.

Учебно-методический В.к. предназначен для повышения эффективности учебно-воспитательной и научно-исследовательской работы обучаемых и преподавателей. Повышение эффективности достигается улучшением наглядности за счет применения различных дидактических подходов к рассматриваемой учебной теме (объекты и явления засняты в натуре до или во время занятий, процессы в лабораториях или на производстве либо их модели в словесном, графическом или матем. виде синтезированы компьютером); совершенствованием педагогического мастерства преподавателей, повышением учебно-методического уровня уроков, лекций и упражнений путём предварительных записей лекций, анализа их методического, содержательного и информационного характера; улучшением самостоятельной работы обучаемых и ее организации на курсах видеокомпьютерных классов, где при помощи адаптивных обучающих видеокомпьютерных курсов ученики и студенты будут углублять и расширять свои знания; расширением информационных возможностей для воспитательной работы и повышения эмоционального воздействия.

Г.Ж.Желев.

ВИНЧЕСТЕРСКИЙ НАКОПИТЕЛЬ, винчестер — *накопитель* на жестких магнитных дисках, в котором магнитные диски и подвижные магнитные головки помещены в герметичный корпус. Первоначально проектировался на емкость 30 М, обозначалась она 30 /30 (отсюда и название "Винчестер", как у ружья с таким же обозначением). Благодаря герметичному корпусу внутрь не проникают пыль и другие частицы из окружающей среды.

В.н. высоконадежен в эксплуатации. У него значительная продольная плотность записи *информации* вследствие небольшого зазора между поверхностью диска и магнитной головкой и высокой скорости вращения диска.

В.Л.Леонтьев, В.Н.Семихов.

ВНЕШНИЕ УСТРОЙСТВА — то же, что и *периферийные устройства*.

ВОСПРИЯТИЕ РЕЧИ — активный процесс декодирования речевой волны, осуществляемый слуховой системой человека. Выделяют три этапа В.р.: преобразование акустического речевого сигнала в последовательность фонем — *символов*, соответствующих миним. единицам, различным по смыслу; переход от фонем к общему смыслу фразы; интерпретация сведений, получаемых на втором этапе, и *принятие решения* о типе реакции на эти сведения. Исследование свойств слуховой обработки сигнала, в частности выделение “полезных признаков” акустической картины, необходимо для моделирования *процедур* “человеческого” В.р. в технических распознающих системах. Кроме того, интерес к восприятию звуковых единиц обусловлен практическими потребностями *обучения* фонетике иностранных языков.

Слуховая система человека состоит из уха, слухового нерва и соответствующего отдела центральной нервной системы, отвечающего за В.р. Осн. элементом восприятия звуковых колебаний является улитка внутреннего уха, в которой находятся окончания слухового нерва. Улитку можно рассматривать как спектральный анализатор, благодаря которому человеческое ухо различает частоты речевых сигналов. В зависимости от частоты сигналы, воспринятые рецепторными нервными клетками внутреннего уха, передаются затем по определенным слуховым нервам в виде параллельно-последовательных комбинаций в кору головного мозга, где и происходит их распознавание и анализ. Упрощенно можно считать, что восприятие высоты тона зависит от того, какие именно нервные волокна параллельно передают импульсы в головной мозг, а громкости — от числа импульсов, посылаемых по каждому из частотно-зависимых нервных волокон. Следовательно, мы слышим, воспринимая амплитуду сигналов нескольких частот одновременно. Слуховая система позволяет определять амплитуды частот в звуковом диапазоне (приблизительно от 15 до 20 000 Гц).

В результате первичной сегментации речевого сигнала накапливается информация о количестве слогов. Для решения о качественном звуковом составе слогов и о звуковом составе речевого отрезка в целом используется не вся спектральная картина акустического сигнала, а лишь его отдельные признаки, напр., макс. значения определенных параметров, скорость изменения значений параметров и т.д. При этом звуки распознаются не последовательно, один за другим, а человек принимает решение о предыдущем звуке только после анализа последующего звука, т.е. он должен проанализировать весь слог. Из анализа таких физических параметров речевого сигнала, как частота осн. тона, длительность и интенсивность звуков, речевой темп, получают сведения, необходимые для установления количества

слов в высказывании, их выделения и опознавания. При этом решения о словах человек принимает, не дожидаясь окончания фразы; это связано с ограниченным объемом оперативной памяти человека, не позволяющим вместить образ всей фразы в целом. *Информация* об интонации и ритме речи существенна для *синтаксического анализа* высказывания. Из этой информации можно получить также сведения об особенностях конкретного процесса речевого общения: является ли высказывание *запросом* или сообщением информации, какой именно информации, как относится говорящий к сообщаемой им или ему информации и т.д. Большая разборчивость при восприятии слов и фраз по сравнению с разборчивостью бессмысленных слогов в значительной мере зависит от того, что, распознавая осмысленное сообщение, человек опирается на знания высших уровней языка, в частности семантики, а также на дополнительные источники информации о речевой ситуации.

Т.В.Людовик.

ВОЗДЕЙСТВИЕ предмета *B* на предмет *A* — событие, состоящее в том, что предмет *B* (возможно, совместно с к.-л. другими предметами) вызывает или предотвращает некоторое изменение предмета *A*. Может быть непосредственным и опосредованным. При опосредованном *B* существует (и учитывается исследователем) по меньшей мере один такой предмет *C*, что *B* воздействует на *C*, а *C* воздействует на *A*. Частным видом *B* являются целенаправленные действия. Их можно трактовать как *B*. *решателей* на соответствующие предметы задач, осуществляемые для их решения. *B* может осуществляться различными способами, т.е. посредством разных операций или *процедур*. Напр., *B*, приводящее к превращению числа 2 в число 8, может быть осуществлено посредством таких операций, как: прибавление к числу 2 числа 6; умножение числа 2 на число 4; возведение числа 2 в куб. См. также *Оператор*.

Г.А.Балл.

ВОПРОС — тип реплики в *диалоге*, предполагающий в качестве реакции тот или иной *ответ*; форма мысли, ориентированная на поиск недостающего *знания*. С точки зрения теории *задач* — знаковая *модель* (обычно — словесная формулировка) требования *познавательной задачи* либо *коммуникативной задачи* (или же части такого требования, относящейся хотя бы к одному из фигурирующих в задаче искомых предметов). В естественном языке *B* наиболее адекватно выражается вопросительным предложением, хотя может выражаться также повелительными (напр., “Найдите периметр треугольника”) и повествовательными (напр., “Требуется найти периметр треугольника”) предложениями. В свою очередь вопросительное предложение иногда выражает утверждение (так наз.

риторические В.), просьбу, псевдовопрос (фактически не передающий требования к.-л. задачи — (напр., “Какова температура атома газа?”).

При диалоге с компьютером появляются дополнительные аудио-визуальные возможности для предъявления В., повышающие адекватность их восприятия человеком (напр., прерывистый звук, мигание текста В., его локализация в определенном месте экрана и др.).

В. наз. закрытым, если множество ответов на него перечисляется явно или задается его грамматической формой. Прочие В. наз. открытыми. Напр., В. “Чем различаются снег и лед?” является открытым, а В. “Что быстрее тает: снег или лед?” — закрытым. К закрытым относятся, в частности, В., требующие ответа “да” или “нет”.

Г.А.Балл.

ВОПРОСНО-ОТВЕТНЫЕ СИСТЕМЫ — системы общения, в которых основное внимание уделяется анализу и синтезу языковых выражений с помощью соответствующего лингвистического процессора. Используют модель языка (словарь и грамматику) и практически не применяют знания о предметной области, о целях общения, способах их достижения и т.п. В.-о.с. в основном используют жесткую структуру диалога — роли участников общения неизменны, однозначны и predeterminedены заранее. В простейшем случае такая структура сводится к двум взаимосвязанным высказываниям участников (*вопрос — ответ*) с указанием участника, владеющего инициативой.

Осн. внимание при разработке В.-о.с. уделялось не столько возможностям их практического применения, сколько развитию моделей и методов, позволяющих осуществлять перевод естественных языковых высказываний, относящихся к фиксированным предметным областям, в формальное представление, а также обратный перевод. Накопленный опыт разработки В.-о.с. дал возможность, с одной стороны, углубить понимание процесса естественного языкового общения, а с другой — оценить перспективы практического применения естественных языковых систем. В процессе своего развития В.-о.с. переросли в системы естественного языкового общения с базами данных (см. Информационно-поисковая система, ПОЭТ, ДИЛОС, ДИСПУТ, ЗАПСИБ, ЗАПСИБ-ПРОЛОГ).

Н.А.Власенко, С.П.Кудрявцева.

ВРЕМЯ ОТВЕТА КОМПЬЮТЕРА — промежуток времени между вводом сообщения в компьютер и появлением первого символа ответа на экране дисплея. Психологически и эргономически обоснованный выбор В.о.к. на различные запросы пользователя играет важную роль в обеспечении его эффективного использования (табл.). При

Значение времени ответа для различных действий пользователя

Характер действия	Определение времени ответа	Допустимый максимум, с
1	2	3
Ввод через клавиатуру	От нажатия на клавишу до начала ответа	0.1
	От нажатия на клавишу до появления символа	0.2
Инициация системы	От конца запроса до ответа	3
Вставление символов	От вставки до ответа	2
Выполнение простого запроса	От конца запроса до ответа	2
Выполнение сложного запроса	От конца запроса до начала ответа	5
Переворачивание страницы	От конца запроса до первых видимых строк	1
Сканирование страниц	От конца запроса до начала движения текста	0.5
Выбор функции	От выбора команды до ответа	2
Указание	От ввода указания до его появления	0.2
Манипулирование графикой	От конца запроса до начала ответа	10
Выполнение действий с помощью светового пера	От активации светового пера до ответа	1
Окончание формулировки проблемы	От конца запроса до ответа	10
Модернизация файла	От конца запроса до окончания	10
Сообщения об ошибках	От ввода до появления сообщения об ошибке	2

выделении временных параметров *процедуры* взаимодействия *пользователя* с компьютером необходимо стремиться к постоянству В.о.к. на однотипные запросы пользователя и к минимизации этого времени. При проектировании временных параметров *диалога*

необходимо учитывать, что время реакции человека в среднем составляет 2 с; В.о.к. должно соответствовать сложности и иерархии (цене) запроса; разброс В.о.к. должен быть минимальным (даже если для этого необходимо увеличить среднее время *ответа*); производительность системы должна с течением времени оставаться постоянной.

Н.И. Повакель.

ВУЙСОР — универсальная открытая интерпретирующая *авторская система*, предназначенная для создания диагностирующих и обучающе-контролирующих программ *учебного назначения* (ПУН) на персональных компьютерах типа IBM PC. Разработана в 1988 в Чешском политехническом институте (г. Прага). Состоит из трех частей: подсистемы управления обучением, подсистемы обработки результатов обучения и авторской подсистемы.

Подсистема управления обучением позволяет управлять процессом обучения как отдельного *обучаемого* (педагогический процессор), так и всего класса. Для первого применения используются модули: *принятия решения о следующем обучающем воздействии* на основе последнего *ответа* обучаемого и истории его взаимодействия с системой; *анализа ответа обучаемого* и сохранения протокола взаимодействия; сохранения (восстановления) текущего состояния обучаемого при завершении (возобновлении) сеанса работы с программой; вывода графической *информации*; генерации индивидуальных заданий. Модули, используемые для второго применения: регистрации обучаемых; обмена сообщениями между *преподавателем* и обучаемыми; наблюдения и модификации в реальном масштабе времени процесса обучения отдельных обучаемых. Теоретическую основу подсистемы управления обучением составляет *модель диалога "преподаватель - обучаемый"*.

Подсистема обработки результатов обучения содержит модули обработки протоколов взаимодействия обучаемых с системой для оценки их *знаний* и выработки рекомендаций по улучшению ПУН.

Авторская подсистема позволяет создавать ПУН без знания *программирования*. Основу ее составляют *текстовый редактор*, *графический редактор* и средства компоновки учебных материалов. Система имеет комплекс пособий для всех категорий *пользователей*: обучаемых, преподавателей, *авторов*.

Э.Мазак, Я.Полашек, Д.Полашкова.

ВЫСКАЗЫВАНИЕ ОБУЧАЕМОГО — *информация*, которая передается *обучаемым* программно-техническому средству *компьютерной технологии обучения* в процессе взаимодействия с ним. Посредством высказываний обучаемый сообщает системе обучения результаты мышления, *вопросы*, намерения и т.п. В зависимости от

способа управления *диалогом* В.о. разделяют на реакции и акции. Реакция — высказывания, которые были потребованы от обучаемого или инициированы средством обучения. Представляют собой *ответы* на вопросы, итоговые и промежуточные результаты решения *задач*, выбор из *меню*, предложенного системой. Их языковая форма определяется предыдущим требованием системы и зависит от возможностей *анализа ответов обучаемого* средством КТО. Акции — высказывания, передаваемые обучаемым по собственному почину. Среди акций выделяют директивы (*команды*), вопросы и сообщения. С помощью директив обучаемый требует у системы выполнения определенных действий, напр. вычислений по указанной в директиве формуле, вывода текстовой и графической информации на *периферийное устройство*. Вопрос может рассматриваться как директива о предоставлении определенной информации. Сообщения используются для пополнения *базы знаний* системы и ввода *данных* для последующей обработки с помощью директив.

Передача В.о. средством КТО реализуется с помощью технических средств ввода информации (*клавиатура*, графический ввод, *световое перо*, средства ввода речевой информации).

Ф. Анакер.

GUIDON — *экспертная обучающая система* по медицинской диагностике, разработанная в Стенфордском университете (США) в начале 80-х гг. Обучает решению *задач* так же, как *экспертная система* (ЭС) MYCIN, использующая в процессе вывода около 500 правил продукций (см. *Продукционные методы представления знаний*), обеспечивающих решение задач диагностики инфекционных заболеваний. *Обучаемому* предлагается история болезни пациента, диагноз заболевания которого он должен установить. Диагностика происходит в диалоге с системой, у которой можно получить дополнительные сведения о пациенте, а также подсказку и помощь. GUIDON поддерживает *диалог* со смешанной инициативой, т.е. управление обучением может исходить как от системы, так и от обучаемого (он сообщает о своих намерениях посредством *команд*).

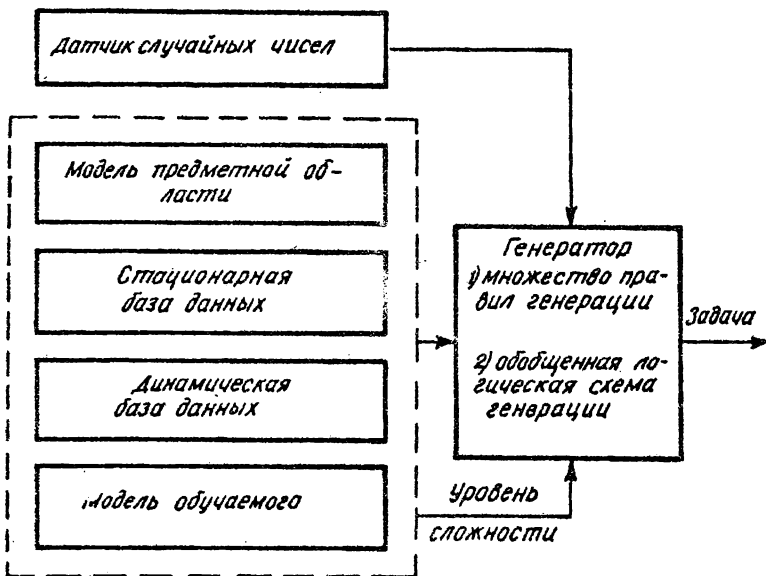
База знаний о предметной области GUIDON включает БЗ MYCIN, а также *данные* клинических и лабораторных исследований более чем 100 пациентов, используемые для постановки задач обучаемому. Каждое правило MYCIN снабжено так наз. фактором уверенности, позволяющим судить о степени достоверности того или иного заключения. Правила в GUIDON служат объектом изучения и предметом обсуждения, а также используются для получения эталонного решения задачи, предлагаемой обучаемому, и для сопоставления решений. Обучающая часть отделена от предметной и содержит 200 «обучающих» правил для выбора шаблонов диалога, для выбора и оценки значимости правил из БЗ о ПО, для обновления

модели обучаемого. Модель обучаемого — накрывающего типа, представлена подмножеством правил MYCIN, с каждым из которых связана тройка чисел в диапазоне от -1 до 1 . Первое число показывает, насколько система уверена в том, что обучаемый знает данное правило. Второе число отражает умение применить правило при диагностике конкретного случая. Третье число характеризует фактическое использование правила обучаемым при решении задачи для данного случая.

В целом GUIDON продемонстрировала плодотворность подхода к созданию обучающей части, основанной на правилах продукций в рамках консультационной парадигмы. Большое значение имеет независимость обучающей части от предметной, что позволяет рассматривать GUIDON как инструментальную систему обучения диагностике на основе ЭС, подобных MYCIN. Кроме того, в процессе работы над системой были выявлены недостатки MYCIN, препятствующие эффективному обучению. Главные из них заключаются в том, что правила ЭС отражают отдельные элементы знаний эксперта и вне связи друг с другом трудны для понимания и запоминания. Кроме того, важные с точки зрения обучения структурные характеристики ПО (иерархия данных и гипотез) и стратегические знания (метод вывода решения) в правилах явно не представлены. В последующих версиях в качестве базовой ЭС использован новый вариант MYCIN, наз. NEOMYCIN. Б.з. NEOMYCIN содержит метаправила, отражающие *стратегию* решения задачи, т.е. объяснение диагностики, предложенной системой, психологически более приемлемо для человека.

В.А. Петрушин, Е.М. Сеница.

ГЕНЕРАТОР ФОРМУЛ ИСЧИСЛЕНИЯ ПРЕДИКАТОВ ПЕРВОГО ПОРЯДКА — программный модуль, предназначенный для автоматического конструирования логических формул исчисления предикатов первого порядка. Работа генератора основывается на *модели предметной области*, а также *модели обучаемого*, определяющей уровень сложности (см. *Сложность задачи*) генерируемого задания (рис.). Для каждого уровня сложности определяется соответствующая группа формульных структур, на базе которых генератор, используя обобщенную логическую схему генерации и правила генерации логических формул, конструирует задание *обучаемому*. В качестве модели *предметной области* (ПО) используется концептуальное дерево; вершины его в общем случае представляют собой набор элементарных заданий (*задач*), которые должны быть усвоены обучаемым, а дуги указывают на возможность получения из них более сложных задач. Нижний уровень концептуального дерева — *стационарная база знаний* (БЗ) определяет набор предикатных символов, *символов* переменных, логических операций,



кванторов. В процессе генерации выбор элементов каждой из групп стационарной БЗ осуществляется случайным образом. Для этого используется датчик случайных чисел. В результате работы генератора формируется также динамическая база данных, представляющая собой концептуальное дерево с четко выраженной иерархией элементов и состоящая из множества сгенерированных логических формул, из которых можно сформировать формулы более высокого уровня сложности. Такая организация модели ПО позволяет иметь значительный запас учебной информации по изучаемой ПО. Если элементы стационарной БЗ и необходимое их число определены заранее, строго фиксированы и неизменны в течение всего процесса генерации, то элементы динамической БЗ не предсказываются заранее; известно только, что каждый из них обладает заданным уровнем сложности и числом составляющих элементов.

Процесс Г.ф.и.п.л.п. состоит из предварительного этапа, актуализации структуры и итогового этапа. На первом этапе по заданному уровню сложности выбирается соответствующая группа формульных структур. На втором — производится покомпонентная конкретизация структуры (в итоге — выбор одной из структур, определенных методикой обучения для данного уровня сложности) и параллельное заполнение конструируемого формата конкретными

значениями; напр., сгенерированы формулы более низкого уровня, символы связей, отрицания, кванторов. На итоговом этапе сгенерированные компоненты структуры последовательно выводятся в буфер и затем распечатываются в виде формул, предоставляемых обучаемому.

Л.Н.Гецко.

ГЕНЕРАЦИЯ ЗАДАЧ — процесс автоматического порождения множества задач определенного типа. Генерация множества вариантов задач необходима при организации *контроля знаний и адаптации* к индивидуальным характеристикам обучаемых. Процесс Г.з. основывается на *модели предметной области*, а также *модели обучаемого*, определяющей уровень *трудности задачи*. В общем случае Г.з. может осуществляться на основе совокупности шаблонов задач. Алгоритм порождения при этом разбивается на два осн. этапа: определение и выбор подходящего шаблона; порождение подходящей задачи выбранного типа.

Пусть m -число элементарных операторов, применимых при решении некоторого класса задач. Под сложностью задачи T будем понимать вектор

$$E(T) = (E_1(T), E_2(T), \dots, E_m(T)),$$

где $E_j(T)$ — число применений j -го элементарного оператора для решения задачи T . Задачи T_1 и T_2 наз. аналогичными, если $E(T_1) = E(T_2)$.

Шаблоном определяется класс аналогичных задач, имеющих для данного обучаемого одну и ту же сложность. Множество шаблонов разбивается на подмножества, предназначенные для обучения применению одного или нескольких элементарных операторов. В пределах одного подмножества шаблоны выбираются либо из числа задач, ранее не решенных данным обучаемым, либо по определенным критериям, учитывающим множество элементарных операторов, применяемых для решения задач по данному шаблону, скорость “забывания” знаний об элементарном операторе и т.п. Различают *генерацию концептуальных задач*, *генерацию логических задач* и *генерацию математических задач*.

М.Ю.Федосенко.

ГЕНЕРАЦИЯ КОНЦЕПТУАЛЬНЫХ ЗАДАЧ — генерация задач на определение или распознавание элементов *концептуальной модели* решения задач. Для Г.к.з. необходимо составить описание задачи, состоящее из трех осн. компонентов: описание *предметной области* (система понятий данной предметной области и система отношений

между этими понятиями), описание условий задачи (совокупность объектов и совокупность утверждений о них), описание цели (результата) решения задачи. Результатом такого представления задачи является осознание отсутствия определенных *знаний* (знаний о незнании). При первоначальном описании задачи цель задачи определена, как правило неявно — лишь описанием контекста условий, которым должен удовлетворять искомый элемент или вся система в целом. Поэтому при решении задачи появляется необходимость уточнения цели. При этом возникает новая задача — задача уточнения особенностей проблемной ситуации, т.е. задача опознавания (идентификации понятий). Смысл отдельного понятия удобно описывать с помощью семиотического четырехугольника (С-системы):

$$C = \langle N, M, P, O \rangle$$

где N — термин, имя понятия; M — модель понятия; P — предикат, содержание понятия; O — множество объектов, соответствующих данному понятию, объем понятия. В зависимости от того, как ставится задача, что известно и что требуется найти, можно выделить разные типы задач. Напр., при известных N и M найти P и O — определить понятие и привести соответствующий пример. Используя определенный *алгоритм*, можно автоматически генерировать требуемые типы задач.

М.Ю.Федосенко.

ГЕНЕРАЦИЯ ЛОГИЧЕСКИХ ЗАДАЧ — процесс автоматического порождения множества логических задач. К таким задачам относятся: распознавание высказываний естественного языка; символическое кодирование сложных высказываний, доказательство истинности сложных высказываний с помощью *операций* алгебры логики; конструирование правильно построенных логических формул; преобразование логических формул; минимизация логических функций и т.п. Язык алгебры логики — один из наиболее часто применяемых формальных языков во многих технических и гуманитарных дисциплинах. Напр., одним из типов задач является определение истинности сложных высказываний. Сложное высказывание представляется в виде логической формулы, значение истинности которой можно вычислить с помощью формальных приемов. При этом возникает необходимость формировать большое количество логических формул различной сложности. Множество правильно построенных логических формул можно генерировать двумя путями: блочным и посимвольным. При блочной генерации формулы порождаются последовательной конкретизацией исходной формулы — схемы формулы с помощью правила подстановки. При посимвольном

способе логическая формула порождается *символ* за символом в соответствии с синтаксическими правилами *грамматики* правильно построенных формул.

На базе генератора правильно построенных логических формул легко создаются генераторы других типов заданий. Напр., можно создать генератор задач минимизации логических функций, представленных в совершенной нормальной дизъюнктивной форме. При этом используется датчик псевдослучайных чисел с нормальным распределением. Число термов определяется случайным образом, а сами термы таким же образом выбираются из базы, содержащей все возможные термы.

Ю.И.Лобанов, М.Ю.Федосенко.

ГЕНЕРАЦИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ — генерация задач, для решения которых необходимо применить некоторый математический аппарат. В основном это задачи на вычисление или на доказательство. Процесс Г.м.з. основан на применении порождающих *грамматик*. Порождающая грамматика G состоит из конечного алфавита A , разделенного на две части — основной (A_m) и вспомогательный (A_a) алфавиты, конечного множества правил вывода P , представляющих собой продукции вида $S \rightarrow T$ (S, T — цепочки), и некоторого начального *символа* K ($K \in A$):

$$G = \langle A_m, A_a, P, K \rangle.$$

Формальный язык, порождаемый такой грамматикой, — это множество цепочек в осн. алфавите, выводимых из начального символа. Применение грамматик позволяет моделировать правильную структуру матем. выражения. Используя затем правила подстановки и заменяя отдельные символы выражениями, получают сколь угодно большое множество задач данного типа.

М.Ю.Федосенко.

ГЕНЕРАЦИЯ УЧЕБНЫХ ЗАДАЧ — автоматическое формирование задач, удовлетворяющих определенным дидактическим требованиям. Применяется в отдельных *программах учебного назначения*; позволяет упростить работу *преподавателя* по подготовке *учебных задач* и их решений для *обучения* и контроля решения задач определенного класса.

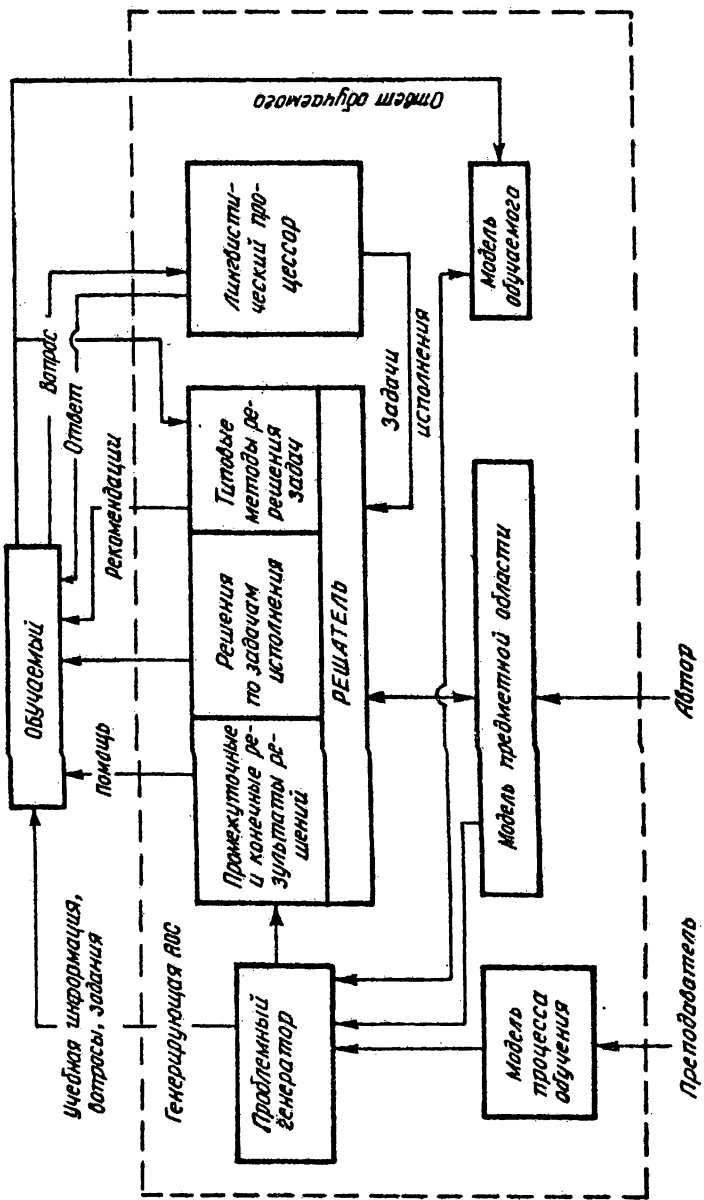
Первоначально индивидуализация заданий достигалась случайным выбором конкретной задачи из заранее подготовленного множества однотипных задач, а контроль решения сводился к сопоставлению ответа *обучаемого* с эталоном, подготовленным преподавателем. Для динамического выбора задач в разветвленных *обучающих программах*

приходилось прибегать к сложной системе нумерации задач или их дублированию. На следующем этапе формирование индивидуальных заданий обеспечивалось варьированием числовых значений параметров задач, а решение обучаемого сравнивалось с результатами, полученными при решении по программе задач данного класса. К настоящему времени сформировалась дидактико-методическая схема подготовки задач для контроля и тренировки, заключающаяся в следующем. Формулируется осн. задача, содержащая все необходимые параметры, которые определяют план ее решения. Для каждого параметра устанавливаются диапазон значений (в пределах его не изменяется план решения задачи) и допустимая величина шага изменения. Конкретная задача из данного класса формируется подстановкой фактических значений параметров (их можно получить с помощью генератора случайных чисел) в текст задачи.

Пример. Пусть квадрат накрыт полукругом, центр которого расположен на стороне квадрата. Необходимо вычислить площадь квадрата, оставшуюся ненакрытой. Параметрами в этой задаче служат длина стороны квадрата и радиус круга. Пока при вариации числовых значений параметров радиус круга не превосходит половины длины стороны квадрата, способ решения задачи не меняется, изменяются лишь числовые значения результата. Перемещение центра круга по стороне квадрата в определенных пределах также не приводит к качественным изменениям. Если ввести третий параметр, определяющий перемещение центра круга перпендикулярно одной из сторон квадрата, поиск оставшейся площади усложняется. Задачи из этого нового класса также решаются по одному плану и имеют одинаковую степень сложности. Если с помощью четвертого параметра определить угол поворота квадрата вокруг собственного центра, возникнет класс задач нового качества, в котором различные задачи решаются различными способами. Оба вышеописанных класса представляют собой частные случаи данного. Таким образом, поскольку изменение отдельных параметров может влиять на план решения полученной задачи, вариация параметров осуществляется с учетом дидактических принципов и индивидуальных данных о процессе обучения.

Э.-Н.Кунат.

ГЕНЕРИРУЮЩАЯ АВТОМАТИЗИРОВАННАЯ ОБУЧАЮЩАЯ СИСТЕМА (ГАОС) — автоматизированная обучающая система, формирующая и осуществляющая обучающую программу (или ее существенные компоненты — учебную задачу, вопросы, разъяснения, помощь и др.), направленную на достижение цели обучения, на основе знаний о предмете, пользователе, методах обучения. Работа ГАОС (рис.) основана на модели предметной области, которую формирует в памяти машины автор, специалист в этой предметной



области (ПО), владеющий необходимыми средствами *программирования* таких моделей. На основании обобщенного *плана обучения*, знаний о целях и методах обучения, представленных в модели процесса обучения, ГАОС, используя *модель обучаемого*, конструирует *план решения задачи обучения* для конкретного обучаемого, реализуя принцип индивидуального обучения (см. *Индивидуализация обучения*). Согласно этому плану проблемный генератор формирует и выдает обучаемому порции учебной информации, вопросы и задания требуемого уровня сложности обучения. Наличие проблемного генератора обеспечивает *генерацию задач* по определенной ПО и позволяет иметь значительный запас учебного материала, который конструируется в процессе обучения, а не вводится в систему заранее. *Решатель* контролирует правильность выполнения сгенерированных заданий путем автоматического их решения и сравнения с результатами обучаемого. В зависимости от назначения ГАОС решатель может производить вычисления, синтезировать *программу*, конструировать правильный *ответ* на сгенерированный вопрос и т.д. Наличие решателя позволяет обучаемому передать решение ряда задач исполнения системе. При необходимости обучаемому выдаются помощь и разъяснения: программа, конечные и промежуточные результаты решения задач; рекомендации по типовым методам решения задач. *Лингвистический процессор* обеспечивает согласованные двухсторонние *диалоги* на естественном языке с помощью знаний об изучаемом предмете, позволяет ГАОС отвечать на вопросы обучаемых по очередным порциям учебного материала, контрольным заданиям, всему материалу, представленному в модели ПО. Анализируя ответы и вопросы обучаемого, ГАОС корректирует его модель, накапливая *информацию* о ходе обучения, ошибках, их характере, причинах для подведения итогов взаимодействия на данном шаге обучения и конструирования сценария последующего диалога. ГАОС реализует также смешанно-иницируемый метод обучения, при котором система способна переорганизовать урок, изменить его содержание согласно требованиям обучаемого. Примеры ГАОС: *MALT, COACH, SCHOLAR*. Способность ГАОС автоматически решать задачу обучения с учетом индивидуальных особенностей обучаемого, самостоятельно принимать решения в непредвиденных ситуациях (напр., конструировать ответы на незапрограммированные вопросы), вести согласованные диалоги с обучаемым на естественном языке позволяет отнести эти системы к системам *искусственного интеллекта*.

Для создания ГАОС необходимо: разработать *алгоритмы* различных видов взаимодействия пользователя и ГАОС; выработать требования и создать *программное обеспечение* (инструментарий) для реализации этих алгоритмов в различных ПО, включая средства формирования модели ПО; распределить функции по реализации *процедуры* обучения между ГАОС, обучаемым, *преподавателем* и

автором модели ПО; установить соотношение между фиксированными и формируемыми частями обучающей программы, выбрать или разработать инструментальные средства программирования осн. блоков ГАОС; разработать типовые алгоритмы и программы для анализа ответов обучаемого, проблемного генератора и решателя, ориентированных на класс предметных областей.

Л.Н.Гецко.

ГЕТЕОФОРІЯ (от греч. *ετερι* — различный, *φερω* — стремиться) — скрытое косоглазие, являющееся следствием нарушения мышечного равновесия глаз, обеспечиваемого наружными мышцами глаза. При Г. бинокулярный характер зрения не нарушается. Различают Г. в горизонтальном направлении (экзофория, т.е. тенденция к отклонению глаза наружу, и эндофорию — внутрь) в вертикальном направлении (гиперфория — вверх, гипо- или катафория — вниз). Причинами Г. могут быть врожденные индивидуальные особенности строения глазных яблок прикрепления мышц, а также аметропии (см. *Рефракция*).

При работе с видеотерминалом Г. проявляется в мышечной *астенопии*, признаками которой являются головные боли, быстрое утомление, иногда — периодическое двоение изображения. Г. возникает чаще при наличии астигматических аметропий и неправильной коррекции зрения без учета расстояния фиксации до объекта.

В.Г.Мартиророва.

ГОМОМОРФІЗМ (от греч. *ομοσ* — общий и *μορφη* — форма) — отношение между системами *A* и *B*, состоящее в подобии их структур. При этом каждому компоненту системы *A* соответствует определенный компонент системы *B* и каждому отношению между компонентами системы *A* — определенное отношение между компонентами системы *B*. Систему *B* называют гомоморфным образом системы *A*. Напр., географическую карту можно считать гомоморфным образом изображенной на карте территории; при этом предполагается, что в число компонентов карты условно включен “пустой” (не выражающийся в к.-л. изображении) компонент, и именно он соответствует некоторым компонентам территории. Территория, однако, не является гомоморфным образом карты, поскольку одинаковым обозначениям на ней могут соответствовать разные объекты на территории. Частный вид Г. — *изоморфизм*. См. также *Модель*.

Г.А.Балл.

ГРАММАТИКА (греч. *γραμματική*, от *γραμμα* — литера, написание) — раздел лингвистики, изучающий закономерности образования и употребления форм слов. Иногда под Г. понимают общие закономерности функционирования языка и науку, изучающую эти

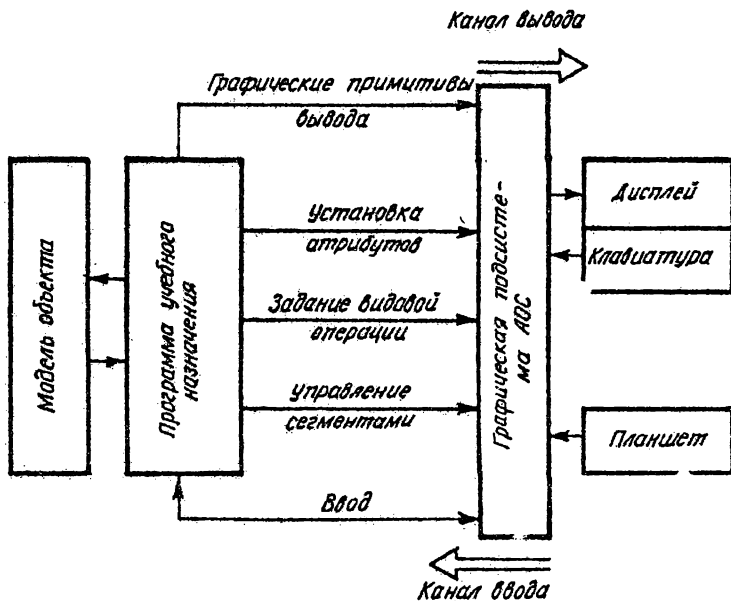
закономерности, что означает отождествление Г. с лингвистикой — наукой о языке. Г. состоит из двух разделов: морфологии (исследование внутренней структуры слов) и синтаксиса (правила сочетания слов в предложении). Американский ученый Н.Хомский ввел в лингвистику понятие формальной Г. — системы правил, позволяющих строго, развернуто и понятно описывать некоторые грамматические закономерности как естественных, так и *искусственных языков*.

Н.А.Власенко.

ГРАФИЧЕСКАЯ ПОДСИСТЕМА автоматизированной обучающей системы — часть программного обеспечения *автоматизированной обучающей системы* (АОС), предназначенная для изображения объектов на экране дисплея и обеспечения связи *программ учебного назначения* (ПУН) с диалоговыми графическими устройствами с целью организации интерактивного графического взаимодействия *пользователя* с компьютером. Функциональные возможности Г.п. определяют уровень использования графики в ПУН и могут колебаться от достаточно простых до весьма сложных, позволяющих формировать изображение трехмерных объектов и реалистические изображения.

Как правило, *операции* Г.п. подразделяются на шесть классов: графические примитивы вывода (*подпрограммы*, формирующие изображение элементарных объектов на видовой поверхности *дисплея*); установка атрибутов — совокупность *программ*, определяющих вид всех последующих графических примитивов; управление сегментами — совокупность программ, предназначенных для группирования логически связанных примитивов в сегменты, т.е. миним. единицы изображения, доступные выборочному изменению; видовая операция — пара функций, в совокупности определяющая изображаемую область в мировой системе координат, и часть экрана, в которую эта область будет отображаться; ввод — набор программ, управляющих взаимодействием *обучаемого* с ПУН; управление — программы, обеспечивающие инициализацию Г.п., ее сброс и другие вспомогательные функции (напр., очистку экрана, закрытие графических устройств и т.д.).

Г.п. АОС организует взаимодействие *обучаемого* с ПУН посредством двух каналов: вывода и ввода (рис.). Канал вывода, последовательно преобразуя описание объекта во все более машинно-зависимые представления, в конечном счете обеспечивает построение изображения на экране. Исходная *информация* о геометрии объекта (расположение) и его топологии (связность) извлекается ПУН из описания *модели* объекта и преобразуется в последовательность вызовов подпрограмм Г.п., которая и осуществляет соответствующие построения на экране. Осн. задача канала ввода Г.п. АОС состоит



в том, чтобы обеспечить передачу данных от обучаемого в ПУН. Эти данные изменяют состояние или логику ПУН и могут инициировать модификацию модели объекта и/или его изображения на экране (напр., перемещение). Эффективность работы каналов ввода и вывода Г.п. АОС играет определяющую роль в обеспечении психологического комфорта при работе обучаемого с компьютером. Разрыв цикла взаимодействия обучаемого с компьютером из-за заметных задержек ответа (превышающих доли секунды) приводит к фактическому снижению дидактической эффективности *компьютерного обучения* с применением графики. В связи с этим одним из осн. требований при разработке Г.п. АОС является обеспечение немедленной реакции на действия обучаемого, т.е. создание такой *обратной связи в обучении*, которая не приводила бы к нарушениям в координации глаз и рук человека.

В.И.Отенко, М.Тагарински.

ГРАФИЧЕСКИЙ РЕДАКТОР — программа, предназначенная для обработки графической информации в интерактивном режиме. В качестве графических изображений выступают геом. объекты различной сложности — от элементарных геом. фигур типа точки, отрезка прямой, ломаной линии, окружности, дуги, эллипса до трехмерных изображений.

Г.р. позволяет создавать и модифицировать графические изображения различными способами: строить и трансформировать геом. фигуры; заполнять (закрашивать) объекты и накладывать их друг на друга; осуществлять динамическое перемещение объектов по экрану монитора, их вращение и масштабирование; копировать объекты в различные позиции монитора; компоновать изображения с использованием библиотек ранее созданных объектов или элементарных графических фигур. При создании изображений для каждого из объектов обычно обеспечивается возможность выбора толщины линий фигур, цветовой палитры для построения изображений, применение различных "инструментов" рисования (карандаша, кисти, ластика). Допускается увеличение отдельных фрагментов изображений с целью внесения тончайших изменений при помощи инструмента, наз. "лупой". Г.р. обеспечивает возможность наложения на графические изображения текстовой информации. Для этого в его состав могут входить как встроенные, так и загружаемые из внешней памяти ЭВМ наборы шрифтов. Допускается расположение текстовой информации на изображениях по горизонтали, по вертикали или под углом. Возможности Г.р. определяются набором поддерживаемых им аппаратных средств ввода-вывода графической информации, входящих в состав компьютера. Для ввода информации помимо клавиатуры, можно использовать специальные графические устройства ввода — мышь, шар, световое перо, джойстик (для "ручного" ввода), а также сканирующие системы для ввода цветных или тоновых изображений. Для вывода изображений используются векторные и растровые дисплеи, печатающие устройства и графопостроители для создания твердых копий. Качество графических изображений зависит от степени разрешающей способности устройств вывода, а также их цветовых характеристик. Графическую информацию можно сохранить или получить из внешнего запоминающего устройства. Напр., при сохранении экрана (если используется растровый дисплей) в файле может сохраняться битовая карта изображения, полученная из видеопамяти компьютера. В дальнейшем этот файл может быть использован как Г.р., так и прикладными программами (если Г.р. применяет графические стандарты).

В настоящее время разработано несколько стандартов машинной графики. Наиболее распространен стандарт GKS (Graphical Kernel System — ядро графической системы), определяющий совокупность графических функций, используемых прикладными программами. Для обеспечения дружественного интерфейса с пользователем в Г.р., как правило, применяется иерархическая система меню с использованием пиктограмм — условных графических обозначений выбираемых действий, геом. фигур, инструментов для рисования и т.п.. Некоторые Г.р. позволяют создавать не только статические графические изображения, но и мультипликационные. Для этих целей используют различные методы; один из них — последователь-

ное отображение на экране небольших кадров, каждый из которых можно получить из созданного фона ("режим слайдов"), т.е. при этом содержимое всего экрана "разрезается" на несколько кадров; более сложный метод — движение по относительно неподвижному фону нескольких, как правило прямоугольных, объектов с привлечением разного рода графических эффектов и звукового сопровождения. Г.р. может быть как автономной программой, так и частью интегрированной системы.

В компьютерной технологии обучения Г.р. начали широко распространяться с середины 70-х гг. в связи с появлением персональных электронных вычислительных машин. С их помощью осуществляется подготовка слайдов, а также мультипликационных сюжетов, демонстрирующих суть предмета, излагаемого преподавателем.

А.В. Чигорев.

ГРАФОПОСТРОИТЕЛЬ, плоттер — устройство вывода информации из электронной вычислительной машины в графической форме (в виде схем, чертежей, графиков, трафаретов, шаблонов и др.) на специальные носители информации (бумагу, пленку). Различают Г. по способу формирования изображения — электромеханические (координатные), растровые и электронные, а также по способу управления — с автономным управлением, управлением от ЭВМ и универсальным управлением.

Электромеханические Г. подразделяют на планшетные и барабанные. Формирование изображения в планшетных Г. осуществляется перемещением регистрирующего органа по осям X и Y при неподвижном носителе (листе бумаги). В барабанных Г. изображение формируется перемещением регистрирующего органа вдоль оси барабана; носитель (рулон бумаги) перемещается вследствие вращения барабана. Регистрирующим органом планшетных и барабанных Г. служат шариковые стержни, карандаши, резцы, перья с чернилами и др. Узел записи может иметь одно или несколько перьев различного цвета. Разновидность электромеханических Г. — координатографы, отличающиеся повышенной точностью и способом нанесения изображения (вырезание, гравирование, экспонирование с помощью фрезерных режущих головок, фото головок с набором диафрагменных дисков и других инструментов).

Широко распространены быстродействующие растровые Г. с электростатическими, электрохимическими и другими немеханическими способами регистрации изображений на обычные и специальные носители. Для управления электродами используются выходные регистры устройства управления. Перед выводом в растровый Г. изображение (в ЭВМ хранится в виде таблицы чисел и математических выражений, им соответствуют отрезки прямых и

фрагменты кривых линий) специальным процессором преобразуется в последовательность строк, разрядность которых соответствует числу электродов в пишущем узле. В процессе вывода каждая очередная строка (координата Y) управляет подачей напряжения на соответствующие электроды пишущего узла. Одновременно осуществляется синхронное движение носителя по координате X . Эти устройства имеют высокую разрешающую способность — 5-10 линий/мм.

Электронные Г.(графические дисплеи) строятся на базе электронно-лучевых трубок. Характеризуются высокой скоростью (до 200 м/с), большой разрешающей способностью (10-15 линий/мм), высокой надежностью, возможностью получения значительного числа копий, а также возможностью микрофильмирования. Изображение на экране формируется следящим или растровым методом. При следящем методе на отклоняющую систему электронно-лучевой трубки подается напряжение, пропорциональное координатам выводимых точек. При растровом методе электронный луч сканирует поле экрана с определенным шагом и засвечивает его в тех точках, координаты которых совпадают с координатами точек графического изображения. Для получения копии изображения в электронных Г. используют оптические и электростатические методы регистрации на светочувствительную или обычную бумагу, реже записывают на электрохимические и термопластические материалы. Микрофильмирование перспективно для автоматизации изготовления, хранения, поиска и тиражирования документации.

В.П.Козлова, З.Ш.Примакова.

ДААННЫЕ — информация, фиксированная в определённой форме, пригодной для последующей обработки, хранения и передачи. Воспроизводят изображения и представления реального мира. В отдельных областях реального мира можно, не прибегая к физическому моделированию, представлять его с помощью матем. формул, которые дают возможность прогнозировать результаты при изменении переменных в соответствии с возможными событиями в реальном мире. При работе с Д. и информацией различают инфологический и датологический аспекты. Инфологический аспект применяется при рассмотрении вопросов, связанных со смысловым содержанием Д., независимо от способов их представления. Существует так наз. инфологическое проектирование, при котором выделяется часть реального мира (*предметная область проектируемой системы*), определяющая информационные потребности этой системы. Датологический аспект используется при рассмотрении вопросов представления Д. в памяти *информационной системы*. При датологическом проектировании системы, исходя из возможностей имеющихся средств восприятия, хранения, обработки информации, разрабатываются соответствующие формы представ-

ления информации в системе посредством Д. Приводятся также модели и методы представления и преобразования Д., формулируются правила их смысловой интерпретации.

Для дальнейшего использования Д. необходимо знать их смысловое содержание — семантику Д. Поэтому при работе с Д. необходимо сформулировать правила их смысловой интерпретации. Осн. средством представления семантики Д. является естественный язык, однако используются также формализованные языки, позволяющие эффективно организовывать обработку Д. с помощью вычислительной техники и представить необходимую семантику Д., удовлетворяющую практическую потребность целого ряда прикладных задач.

Каждому средству обработки Д. соответствует своя форма представления Д. Совокупность свойств, отличающая некоторый набор Д. и определяющая множество значений, которые они могут принимать, наз. типом данных. См. также *Данные о процессе обучения*.

А.П. Ильяшенко, М.Е. Козлов.

ДА́ННЫЕ О ПРОЦЕ́ССЕ ОБУЧЕ́НИЯ — информация о ходе и результатах обучения, которую можно собрать и/или обработать в автоматизированных обучающих системах (АОС). В отличие от других групп данных в АОС (напр., данных обучающей программы или данных об истории обучения), Д. о п.о. собираются или образуются только непосредственно в процессе обучения на базе компьютера. Подразделяются на учебные, актуально-управляющие и внутрисистемные.

Учебные Д. о п.о. делятся на первоначальные и производные. Первоначальные учебные данные можно получить только из самого обучающего процесса, т.е. они не реконструируются из других данных. Эти данные характеризуют деятельность *обучаемого*. К ним в первую очередь относятся *высказывания обучаемого* в связи с соответствующей ситуацией *учения*, а также временные характеристики его деятельности (время начала и/или окончания высказывания). В отдельных случаях можно собрать данные о психофизиологических и других индивидуальных характеристиках. Производные учебные данные образуются при соотношении первоначальных учебных данных с другими имеющимися в распоряжении группами данных (прежде всего данными обучающей программы) некоторым способом, определённым *автором обучающей программы или обучающей системой*. Широко распространёнными способами получения производных данных являются отнесение классов ответа к оценкам результатов обучения и их накопление, а также формирование данных о частоте использования системных средств или проработки определённых частей обучающей программы. Если в результате

обработки большого объёма различных данных по сложным алгоритмам образуются компактные производные данные, их наз. модельными. Актуально-управляющие Д. о п.о. передаются преподавателем в процессе обучения на базе компьютера. Они содержат все не определённые обучающей программой управляющие меры, вытекающие из актуального наблюдения и анализа деятельности учения. К внутрисистемным Д. о п.о. относятся все генерируемые системой данные, поскольку они не однозначно реконструируются из других групп данных. Это прежде всего относится к данным, получаемым с помощью генераторов случайных чисел. Поскольку затраты на реконструкцию высоки, внутрисистемными считаются и данные, возникающие при работе обучаемого с системными и прикладными программами и автоматически передающиеся обучающей системе.

Д. о п.о. используются, с одной стороны, для управления учением и оценки его результатов и являются существенной основой для выбора индивидуальных обучающих воздействий. Для создания адаптивных обучающих программ используются учебные и внутрисистемные Д. о п.о. С другой стороны, собираются и накапливаются выбранные Д. о п.о. для решения организационных вопросов (напр., определения точки возобновления работы после сбоя системы или прерывания работы) и составления так наз. протоколов на машинных носителях для дальнейшего анализа хода учения. С помощью протокола осуществляется по возможности полная реконструкция взаимодействия между обучаемым и системой (см. *Взаимодействие человек-машинное*) и анализ в различных аспектах (в т. ч. обратная и детальная информация для преподавателей и обучаемых, информация для апробирования обучающих программ и научных исследований). При работе в локальной сети учебного назначения можно использовать выбранные подмножества накопленных учебных и внутрисистемных Д. о п.о. так, чтобы составить актуальную обзорную информацию для преподавателя и по его запросу вывести её на экран дисплея.

В.-Х.Хартвиг.

ДЕКЛАРАТИВНОЕ ПРЕДСТАВЛЕНИЕ ЗНАНИЙ — метод представления знаний, который основывается на некотором универсальном множестве процедур, обрабатывающих утверждения для рассматриваемого класса задач, и на множестве независимых утверждений, описывающих предметную область конкретной задачи. Характеризуется независимостью утверждений, т.е. простой модификацией; доступностью для восприятия; простым и лёгким конструированием информационных единиц. При Д.п.з. нет необходимости указывать, как используется то или иное утверждение (в этом

преимущество данного способа по сравнению с *процедурным представлением знаний*). Простые утверждения можно использовать многократно, поэтому не требуется заранее описывать все возможные способы их применения. Это свойство Д.п.з. обеспечивает его гибкость и экономичность. Преимуществом Д.п.з. является также возможность модификации *знаний* и *обучения* новым знаниям простым добавлением или устранением утверждений (поскольку знание рассматривается как множество независимых фактов). Для процедурного представления проблема модификации усложняется, т.к. необходимо описание, как использовать данное утверждение. Выбор представления — декларативного или процедурного — обусловлен не только соображениями эффективности. Так, неполнота знаний об окружающем мире не позволяет описать любые явления в виде процедур, а сильная связанность некоторых явлений не даёт возможности использовать Д.п.з. В связи с этим разрабатываются модели *представления знаний*, соединяющие в себе достоинства обоих типов знаний.

К Д.п.з. относятся логические, сетевые (см. *Семантическая сеть*) и *фреймовые методы представления знаний*. Логические методы основаны на совокупности логических формул, с помощью которых строится *модель предметной области*. Специфика этих методов состоит в наличии правил вывода, определяющих процедуру доказательства теорем в данном *исчислении логическом*. Здесь наибольшее распространение получили пропозициональная логика и исчисление предикатов. Правила вывода в логических системах используются для анализа *базы знаний* (БЗ) и для вывода *новой информации* из имеющейся. На каждом шаге вывода особое значение приобретает оптимальный порядок используемых правил, что может быть достигнуто консультацией с *пользователем*. Для соединения процедурного и декларативного методов представления знаний (в результате которого в итоге повышается эффективность БЗ) используются *семантические сети* и *фреймы*.

В *автоматизированных обучающих системах* (АОС) целесообразно использовать Д.п.з. при описании предметных областей со сложной структурой, являющихся объектом изучения и требующих большого количества логических выводов, напр. для определения значений некоторых параметров, а также объектов, легко моделируемых в логике предикатов первого порядка. Сетевые модели для АОС представляют интерес как средство описания сценариев обучения. Они также широко используются для реализации естественных языковых *интерфейсов* между *обучаемым* и АОС. Примерами использования сетевых представлений могут служить *операционная система "Географ"* и *экспертная обучающая система САКИО*.

О.Н.Золотун, В.В.Колос.

ДЕКЛАРАТИВНЫЙ ЯЗЫК — *алгоритмический язык*, предоставляющий средства описания, достаточные для автоматической генерации *алгоритма* решения задачи. Технологическая направленность Д.я. позволяет широко использовать их для проектирования и спецификации программ (LOBO и др.), а также в качестве языков программирования (Пролог, LOBO и др.). По проблемной ориентированности Д.я. разделяются на проблемноориентированные (LOBO — язык для математиков), универсальные (Пролог) и машинноориентированные (Пролог для ПРОЛОГ-машины).

В.А.Третьяк.

ДЕЛОВАЯ ИГРА — *игра*, целью которой является поиск управленческих решений в условной проблемной ситуации. Может иметь широкий диапазон организационных форм: от ситуаций, в которых несколько специалистов свободно, без особых правил, разыгрывают и обсуждают определённые научно-технические проблемы, до сложнейшей многодневной имитации деятельности целой народнохозяйственной отрасли (см. *Имитационная игра*), где принимают участие десятки и сотни людей, а для управления игрой создаётся специальный штаб (рабочая группа). В Д. и. сочетаются игровой и неигровой (научный, учебный, проектный и др.) подходы. Цели Д.и. тесно связаны с функциональными характеристиками, которые и задают аспекты существования игры: её можно представить как особую исследовательскую модель (познавательный аспект), как форму игрового сознания (игровой аспект) и как самостоятельный вид деятельности (институциональный аспект).

С самого начала своего формирования (первые игры — 30-е гг.) Д.и. используются в целях обучения и повышения квалификации специалистов. В настоящее время Д.и. в этом аспекте рассматриваются как один из активных методов обучения, характеризующийся: а) наличием проблемы управления социально-экономической или социально-психологической системой и/или моделирования профессиональной деятельности; б) наличием общих целей игровых коллективов; в) наличием ролей и назначением на них участников игры; г) различием интересов участников и учётом условий неопределённости (напр., вероятностного характера деятельности); д) принятием и реализацией определённой последовательности решений в процессе игры, каждое из которых зависит от решения, принятого данным участником на предыдущем этапе, и от действий других участников; е) наличием достаточно мощной системы стимулирования, обеспечивающей мотивацию к деятельности, подчинение интересов участников интересам коллектива, а также объективную оценку личного вклада каждого участника; ж) объективностью оценки результатов игровой деятельности. Как инструмент познания Д.и. обладает важной особенностью: её познавательные средства в большей

степени, нежели в других научных методах исследования, включают человеческую субъективность; по сравнению с иными методами обучения Д.и. обеспечивает более высокую мотивацию к деятельности и "погружение" *обучаемого* в среду, которую необходимо освоить и где ему предстоит принимать решения. Реализация форм игрового сознания в Д.и. происходит прежде всего благодаря построению такой реальности, в которой можно разыграть осн. конфликты и проблемы, относящиеся к данной деловой сфере. Само по себе игровое сознание в плане "серьёзной деятельности" не даёт преимуществ, но в соединении с моделирующей функцией становится эффективным инструментом решения деловых проблем и задач: произвольность и условность игры не только не противоречат познавательному отношению, но, наоборот, усиливают его за счёт расширения моделирующих возможностей в пограничных областях. Д.и. — не только исследовательская модель и форма игрового сознания, но и самостоятельный вид деятельности, имеющий ряд специфических организационных форм.

В Д.и. важную роль играет рефлексия, полагающая предметом осмысления саму игру и приводящая к уточнению её компонентов как игры. Д.и. имеет много свойств, присущих любой игре (условность, особая знаковая система и пр.), но у неё есть и свои отличительные особенности как у "серьёзной" игры, сближающейся с "делом" и обуславливающей социально значимые результаты в "сфере серьёзного". Применение компьютеров позволяет реализовать наиболее сложные Д.и., т.к. быстродействие компьютера позволяет сразу получить расчёт вариантов различных принятых решений, продемонстрировать последствия принятых решений в удобной для игрока форме привычного документа, графики или изображения ситуации.

Э.Л.Ивахненко, Е.Д.Маргулис.

ДЭСТА (диалоговый естественноречевой синтезатор алгоритмов) — интеллектуальная интегрированная технологическая система, обеспечивающая пользователям-программистам инструментальную поддержку для высокопроизводительной разработки комплексов структурированных программ по естественноречевой (ЕЯ) спецификации их функционального назначения. Конкретное наполнение *базы знаний* (БЗ) ориентировано на следующие языки программирования (ЯП): Турбопаскаль, Модула-2, Си. В процессе создания прикладного программного обеспечения важное место занимает этап разработки спецификаций программ. В системе Д. спецификации разрабатываемых программ (программных модулей) записываются на специальном языке ФОРА (Формализованное Описание Разрабатываемых Алгоритмов), применяемым для записи ЕЯ-предложений. Эта система осуществляет автоматический поиск в

БЗ по ЕЯ-описанию уже созданных программных модулей (в виде загрузочных модулей, процедур, функций или команд операционной системы) и автоматическое их включение в спецификацию разрабатываемого программного модуля. Система Д, обеспечивает автоматическое преобразование спецификации разрабатываемого модуля в структурированный прототип программы на ЯП, обеспечивает доступ к *текстовому редактору* и интегрированной среде без выхода из системы (в режиме "on line"). Все режимы системы сопровождаются контекстными пояснениями. *Пользователю* предоставляется возможность настраивать БЗ (на уровне ЕЯ-описания модулей) на привычную ему терминологию и пополнять БЗ новыми модулями и их описаниями, повышая тем самым "интеллектуальный" уровень системы. Пользователь может создать командный язык для быстрого вызова часто используемых программных модулей и команд *операционной системы*. Система может работать в *информационно-справочном режиме*. Задавая ЕЯ-команды (имена ЕЯ-операторов), пользователь получает полный перечень соответствующих ЕЯ-операторов и поддерживающих их программных модулей, которые, по желанию пользователя, можно тут же выполнить. Специфицируя с помощью ЕЯ-оператора функциональное назначение искомого или подлежащего разработке модуля, пользователь получает описание исходных модулей и в режиме меню может немедленно ознакомиться с их работой. Задавая любые варианты масок (шаблонов) имен *файлов*, пользователь практически мгновенно получает перечень соответствующих этим маскам файлов с их характеристиками. В любой момент времени он может просмотреть содержимое любого файла и/или выполнить активные файлы. В системе реализован многооконный принцип работы. В процессе работы системы на экране *дисплея* может одновременно находиться несколько окон, предназначенных для выполнения объектов разных типов: программных и текстовых файлов, рабочих каталогов, управляющих меню, атрибутов БЗ, информационных сообщений и пояснений. Окна также можно использовать и для ввода различной *информации*, запрашиваемой системой у пользователя.

В.А.Ловицкий

ДЖОЙСТИК (англ. joystick, joy — удовольствие, игра, stick — осозание, удар) — см. *Манипулятор ручной*.

GPM (General Purpose Macro Generator — макрогенератор общего назначения) — универсальный *макропроцессор*, предназначенный для замены одной последовательности *символов* другой. В качестве ввода берет исходный *текст* и порождает выходной текст, отличающийся от входного тем, что из него исключаются *макроопределения*, и

вместо *макрывывозов* по определенным правилам подставляется заменяющий текст. Макроопределения и макривывозы в GPM идентичны по синтаксической структуре, но отличаются семантически. В описаниях, в равной степени относящихся к макроопределениям и макривывозам, будет употребляться слово макрос.

В GPM три осн. конструкции, соответствующие: вызову макроса (%); подстановке аргумента (#); представлению строки в виде литерала. Знак % резервируется исключительно для выделения макросов из остальной части текста. Фактически % и; могут рассматриваться как пара скобок, выделяющая любой макрос.

Макривывоз имеет вид: %<имя>[,<пар>....,<пар>]; где <имя> — имя макроопределения; <пар> — фактический параметр. Макроопределение обеспечивается специальным системным макросом DEF, имеющим два аргумента — имя определяемого макроса и заменяющий текст. Макроопределение представляет собой лишь вызов другого макроса.

Заключение входной строки в специальные скобки [и] запрещает вычисление строки и означает то, что строка должна быть явно скопирована. Скобки предохраняют от трактовки литеры % как начала макроса, а # — как подстановки параметра. Макросы (как определения, так и вызовы) могут содержать друг друга.

При обработке текста GPM просматривает вводной поток слева направо и копирует в выходной поток, пока не встретится макривывоз. Макривывоз вычисляется. Результат вызова копируется в выводной поток, замещая при этом сам макривывоз, после чего возобновляется просмотр предыдущего вводного потока. Если во время вычисления имени или параметра встречается макривывоз, он также сразу вычисляется. Следовательно, обработка макривывоза — рекурсивный процесс.

Приведем пример, демонстрирующий некоторые возможности макросов.

```
%DEF,SUC,[%1,2,3,4,5,6,7,8,9,10,%DEF,1,[#]#1;];
```

Результатом вызова приведенного выше макроса будет первый аргумент (цифра от 0 до 9), увеличенный на 1.

Вызов: %SUC,7;

Результат генерации: 8.

Ю.А.Загура.

ДИАГНОЗА ЗАДАЧА — обнаружение причины некоторых наблюдаемых явлений. В *компьютерной технологии обучения* решается при определении причины ошибок или пробела в знаниях *обучаемого* с целью применения адекватного регулирующего воздействия в виде дополнительной или разъясняющей информации. В теории задач Д.з. — задача совершенствования знаний, в актуальном состоянии которой можно выделить в качестве известного

некоторое конечное (наблюдаемое) состояние объектов, а из начальное состояние — в качестве неизвестного. Примером эффективного решения Д.з. может служить система *BESS*, реализующая вероятностный метод принятия решений вычислением вероятностей некоторых гипотез (причин) на основе оценки вероятностей определенных симптомов (наблюдаемых явлений).

А.М.Довгялло

ДИАГНОСТИКА ЗНАНИЙ — процесс целенаправленного рас-
почаивания параметров концептуальной модели знаний обучаемых
Для Д.з. применяют различные *диагностические тесты*. *Стратегии*
Д.з. наз. безусловной, если очередной проверочный шаг не зависит
от результатов предыдущих проверок, и условной, если в диаг-
ностическом дереве хотя бы двум вершинам некоторого ранга
соответствуют разные тестовые проверки. При условной стратегии Д.з
последовательность проверок зависит от результатов предыдущих
проверок. В этом случае необходимы правила выбора очередной
проверки, которые могут задаваться таблицами принятия решений
или схемами условных алгоритмических предписаний. По виду
правил окончания проверок различают стратегии Д.з. с безусловной
и условной остановкой. В первом случае проверка заканчивается
после реализации всех шагов тестирования, во втором — при
достижении некоторых фиксированных состояний.

Ю.И.Лобанов.

ДИАГНОСТИЧЕСКИЙ ТЕСТ — совокупность заданий (*вопро-*
сов), предназначенных для *диагностики знаний*. Д.т. различают по
правилам выбора проверочных действий и окончания проверок.
Каждая элементарная проверка (шаг теста) t позволяет разбить
множество состояний P модели обучаемого на k классов. Число k
соответствует количеству различных результатов шага t , т.е.
результат проверки t используется в качестве признака клас-
сификации возможных состояний знаний обучаемого. Алгоритм
диагностики представляется в виде ориентированного графа; про-
стейшим представлением служит диагностическое дерево. Каждой
вершине дерева ставится в соответствие пара чисел (p, q) , где p —
ранг вершины (число дуг пути из начальной вершины в данную);
 q — номер вершины в ранге. Каждой вершине (p, q) ставится в
соответствие шаг проверки t и подмножество состояний модели P_{pq} .

Ю.И.Лобанов.

ДИАЛОГ (греч *dialogos* — разговор, беседа) — попеременный
обмен репликами (в широком смысле репликой считается и ответ в
виде действия, жеста, молчания) двух и более людей; форма

взаимодействия человеко-машинного, приближающаяся к свободному речевому общению людей при совместном решении некоторой задачи. Среди специалистов в области информатики и вычислительной техники существует также упрощённое (метафорическое) понимание Д. лишь как процесса достаточно быстрого обмена сообщениями в режиме непосредственной работы на *персональной электронной вычислительной машине* или *системе коллективного пользования*. Д.- один из наиболее удобных способов взаимодействия двух партнёров — человека и машины. В *компьютерной технологии обучения* (КТО) Д. играет важнейшую роль, поскольку эффективный процесс *обучения* по своей структуре диалогичен, причём быстрота и качество усвоения материала *обучаемым* существенно зависят от диалоговых возможностей, которые может предложить компьютер. Иными словами, КТО во многом определяется свойствами Д., реализованного в *обучающей системе*.

Д. как речевое общение при совместном решении задачи предполагает: наличие у партнёров взаимосвязанных целей, (совпадающих, согласованных или конфликтных); взаимопонимание партнёров, выражающееся как в знании обоими некоторого языка общения, так и в наличии у обоих некоторых *знаний* о предмете Д.; определённое (более или менее равноценное) распределение между партнёрами функций по совместному решению задачи (включая двустороннее или смешанное управление взаимодействием, двустороннюю инициативу и др.); достаточно быстрый темп обмена сообщениями, естественный для обычного Д. между людьми. При Д. партнёры стремятся получить помощь или чему-то научиться, либо, наоборот, предоставить некоторую *информацию* или обучить некоторым действиям. Это общие, так наз. отдалённые цели, иницирующие Д., в процессе же его возникают и детализируются разнообразные ближайшие цели (см. *Структура диалога*), достижение которых является средством продвижения к более отдалённым. Выделение и структуризация целей, возникающих у партнёров, позволяют определить и классифицировать типичные фрагменты Д. и диалоговые режимы.

Языки общения пользователей *диалоговых систем* прошли несколько этапов развития — от простых типа “да — нет” и “меню” до относительно свободного общения на *естественном языке ограниченного*. Наряду с усложнением языка общения усложнялись и способы представления в компьютерах *знаний* о предметной области (ПО) общения. Наиболее простым классом существующих диалоговых систем являются системы, базирующиеся на итеративных *алгоритмических языках* (языках операторного типа). Поддержка их осуществляется специальными средствами матем. обеспечения — возможностью работы в режиме Д. в рамках соответствующей *операционной системы*.

Следующий шаг на пути приближения компьютера к *пользователю* — диалоговые системы, опирающиеся на жёсткий информационный язык. К этому классу систем относятся многочисленные *информационно-поисковые системы* с языками дескрипторного типа (см. *Авторский язык программирования*). Принцип распознавания сообщения-запроса по ключевым словам благодаря его универсальности и простоте используется также и в диалоговых системах со слабо формализованным или нормализованным естественным языком (ЕЯ). Системы этого типа способны понимать ограниченное число предложений ЕЯ определённой структуры.

Дальнейшим шагом в развитии систем, понимающих ЕЯ, стали *лингвистические процессоры*, способные осуществлять *синтаксический анализ, морфологический анализ и семантический анализ* входного сообщения и ориентированные на работу с конкретной ПО. К этому типу систем относятся широко известная *диалоговая информационно-логическая система* и её аналоги, а также система *ПОЭТ, диалоговая система для планирования и управления на транспорте* и др. Однако проблема понимания машиной даже ограниченного ЕЯ пока ещё далека от своего практического решения, поэтому правомерен некоторый упрощённый подход к решению проблемы взаимопонимания человека и компьютера. Суть его состоит в следующем. Во-первых, взаимопонимание связывается с непосредственными целями взаимодействия — с решаемой задачей из точно определённой узкой ПО. Во-вторых, рассматривается понимание сообщений, представленных отдельным утверждением, *вопросом* или императивом. Если машина выполнила именно те действия, которые ожидал от неё *пользователь*, передавая своё сообщение, считается, что она поняла его сообщение. От пользователя же требуется соответствующим образом сформулировать своё сообщение, а также уточнить эту формулировку, если машина её не понимает. Непосредственное взаимодействие с компьютером открывает возможности к интенсивному обмену уточняющими сообщениями, необходимому для установления понимания переданного сообщения. В-третьих, достаточно, чтобы сообщения от машины к человеку поступали на естественном для человека и решаемой задачи языке, а от человека к машине — на некотором формализованном, близком к естественному, языке, ускоряющем понимание сообщений машиной и в то же время таком, чтобы необходимость его изучения пользователем не привела к прекращению взаимодействия или подмене его цели. При такой постановке проблемы взаимопонимания очевидна ещё одна роль *“тактическо.о” автоматизированного обучения* пользователя машинным *средствам решения задач*. В результате такого обучения человек должен так формулировать свои сообщения, чтобы машина могла выполнять именно те действия, которые он от неё ожидает.

Распределение функций по решению задачи между партнёрами является естественной чертой любого Д., поскольку именно распределение знаний между партнёрами делает процесс диалогового взаимодействия обязательным. Если же хотя бы один из партнёров может выполнить все действия по решению задачи самостоятельно, то Д. не обязателен. Распределение может быть как равноценным, так и неравноценным, что зависит от возможностей каждого партнёра (наличия определённых знаний, способности выполнять различные функциональные действия, лингвистических способностей и др.). Считают, что инициатива в Д. принадлежит тому партнёру, который формулирует очередную задачу (или подзадачу), необходимую для достижения его целей. Партнёр-инициатор должен располагать знаниями как о своих возможностях, так и о том, что требуется выполнить другому партнёру на очередном этапе взаимодействия. Инициатива может передаваться партнёрами в случае возникновения конфликта, при отсутствии необходимых знаний или нарушении взаимопонимания. Если, напр., *автоматизированная обучающая система* позволяет пользователю перехватывать инициативу в различных ситуациях, то считается, что система предоставляет *обучаемому* определённую "степень самостоятельности" или "поле самостоятельности" (см. *Учебный диалог*). В современных компьютерных системах обоюдная активность партнёров может быть достигнута за счёт программного управления пользователем при выполнении таких интеллектуальных действий, как уточнение формулировки задачи, конструирование и *тестирование программы* её решения. Характерным примером диалогового взаимодействия с точки зрения равноценности деятельности партнёров может служить такое *программирование на ПЭВМ* или системе коллективного пользования, при котором одни части прикладной программы пользователь строит самостоятельно, а другие — совместно с машиной, под её управлением (см., напр., *АФРОДИТА*).

Традиционное распределение функций, при котором априори человеку поручалось всё управление, а компьютеру — только исполнение инструкций, основывалось на предположении, что человек всегда располагает преимущественными по отношению к вычислительной машине знаниями, в то время как машина всегда превосходит человека в скорости выполнения преобразований. Однако в ряде случаев компьютер, а точнее — диалоговая программная система на базе компьютера, может превосходить человека по уровню знаний, напр., машинных средств решения задач. В общем случае распределение функций между человеком и машиной должно осуществляться динамически в процессе взаимодействия на основании некоторых критериев эффективности (время решения, стоимость и пр.). Это в свою очередь должно обеспечивать наилучшее использование возможностей как человека, так и современных компьютеров.

А.М.Довгалло, М.В.Лёгкий.

ДИАЛоговая ИНформационно-ЛОгическая СИстема (ДИЛОС) — *диалоговая система*, предназначенная для непрофессиональных пользователей при решении ими прикладных *задач*. В одном разделе *базы данных* системы специальными формальными средствами отображается *информация* об исследуемых *предметных областях*, в другом — содержатся описания прикладных *программ* и *данных*, используемых при решении различных практических задач. Все *процедуры* ДИЛОС сгруппированы в несколько блоков, наз. “процессорами”. Преобразование входных фраз с естественного языка (русского или английского) на внутренний системный язык осуществляет *лингвистический процессор*. Осн. нагрузку при этом несёт *словарь*, что связано с привязкой достаточно сложной *информации* к словарным лексемам. Лингвистический процессор предоставляет *пользователю* возможности определять новые слова и пополнять словарь. Работа лингвистического процессора непосредственно связана с работой семантического процессора, осуществляющего семантическую интерпретацию выражений внутреннего языка. Совместная работа двух процессоров обеспечивает адекватную интерпретацию входных предложений в контексте данной предметной области. С помощью лингвистического и семантического процессоров осуществляется связь с “исполнительными” процессорами системы: информационным (обрабатывает *запросы* на внутреннем языке), вычислительным (организует формирование программ вычислений) и логическим (реализует логический вывод и соотносит работу всей системы с целями и задачами той системы, для которой используется ДИЛОС).

О.Н. Золотопуп.

ДИАЛоговая ИНформационно-ОБУЧАЮЩая СИСТема (ДИОС) — *авторская система*, обеспечивающая структурирование учебного содержания по данному учебному предмету и создание *базы знаний* учебного назначения в памяти *персональных электронных вычислительных машин* “Правец-8х” или “Правец-16” (IBM PC / AT/XT); выбор *процедур* для представления базы знаний обучаемым; автоматизированное планирование познавательной деятельности; сбор и анализ результатов *обучения*. Состоит из пяти независимых программных модулей. *Программы* соответственно включают: для структурирования — мощные *текстовые редакторы* и *графические редакторы*, а также редактор для структурирования учебного содержания; для интерпретации учебного содержания — процедур 1 вида “*читай*”, “*решай*”, “*анализируй*” и т.д.; для планирования познавательной деятельности — специальный редактор для создания индивидуальных заданий для каждого обучаемого; для сбора и регистрации результатов обучения — специализированную *локальную сеть* учебного

назначения и подпрограммы для визуализации и регистрации результатов. Программа для обработки результатов анализирует их в двух плоскостях: учебный предмет и обучаемые.

Техническое обеспечение *рабочего места* обучаемого: ПЭВМ "Правец-8х" или "Правец-16" (и совместимые с ними); одно или два устройства для мягкого диска. Рабочее место *преподавателя* включает ПЭВМ "Правец-8х" или "Правец-16" (и совместимые с ними); два устройства для гибкого диска; *печатающее устройство*; жёсткий диск на 5, 10, 20 М.

К.Спиров.

ДИАЛОГОВАЯ РАБОТА в графическом режиме — способ организации взаимодействия пользователя с компьютером, состоящий в диалоге между ними на основе графического представления информации. Имеет вид "вывод—ввод—вывод—ввод...". В процессе изменения или расширения одного и более частично завершённых объектов на экране и модификации их визуальных характеристик интерактивный пользователь (*автор, преподаватель, обучаемый*) обычно имеет дело с видом этих объектов. При этом графическую систему можно рассматривать как воображаемую фотокамеру, что особенно естественно для трёхмерных графических представлений. В простейшем случае пользователь может просто указать, как переместить воображаемую фотокамеру для получения другого вида того же объекта или как изменить некоторые непредставимые визуально атрибуты объекта. Независимо от намерений пользователя для сообщения этих намерений должны быть использованы устройства ввода. *Программа учебного назначения* (ПУН) расшифровывает эту информацию и применяет её либо для управления *графической подсистемой* АОС с целью изменения параметров видовой операции (т.е. изменения "настройки фотокамеры"), либо для изменения *модели объекта* в структуре *данных*. После модификации структуры данных ПУН даёт команду графической подсистеме АОС вывести изменённое изображение объекта на экран. Пользователь может задать также перемещение изображения на экране. Это можно сделать лишь после обработки ПУН *запроса* на ввод и подачи сигнала графической подсистеме на выполнение соответствующего действия. Д.р. в графическом режиме составляет основу *интерактивной машинной графики*. Эта модель в принципе не представляет затруднений для реализации, однако формализация таких диалогов всё ещё является предметом исследований.

В.И.Отенко.

ДИАЛОГОВАЯ СИСТЕМА — система, в которой реализован режим *диалога* между человеком и вычислительной машиной. Отличительная особенность Д.с. состоит в том, что и человек и

вычислительная машина могут управлять процессом взаимодействия при решении *задач* определённого класса. Эта способность выражается в предъявлении *команд* или *вопросов*, обеспечивающих целенаправленное выполнение последним некоторых действий или выдачу *ответов*, имеющих отношение к предмету диалога (см. *Взаимодействие человека-машины*). Различные Д.с. отличаются уровнем подготовки *пользователей*, на которых ориентированы эти системы. В Д.с., ориентированных на опытных пользователей, диалогом управляет преимущественно человек. Среди этих Д.с. различают системы, предназначенные для оперативного описания *алгоритма* решения задачи, для ввода и редактирования алфавитно-цифровой и графической информации, для вызова и выражения *программы* решения задачи, для *отладки программы* и т.д. В Д.с., ориентированных на неподготовленного пользователя, диалог управляет вычислительная машина, что позволяет человеку обходиться без посредника-программиста при решении своих задач. Связь пользователей с компьютером в Д.с. осуществляется с помощью клавишных устройств связи (напр., телетайпов), алфавитно-цифровых и графических *дисплеев*, устройств анализа/синтеза устной речи.

А.М.Давылло.

ДИАЛОГОВАЯ СИСТЕМА ДЛЯ КОНТРОЛЯ (ДИСК) — *генерирующая автоматизированная обучающая система*, включающая инструментальные средства для автоматизации создания обучающе-контролирующих курсов и средства проведения *контроля знаний* с помощью компьютеров серии ЕС и абонентских пунктов типа ЕС7906 или ЕС-7920 (версия ДИСК/ЕС), а также *микро-ЭВМ* типа СМ-1800/1810 (версия ДИСК/МИКРО, информационная база которой формируется в среде ДИСК/ЕС). Обеспечивает: сопровождение *автора* курса при создании информационной базы по изучаемому предмету; генерацию контролирующих тестов и формирование из них вариантов контрольной работы, оформленных как обучающе-контролирующий курс; проведение контроля знаний в режиме *диалога* под управлением системы; сбор и накопление статистических *данных* о результатах проведения контрольной работы; формирование оценки знаний *обучаемых*; консультацию *обучаемых* о правильных решениях тестов, вошедших в вариант контрольной работы. В системе предусмотрено взаимодействие с *пользователями*, имеющими статус *диспетчера*, *преподавателя* (автора курса) и *обучаемого*. Осн. организуемые в системе режимы: загрузка информационной базы по изучаемому предмету и генерация *файлов*, содержащих статистические данные; генерация контролирующих тестов и вариантов контрольной работы; проведение контрольной работы; регистрация результатов занятия. Каждый режим сопровождается соответствующим *меню*.

Процедура формирования информационной базы, возлагающаяся на диспетчера системы, заключается в загрузке файлов, структурированных в соответствии с форматами входных бланков, которые выполняются преподавателем (автором курса) на естественном языке и содержат: перечень *понятий* курса; перечень возможных вопросительных конструкций; перечень идентификаторов переменных и констант формульных выражений курса; список формул, используемых в курсе. В результате генерации содержимого обучающе-контролирующего курса, выполняемой под управлением преподавателя, формируются тесты с выборочным ответом для проверки знания осн. понятий курса и их атрибутов, а также тесты с вычисляемым результатом для проверки знания аналитических выражений, имеющих в изучаемом предмете. Из наборов тестов система компонуется варианты контрольной работы для каждого из зарегистрированных обучаемых. В режиме проведения контрольной работы на *дисплей* обучаемого последовательно выводятся тесты, входящие в его вариант задания. После *ответа* на все тесты варианта контрольной работы система предоставляет обучаемому возможность откорректировать свои ответы по всем или по части тестов варианта. Затем система переходит к анализу и оценке правильности ответа обучаемых. Критерием завершения контрольной работы является либо выполнения всеми обучаемыми полученных заданий, либо истечение времени, отведенного преподавателем на контрольную работу. Результаты анализа ответов выводятся на *печатающее устройство* компьютера и выдаются преподавателю. Форма вывода результатов табличная. Система ДИСК включает в себя также все возможности *пакета прикладных программ* ГЕНЕРАТОР PRG.

В.Н.Сороко.

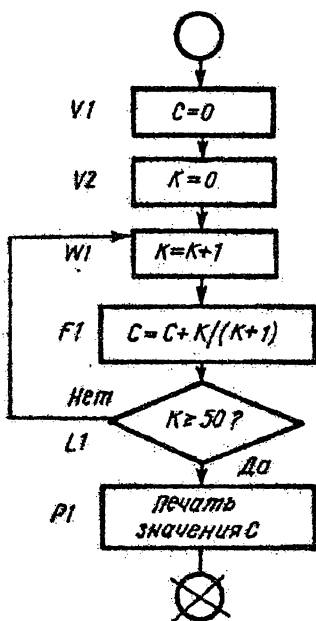
ДИАЛОГОВАЯ СИСТЕМА ДЛЯ ПЛАНИРОВАНИЯ И УПРАВЛЕНИЯ НА ТРАНСПОРТЕ (ДИСПУТ) — *диалоговая система* для получения информации из базы данных и решения задач оперативного планирования при управлении транспортным узлом. Естественноречевое общение обеспечивается в системе *лингвистическим процессором*, переводящим естественноречевые конструкции во внутреннее формальное представление, инвариантное смыслу *запроса*. В лингвистическом процессоре используется синтактико-семантическое *кодирование* слов исходного предложения с его последующим грамматическим анализом. Осн. части процессора: *словарь* словоформ, кодировщик, редактор, грамматический анализатор. Изменения и добавления в словарь вносятся с помощью специального модуля. Спецификой организации словаря в системе является использование структур *данных* типа *деревьев*.

Для работы с *текстами* на естественном языке система снабжена общими и специальными знаниями. Общие знания хранятся в

семантической сети в виде *фреймов*. Они используются для выявления темы *диалога*. Специальные знания служат для анализа предложений в рамках выбранной тематики диалога. Эти знания включают декларативную и процедурную составляющие (см. *Декларативное представление знаний, Процедурное представление знаний*). Декларативная часть знаний хранится в словарях, а процедурная — в грамматическом анализаторе. По мере нахождения в словаре слов, составляющих *запрос*, производится обращение к семантической сети и заполнение слотов *фреймов*. Грамматический анализатор переводит сообщение во внутреннее представление. Внутреннее представление воспринимается прагматическим процессором, предназначенным для перевода результатов грамматического анализа исходного запроса в задание планировщику *базы данных*. Прагматический процессор выполняет функции *интерфейса* “лингвистический процессор — база данных”, т.е. обеспечивает независимость лингвистического процессора от базы данных. При смене структуры и содержимого базы данных прагматический процессор должен быть переписан.

О.Н. Золотопул.

ДИАЛОГОВОЕ ПРОГРАММИРОВАНИЕ — вид взаимодействия с компьютером, предоставляющий возможность *обучаемому* приступить к *программированию* алгоритма решения задачи без предварительного изучения входного языка и под управлением компьютера сформировать *программу*. Взаимодействие в режиме Д.п. реализует методику алгоритмического подхода и на начальных этапах *обучения* играет роль “стартового ускорителя”, обеспечивая немедленное получение результатов счёта по программе в условиях миним. знакомства с *языком программирования* (ЯП) и отсутствия навыков работы с операционной системой компьютера. Необходимым условием для работы *обучаемого* в режиме Д.п. является владение методами алгоритмизации задач, усвоение которых можно осуществить как традиционными способами, так и с помощью автоматизированного компьютерного курса (см. *Обучение алгоритмизации задач*). *Алгоритм* решения задачи отображается для компьютера операторным описанием (рис.), которое вводится *обучаемым* в качестве исходной информации для режима Д.п. Функции компьютера, обеспечивающие Д.п. заключаются в следующем: на основе анализа операторного описания алгоритма определяются необходимое подмножество ЯП и конструкции *операторов* для формирования программы; для каждой операции операторного описания у *пользователя* запрашиваются *данные* по его задаче и после их анализа генерируются и предъявляются ему соответствующие операторы программы; компоновка программы завершается автоматически добавлением операторов языка управления заданиями и передачей задания на выполнение.



Задача: 50
 Вычислить $C = \sum_{k=1}^{50} \frac{1}{k+1}$

Операции:

V- установить значение;

W- изменить значение;

F- вычислить по выражению;

L- проверить отношение;

P- напечатать значение;

S- останов.

Операторное описание:

V1 V2 W1 F1 L1 (P1, W1) P1 S1

Обучающий эффект Д.п. заключается в том, что в процессе взаимодействия пользователю, в случае необходимости, предоставляется учебная информация о средствах ЯП, причём в объёме, необходимом для программирования его конкретной задачи. Режимы Д.п. обычно реализуются в составе многофункциональных автоматизированных учебных курсов типа АФРОДИТА, АИДА и др.

В.Д.Рынгац.

ДИДАКТ (ДИДактически-ориентированный Алг ритмический язык для написания Курсов и Тестов) — авторский язык, предназначенный для написания обучающе-контролирующих программ на компьютерах серии ЕС. Разработан в 1980 на факультете вычислительной математики и кибернетики МГУ имени М. В. Ломоносова в рамках автоматизированной обучающей системы ЭКСТЕРН. Учебный и справочный материал определяется с помощью операторов языка. Наряду с операторами можно использовать комментарии. Допустимы логические, арифметические и строковые константы. Для хранения констант, а также для манипулирования ими используются переменные обучаемого, имеющие стандартные обозначения: V1 — числовые, Sk — строковые и Wn — логические,

где l , k , n — номера переменных, определяемые автором при создании обучающей программы.

Для определения информационных, вопросных или подкрепляющих видов информации используются операторы, которые определяют проблемную ориентированность языка: **ОБОЗНАЧЕНИЕ** — для переименования переменных обучаемого; **ФРАГМЕНТ** — для определения к.-л. части текста, используемой в дальнейшей работе; **ПОНЯТИЕ** — для определения информационного кадра учебного материала; **УПРАЖНЕНИЕ** — для определения вопросного кадра; **ПОДКРЕПЛЕНИЕ** — для определения подкрепляющего кадра; **ОТВЕТ** — для задания эталона правильного ответа; **НЕПРАВИЛЬНЫЙ** — для задания эталона неправильного ответа. Кроме перечисленных, реализованы операторы, предоставляющие пользователю следующие возможности: осуществлять управление предъявлением учебного материала; генерировать числовые и текстовые данные; производить над данными арифм. и строковые операции; сегментировать учебный материал на отдельные части (разделы, главы, статьи и т.п.); использовать набор встроенных функций; обращаться к модулям, написанным на других алгоритмических языках; осуществлять идентификацию данных и операторов; принимать и анализировать ответ обучаемого с помощью стандартного набора эталонов, причём в операторе анализа ответа предусмотрен переход на указанный участок программы при совпадении ответа обучаемого с заданным эталоном; осуществлять передачу управления на любую часть учебного материала. В языке ДИДАКТ предусмотрены четыре осн. типа эталонов: текст — любой набор *сим слов*; число — любое число; фраза — лингвистически корректная фраза; выражение — правильное арифм. выражение. Реализация языка ДИДАКТ выполнена как надстройка над языком ПЛ/1 в операционной системе ЕС.

В.А.Третьяк.

ДИДАКТИКА (от греч. *διδάχῃς* — обучающий) — часть педагогики, исследующая закономерности процесса образования и обучения, воспитания в процессе обучения. Изучает общие закономерности и положения, свойственные различным учебным дисциплинам. Специфика обучения конкретным предметам изучается частными дидактиками. Д., имеющая свой предмет и область исследования, оперирует определённым кругом понятий, важнейшие из которых — обучение, процесс обучения, принципы обучения, содержание образования, методы и формы организации обучения.

Е.Д.Маргулис.

ДИДАКТИЧЕСКАЯ ЗАДАЧА — то же, что и *задача обучения*.

ДИДАКТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ — раздел информатики, изучающий вопросы программирования и машинного решения дидактических задач (см. *Задача обучения*) в автоматизированных обучающих системах. Одна из осн. задач Д.п. заключается в синтезе целенаправленной системы оптимального управления учебными действиями, при выполнении которых состояние знаний и умений обучаемого приближается к требуемым. Цель системы — информационное обслуживание и управление. Цель обучаемого — получение максимума знаний за отведённое время или достижение требуемого уровня знаний за миним. время (см. *Концептуальная модель процесса обучения*). Ключом к решению этой задачи служит метод представления знаний. Структура целей обучения и знания обучаемого представляются специальными формальными объектами — концептуальными моделями, с помощью которых представляются множества объектов и действий, а также система отношений между ними (напр., типа род — вид, часть — целое или ассоциативных: быть объектом действия, иметь свойство и др.).

Ю.И. Побанов.

ДИЛОС — см. *Диалоговая информационно-логическая система*.

ДИОПТРИЯ (от греч. *dioptr* — видящий насквозь) — единица оптической силы линзы или системы линз. Число Д. равно обратной величине главного фокусного расстояния в метрах. Знак “+” обозначает выпуклую, “—” вогнутую линзу. Для оптической системы глаза число Д. — это величина преломляющей силы оптической системы глаза, равная обратному значению её заднего фокусного расстояния, т.е. расстояния между задним фокусом и задней главной плоскостью.

А.С.Коваленко, В.Г.Мартиросова.

ДИОС — см. *Диалоговая информационно-обучающая система*.

ДИПРОФОР — первая в СССР система диалогового программирования, использующая язык Фортран. Разработана в Киевском политехническом институте совместно с Институтом кибернетики АН Украины в 1973 и реализована на компьютере “Днепр-21” с операционной системой ДД-3. Предназначена для обучаемых, впервые приступающих к программированию на языке Фортран и изучающих язык в ходе решения на компьютере тренировочных задач. Для работы с системой обучаемому необходимо составить алгоритм решения задачи и сделать его операторное описание, на основе которого ведётся диалог в системе и генерируются операторы

Фортран-программы. Система ДИПРОФОР характеризуется программным (рис.), учебным и организационным обеспечениями.

Программное обеспечение содержит следующие осн. блоки: анализа операторного описания и формирования отображения алгоритма, продуцирования диалога по задаче, обучающие модули, модули *синтаксического анализа* данных задачи, модули генерации операторов программы и модули формирования программы. Блок анализа операторного описания и формирования отображения алгоритма принимает от обучаемого операторное описание алгоритма программируемой задачи, проводит его синтаксический и семантический анализ, порождая отображение алгоритма в виде информационного поля описателей. Каждый описатель содержит необходимую *информацию* для формирования операторов Фортрана: собственную метку генерируемого оператора, если в алгоритме предусмотрен на него переход, метку оператора FORMAT, связанную с генерацией операторов ввода—вывода, и т.д. Блок продуцирования диалога сканирует очередной описатель, распознаёт связанную с ним алгоритмическую операцию и запрашивает у обучаемого *данные* для этой операции в виде конструкций Фортрана. При возникновении у обучаемого трудностей по вводу конструкций Фортрана в работу подключаются соответствующие обучающие модули. Запрошенные данные обрабатываются программами синтаксического контроля и при отсутствии ошибок управление передаётся модулю генерации операторов программы. Модуль генерации операторов, используя информацию описателя из поля отображения алгоритма и информацию, поступившую от модуля синтаксического анализа, строит (генерирует) требуемые операторы программы. Оработка этих блоков программного обеспечения системы ДИПРОФОР может привести, напр., к такому диалогу (К — компьютер, О — обучаемый):

К: Програмуем операцию проверки по отношению — L1.
Введите выражение отношения, которое необходимо проверить.

О: I>10

К: Ошибка. Для обозначения операций отношения в Фортране используется следующая символика: больше .GT.; больше или равно .GE.; меньше .LT.; меньше или равно .LE.; равно .EQ.; не равно .NE. Исправьте ошибку. Введите выражение отношения, которое необходимо проверить.

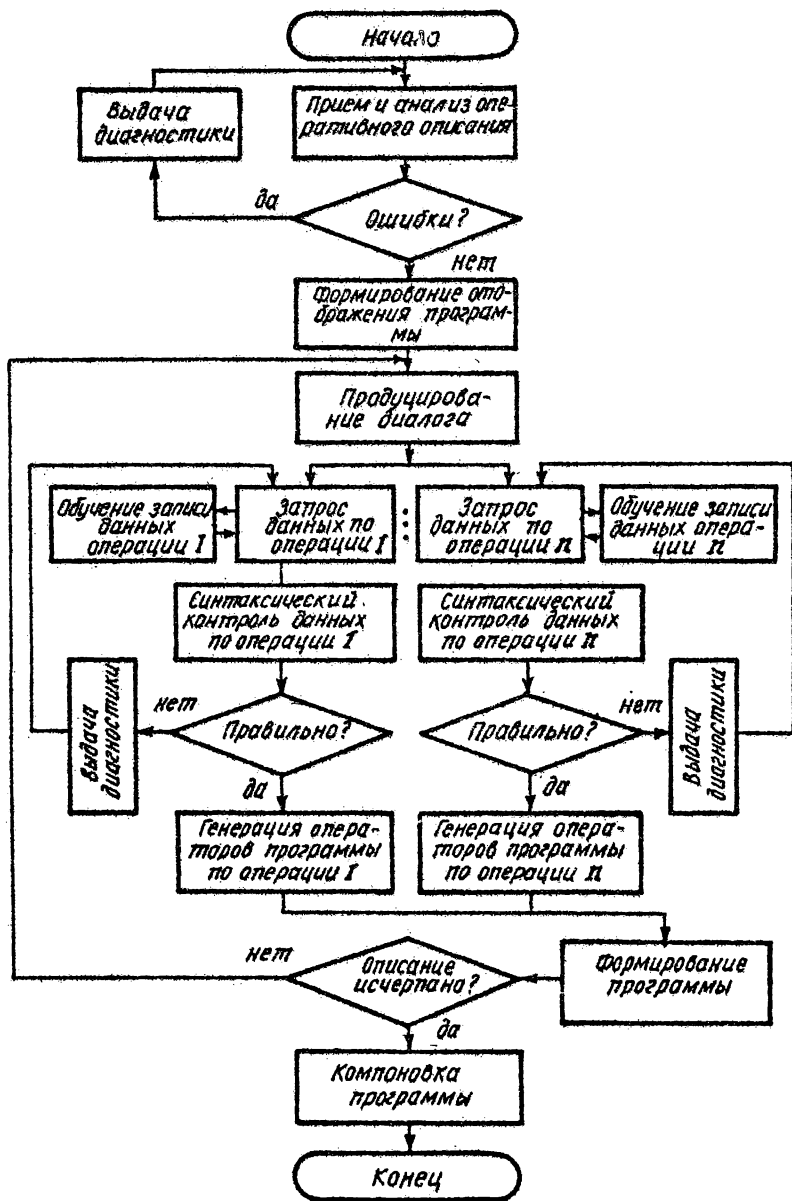
О: I .GT. 10

К: Правильно.

Оператор программы будет таким:

IF (I .GT. 10) GO TO 20

Далее блок формирования программы последовательно пересылает сгенерированные операторы в поле формирования программы и проверяет, все ли описатели просмотрены. Если не все, управление передаётся блоку продуцирования диалога. Процесс продолжается до завершения программирования задачи, после чего производится



компоновка программы добавлением операторов языка управления заданиями, и задание можно передать на обработку в пакетный режим для выполнения.

Учебное обеспечение системы состоит из трёх печатных пособий: руководства для *пользователей*, справочника по языку Фортран, комплекта учебных задач. Организационное обеспечение включает печатные материалы, предназначенные для персонала, обслуживающего компьютер в период функционирования системы. Опыт эксплуатации системы позволил выявить её недостатки и преимущества по сравнению с традиционными *автоматизированными учебными курсами* для обучения программированию; впоследствии модернизированные режимы типа ДИПРОФОР стали проектироваться в составе *многофункциональных автоматизированных учебных курсов* для обучения программированию.

В. Д. Рынгац.

ДИСК — см. *Диалоговая система для контроля.*

ДИСКЕТА — магнитный *носитель информации* в виде гибкого *магнитного диска*, заключённого в квадратный конверт или кассету. От повреждений диск защищён прокладкой из нетканого волокнистого материала. Конверты изготовляют из поливинилхлоридной плёнки, кассеты — из пластмассы. В конверте и кассете есть радиальная прорезь для контакта носителя с магнитной головкой записи (считывания) *данных*. Скорость вращения Д. — 300-360 об/мин, ёмкость — 180, 360, 720 К и 1, 2 М. Д. различаются размерами, условиями применения, плотностью записи. Характеризуются простотой смены в накопителе, компактностью и невысокой стоимостью, удобны в обращении и хранении. Ими оснащают мини- и микро-СЗМ (в частности персональные).

В. Л. Леонтьев, А. Рафф.

ДИСПЕТЧЕР — системный программист и администратор системы, задача которого — обеспечить наиболее эффективное функционирование *автоматизированной обучающей системы (АОС)* и устройств компьютера. К выполняемым Д. *процедурам* относятся: процедуры, связанные с работой *обучаемых*; процедуры, относящиеся к работе *авторов* и обслуживанию курсов; процедуры управления работой системы. Первая группа процедур предназначена для регистрации *обучаемых* в системе и удаления их после изучения *автоматизированного учебного курса (АУК)*, просмотра и изменения *данных обучаемых*. Вторая группа процедур включает регистрацию, копирование, перекомпоновку и удаление курсов, получение распечатки курсов на *печатающем устройстве* и др. Третья группа

процедур связана с ежедневной инициализацией АОС, планированием работы *пользователей*, получением статистических данных о работе обучаемых с АУК.

О.П.Платонова.

ДИСПЛЕЙ (от англ. display — показывать, воспроизводить) — устройство, принимающее *информацию* от ЭВМ, преобразующее и воспроизводящее её в виде *текста*, таблиц, чертежей и др. на экране. Различают Д. прямого видения и проекционные; на электронно-лучевых трубках, жидких кристаллах, светодиодах, плазменных (газоразрядных) и электролюминесцентных панелях. Наиболее распространены растровые Д. на ЭЛТ. Они обладают достаточно высокими характеристиками, технология их производства хорошо освоена. В таких Д. изображение на экране формируется изменением светового потока при последовательном прохождении электронного луча по элементам рабочего поля экрана (развёртка электронного луча идёт по растру). Растровый Д. состоит из управляющей части (видеоконтроллера) и видеомонитора. Управляющая часть включает *запоминающее устройство* для хранения информации, отображаемой на экране (видеопамять), блоки, формирующие сигналы управления яркостной модуляцией электронного луча, горизонтальной и вертикальной синхронизацией растра, а также блоки, обеспечивающие приём информации из ЭВМ. Видеопамять должна обладать ёмкостью, достаточной для размещения всей отображаемой на экране информации, и быстродействием, позволяющим выбирать из него данные синхронно с перемещением электронного луча по экрану.

В зависимости от представляемой на экране информации Д. подразделяют на алфавитно-цифровые (символьные), квазиграфические (алфавитно-цифровые с ограниченными графическими возможностями) и графические. В Д. первых двух типов в видеопамети хранятся коды отображаемых *символов* или элементов графики в последовательности, определяемой текстом. В Д. со стандартным форматом экрана (25 строк по 80 символов) объём видеопамети составляет 2 К. При отображении информации коды символов преобразуются в изображения знакогенераторами. В видеопамети графических Д. хранятся все элементы (точки, пиксели) изображения, которое может иметь произвольный вид (единственное ограничение — разрешающая способность); в этом случае объём видеопамети резко увеличивается.

Д. могут быть цветными и монохромными. Цветные Д. значительно сложнее и предъявляют более высокие требования к объёмам и быстродействию видеопамети.

В персональной ЭВМ видеоконтроллер входит в её состав. Чтобы получить изображение, достаточно к соответствующему выходу подключить видеомонитор, а в простейших бытовых ЭВМ —

телевизор. Для получения качественных изображений с высокой разрешающей способностью, большим количеством цветовых полутонов и градаций яркости используют видеомониторы с высокой разрешающей способностью, подключаемые к ЭВМ с помощью специальных адаптеров, имеющих большой объем видеопамяти (напр., 1 М при разрешающей способности 1024x768 точек).

Д. на жидких кристаллах отличаются малым энергопотреблением при достаточно высоких функциональных параметрах. Современные портативные ПЭВМ снабжены подобными монохромными Д. с разрешающей способностью 100x200 точек в графическом и 25 строк по 80 символов в символьном режимах. Принцип действия Д. на жидких кристаллах основывается на динамическом рассеивании и твист-эффекте. В первом случае молекулы жидкого кристалла, расположенного тонким слоем между двумя прозрачными электродами-подложками, пропускают свет, если к электродам не приложено напряжение, и задерживают его, если напряжение приложено. В кристаллах с твист-эффектом при отсутствии электрического поля молекулы жидкого кристалла между электродами образуют винтовую структуру. В результате луч поляризованного света проходит сквозь кристалл с поворотом плоскости поляризации на 90 градусов. Если к электродам приложить напряжение, то молекулы вытянутся по направлению поля, и поляризованный луч проходит кристалл без вращения плоскости поляризации. Для получения цветных изображений в жидкий кристалл добавляют специальные присадки. Тонкие и вытянутые в длину молекулы присадки располагаются параллельно молекулам в кристалле. В невозбужденном состоянии кристалл имеет определённый цвет, при перестройке молекул цвет исчезает.

Д. на светодиодах большого распространения пока не получили из-за трудности получения необходимой разрешающей способности и цветности изображения. Макс. световой поток, излучаемый светодиодом, определяется в основном количеством светоизлучающего материала.

В плазменных (газоразрядных) Д. используют свойства слабого тлеющего разряда в газовой среде (обычно в неоне), в электролюминесцентных Д. — люминесцентное свечение слоя люминофора при подаче напряжения.

Д., функционально связанный с клавиатурой, служит видеотерминалом.

В.Л.Леонтьев, В.Н.Семихов, В.И.Сигалов.

ДИСПУТ — см. *Диалоговая система для планирования и управления на транспорте.*

ДИСТАНЦИОННОЕ УЧЁНИЕ — технология *учения* на расстоянии. *Обучаемый* получает тематический материал, разработанный преподавателем и представленный пассивным либо активным способом. При пассивном способе учебный материал передаётся от преподавателя к обучаемому и нет обратной связи, показывающей уровень знаний обучаемого. Передача материала осуществляется с помощью книг, фильмов, видео, звукового магнитофона, слайдов, телетайпа, телевидения, радио и т.п. При активном способе после передачи учебного материала от преподавателя к обучаемому существует возможность проверки результатов обучения. Активный способ передачи учебного материала осуществляется через компьютеры, сеть (терминальную), видеотекст (Vtx) и интерактивное видео.

Система Д.у. включает: 1) учебный материал, предоставленный обучаемым в виде книги, с объяснениями, передающийся по радио и телевидению, который только помогает обучаемому, но не обучает сам; 2) *автоматизированные учебные курсы* (АУК), которые готовятся с помощью *авторских систем* так, чтобы: а) дать возможность выполнить задание легко и быстро даже без привлечения знаний, хранимых в компьютере; б) изменять и пополнять учебный материал, подготовленный другим автором, если авторская система используется для обработки “сырого” материала. АУК используются для обучения по различным дисциплинам. После прохождения курса обучаемые сдают экзамены. Обучаемый, сдавший экзамены достаточно успешно, получает удостоверение о среднем или высшем образовании. Это зависит от выбора курсов. В то же время он может “собрать” вместе сданные экзамены в виде так наз. “кредитов” и затем приобрести квалификацию высшей степени. Второй диплом может быть получен теми, кто уже имеет диплом об окончании института. Эта система образования допускает использование — в виде “кредитов” — успешно сданных экзаменов в других университетах или колледжах; результаты этих экзаменов добавляются к вновь сданным, что даёт возможность обучаемому получить новый диплом за год. Уровень требований в системе Д.у. достаточно высок. Обучаемый может сдать экзамены в том случае, если он знает не менее 60-80 % учебного материала.

Система Д.у. для своей поддержки имеет обычно несколько консультационных пунктов, в которых оказание консультаций является главной работой. В распоряжении студентов находятся и случайные (небольшие) консультационные пункты. Эти центры могут контактировать посредством Vtx-терминалов, чтобы использовать компьютеры этих центров для осуществления Д.у. Наиболее употребительны для таких целей Commodore - 64,16,128,+4, Sinclair, компьютеры, совместимые с PC.

Д.Ковач.

ДОКУМЕНТАЦИЯ АВТОМАТИЗИРОВАННЫХ УЧЕБНЫХ КУРСОВ — материалы по описанию целевой установки, концепции, содержания, методической структуры, результатов испытаний и использования *автоматизированных учебных курсов (АУК)*. Является необходимой составной частью поставки АУК и создается параллельно процессу их разработки. Д.а.у.к. можно оформить в виде печатного материала и/или накопить на *машинных носителях информации*, а также (если она ориентирована на *обучаемых*) непосредственно встроить в компьютерную программу.

По виду и назначению выделяют следующие формы Д.а.у.к.: краткая документация, содержащая *информацию* для разработчиков, поставщиков и *пользователей* АУК, предоставляемую в качестве обзорной, напр., для информационно-поисковых или рекламных целей; документация для разработчиков, содержащая информацию, необходимую для разработчиков, а также для лиц, ответственных за сопровождение, обслуживание и актуализацию АУК, и обычно отражающая детализированное описание его технической реализации; документация для *пользователей*, содержащая информацию для *преподавателей* и *обучаемых*. Соответственно этому отдельно выделяют документацию для преподавателей и документацию для обучаемых (которая по возможности должна быть встроена в *обучающую программу*).

В жизненном цикле АУК (рис.) важную роль играют дидактические аспекты и, следовательно, недопустимо осуществлять механический перенос стандартов документации *программно обеспечения* на Д.а.у.к. В связи с этим разработаны рекомендации по составлению Д.а.у.к., которая должна отражать: название курса,

Принятие решения о разработке учебного курса	<i>разработка учебного курса</i>			Испытание и оптимизация учебного курса	Применение курса сопровождение и обслуживание
	Концептуальная фаза	Фаза создания	Вычислительно-технический перевод		
	<i>Проверка и экспертиза</i>				
	<i>Разработка документации</i>				

учебную дисциплину, уровень обучения; учреждение-разработчик и авторский коллектив с указанием должностей; характеристику предполагаемых базовых средств (техническое и системное программное обеспечение), а также необходимого инструментального и встроенного прикладного программного обеспечения, напр., для обработки *текстов, системы управления базой данных* и т.д.; предложения по использованию курса в процессе обучения на основе учебных планов и программ, а также учебно-концептуальных позиций; определение базовых *знаний и понятий*, необходимых для начала работы с курсом; перечень релевантных симптомов для ожидаемых пробелов в знаниях обучаемых и соответствующие педагогические мероприятия по устранению этих пробелов; планируемые условия использования курса и их возможные варианты; список дидактических целей по иерархическому разделению, отчасти в операциональной формулировке; обзор содержания передаваемого учебного материала и поставленных контрольных заданий; структуру обучающей программы, представленной в наглядной форме, с последовательным её уточнением; перечень стандартно допустимых реакций на действия обучаемых; описание возможности использования других дополнительных функций компьютера, напр., возможности вычисления, получения справочной информации и т.д.; специальные указания преподавателям, напр., варианты применения и возможности приспособления к собственной концепции преподавания, возможности оперативного управления, стратегии оценки, варианты обработки протокола занятий; специальные указания обучаемым, напр. по работе с *клавиатурой*, информация о допустимых реакциях или действиях и их кодированном вводе, дополнительные указания по редактированию вводимой строки ответа и т.д.; руководство по обслуживанию курса, напр., загрузка и запуск, прерывание и перезапуск, возможные аварийные ситуации и соответствующие мероприятия; *п. процедуру* проведения испытаний для оценки действенности обучающей программы; вычислительно-техническую документацию, в которой указана миним. конфигурация технических средств и возможные расширения, распечатка исходного текста или же содержимого *файлов*, места подключения и интерфейсы с системным и прикладным программным обеспечением, отправные точки возможных изменений и т.д.; протокол отладки (план отладки с соответствующими результатами).

Для краткой документации или для документации пользователя надо произвести соответствующий выбор из указанных выше материалов и данных. Параллельно с растущей централизацией тиражирования и распространения АУК ныне возникает необходимость создания государственных стандартов для отдельных форм документации.

В.-Х. Хартвиг.

ДООПРЕДЕЛЕНИЕ ЗАДАЧИ — явление, состоящее в том, что решаемая субъектом внутренняя для него *задача*, возникшая в результате принятия им некоторой внешней для него задачи, по своему содержанию не тождественна ей. Содержание доопределённой задачи зависит от установок, мотивов и целей субъекта, имеющихся у него *знаний*, способов действий, которыми он владеет, и т.д. Нередко влияние этих факторов настолько велико, что говорят о “подмене” (переопределении) задачи. Не всегда значительное отличие внутренней задачи от внешней, на основе которой она сформировалась, должно оцениваться отрицательно. Напр., переход субъекта от предложенной извне конкретной задачи к обобщённой — важное проявление творческой активности.

Г.А.Балл

ДОСТУПА МЕТОД — алгоритм хранения и поиска записей в файле данных. Различают Д.м. логический и физический. Логический Д.м. — языковые и соответствующие им программные средства управления *данными* операционной системы, обеспечивающие ввод—вывод пользовательской программой данных, расположенных на различных внешних устройствах. Предполагает макс. логические возможности для описания процесса ввода—вывода и миним. зависимость описания от особенностей конкретного типа устройства. Физический Д.м. — программные средства в операционной системе, обеспечивающие ввод—вывод с внешних устройств компьютера. Позволяет практически полностью использовать аппаратные средства компьютера.

А.П.Ильяшенко, М.Е.Козлов.

ЕСТЕСТВЕННЫЙ ЯЗЫК ОГРАНИЧЕННЫЙ — профессионально-ориентированный язык, представляющий собой подмножество естественного языка, но обладающий рядом особенностей, связанных со сферой его применения (*информационные системы*). Наиболее важные из этих особенностей: используемая лексика ограничена *предметной областью*, к которой они относятся; за словами этих языков стоят точные определения их смысла для системы, т.е. в памяти системы каждую значащую единицу лексики можно полностью проанализировать, используя специальную *информацию*, хранящуюся в словарной статье для этого слова; допускаются только такие *тексты*, смысл которых можно получить на основе истолкования смысла входящих в него отдельных слов с учетом синтаксических и морфологических правил применительно к этому тексту. Е.я.о. лишен таких средств, как метафоры, ассоциации, недоговоренности. Однако по морфологии и синтаксису он мало чем отличается от обычного естественного языка. Академик А.П.Ершов ввел специальный термин для Е.я.о. — “деловая проза”. Под Е.я.о.

(применительно к языку общения конечного пользователя с компьютером) понимается такой язык, используя который в рамках обратных приложений, пользователь может предпринимать обращения к инструкциям и не запоминать различные правила построения своих высказываний. См. также *Искусственный язык*.

Н.А.Власенко.

ЕХСНЕСК — *автоматизированная обучающая система*, предназначенная для обучения математическим дисциплинам (логике, теории множеств и теории доказательств). Разработана в Стэнфордском университете (США) во 2-й половине 70-х гг. группой под руководством П.Суппеса. Помимо традиционного “сценарного” представления учебного материала система содержит базу знаний и процедуры вывода, позволяющие анализировать доказательства теорем, выполненные обучаемым, строить объяснения и аннотированный вывод утверждений на основе имеющихся лемм и теорем. Для этой цели в ЕХСНЕСК используются так наз. процедуры естественного вывода — HYPOTHESIS, INDIRECT PROOF, IMPLIES и ESTABLISH, соответствующие определенным этапам построения матем. доказательства.

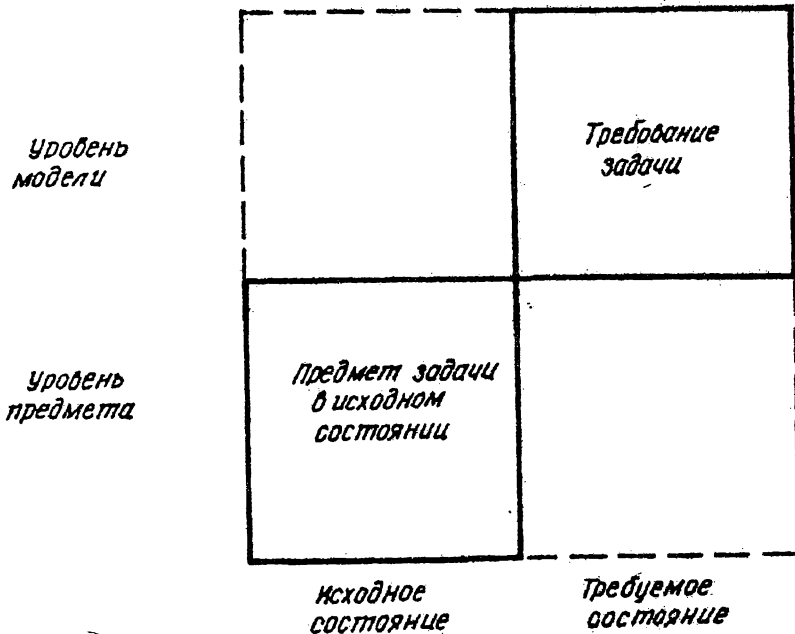
HYPOTHESIS формирует утверждения, принимаемые в качестве исходных предположений (посылку теоремы), а также цель доказательства (заключение). Обращение к процедуре INDIRECT PROOF означает, что доказательство будет выполняться методом “от противного”, т.е. в число предположений включается отрицание цели, а новой целью доказательства выбирается противоречивое утверждение. Процедура IMPLIES применяет определение или некоторый полученный в процессе доказательства результат в качестве правила вывода к заданному контексту. Если применяемое правило не связано с текущим доказательством, а представляет собой некоторый результат (теорему) теории множеств, для получения следствий используется процедура ESTABLISH. Обучаемый указывает в качестве посылок утверждения, полученные в процессе вывода, и формулирует следствие, а поиск теоремы, с помощью которой можно получить такой вывод, осуществляется автоматически.

Построение доказательства выполняется в диалоге с системой; на каждом шаге доказательства должна быть явно указана процедура вывода, с помощью которой можно получить текущий результат. По доказательству, предложенному обучаемым, ЕХСНЕСК может построить его сокращенный вариант, выделяя наиболее существенные, с матем. точки зрения, этапы, без указания способа получения промежуточных результатов. Помимо этого, система может предложить резюме доказательства с указанием используемых положений теории (аксиом, лемм, теорем) на естественном языке. Ввод

доказательства также осуществляется на языке, близком к естественному.

Е.М.Синица.

ЗАДАЧА — совокупность некоторой цели и условий, в которых она должна быть достигнута. В теории задач трактуется как система (рис.), обязательными компонентами которой являются предмет Z , находящийся в исходном состоянии, и ее требование (модель требуемого состояния указанного предмета). Предмет Z может быть



материальным (напр., груз, находящийся на станции отправления) или идеальным (напр., неизвестное числовое значение некоторой функции, поверхностные знания школьника по к.-л. теме). Применительно к указанным примерам требование Z может предусматривать соответственно доставку груза на станцию назначения, вычисление значения функции, углубление знаний школьника.

Решение Z заключается в переводе ее предмета из исходного состояния в требуемое. Если же эти состояния изначально совпадают, Z вообще отсутствует. Последний тезис не следует понимать упрощенно. Напр., Z управления может предусматривать стабилизацию некоторого параметра управляемого объекта. Требование такой

З. состоит, по сути, из двух, указывающих соответственно на саму необходимость стабилизации и на конкретное значение стабилизируемого параметра. Если хотя бы одно из них не выполнено, З. не решена.

Кроме обязательных, З. может содержать дополнительные компоненты, напр. указания о допустимых или рекомендуемых *средствах решения* или *способах решения*, о временных ограничениях, накладываемых на процесс решения, и т.п.

При проектировании человеко-машинных систем, напр. систем "обучаемый - компьютер", в качестве компонентов З. выделяют: предметную область (K) или ее модель (K^*); модель требуемого состояния ПО ($K_{тр}^*$); актуальное, т.е. существующее в момент возникновения З., состояние её ПО ($K_{акт}$). Следовательно, задача $Z = (K, K_{акт}, K_{тр}^*)$ или $Z = (K^*, K_{акт}, K_{тр}^*)$. Такое представление структуры З. является частным вариантом данного выше общего представления. Явное введение ПО или ее модели продуктивно в связи с созданием искусственных *решателей*, для которых постановка З. обычно начинается с формирования *модели предметной области*. ПО задачи (K) в свою очередь является тройкой $(\{A\}, \{\omega\}, \{R\})$, где $\{A\}$ — множество предметов; $\{\omega\}$ — множество операций на $\{A\}$; $\{R\}$ — множество отношений (предикатов) на $\{A\}$. Напр., в известной игре "15" множеством предметов $\{A\}$ является множество фишек, операциями ω на $\{A\}$ — сдвиги фишек на свободное место, множеством $\{R\}$ — множество расположенных фишек. Актуальное состояние ПО $K_{акт}$ — это расположение фишек, существующее в момент постановки З. Требуемое состояние ПО $K_{тр}$ — расположение фишек в порядке возрастания номеров слева направо и сверху вниз. Моделью $K_{тр}^*$ может быть фраза "Расположить фишки в порядке возрастания номеров слева направо".

З. как систему следует отличать от ее знаковых моделей, в т.ч. словесных формулировок. Напр., тексты "Требуется решить уравнение $x^2 - 3x + 4 = 0$ " и "Найдите корни уравнения $y^2 - 3y + 4 = 0$ " можно считать формулировками одной и той же З. (если считать, что решатель обладает необходимыми знаниями по элементарной алгебре). З. могут находиться в различных отношениях друг к другу. В частности, З. N может быть по отношению к З. M *задачей контроля*; задачей разрешения (если требование З. N состоит в установлении того, можно ли выполнить требование З. M); задачей нахождения способа решения задачи M ; подзадачей З. M (если способ решения З. N входит в способ решения З. M ; напр. подзадачей З. вычисления объема усеченной пирамиды является З. вычисления объема полной пирамиды, из которой получилась эта усеченная пирамида).

З. можно рассматривать либо по отношению к определенному решателю (или решателю определенного типа, напр. специалисту

определенного профиля), либо в абстракции от него. Иногда такая абстракция невозможна, в частности если предмет Z совпадает с решателем (напр., в Z . самовоспитания), является его подсистемой (напр., когда коллектив воспитывает одного из своих членов), или, напротив, решатель входит в состав предмета Z . (напр., когда спортсмен решает Z . победы своей команды над соперничающей с ней; предмет Z . охватывает в этом случае обе команды).

Типы Z . выделяются либо без учета характеристик решателя, либо с их учетом. В первом случае различают индивидуальные и массовые Z ., материально и идеально направленные (информационные) Z ., принципиально разрешимые и неразрешимые Z . Индивидуальной наз. такая Z ., предмет которой является единственным (конкретным), массовой — Z ., предмет которой является любым из некоторого класса предметов. В материально направленной Z . ее предмет материален и не выступает в функции модели. Z . решается, чтобы обеспечить сугубо материальное свойство этого предмета (нахождение в указанном месте, определенную конструкцию, химический состав и т.п.). В идеально направленной Z . предметом ее служит модель к.-л. системы (ее описание, изображение, образ в сознании человека, представление в памяти ЭВМ и т.п.). Z . решается, чтобы обеспечить требуемые характеристики информации, которую данная модель несет о моделируемой системе. К идеально направленным принадлежат как относительно простые по структуре Z ., предусматривающие перенос некоторой информации с одного носителя информации на другой или изменение формы ее представления, так и более сложные и интересные — задачи усовершенствования знаний. Z . принципиально неразрешима, если в соответствии с закономерностями той области действительности, к которой она относится, ее решение невозможно, т.е. либо невозможно требуемое состояние предмета Z ., либо, хотя оно в принципе и возможно, невозможен переход к нему из актуального состояния этого предмета. Принципиально неразрешима, напр., Z . построения вечного двигателя или Z . оживления умершего человека после наступления необратимых изменений в его нервной системе. Массовая Z . может быть принципиально разрешимой при одних значениях параметров, характеризующих ее предмет, и неразрешимой при других значениях этих параметров. Говорят также о разрешимости и неразрешимости Z . для определенного решателя. По мере развития науки Z ., считавшиеся принципиально неразрешимыми, могут оказаться разрешимыми (как это имело место, напр., по вопросу о возможности превращения химических элементов).

С учетом характеристик решателя выделяют Z . внешние и внутренние для определенного решателя, рутинные, квазирутинные и нерутинные (*проблемные задачи*), четкие, квазичеткие и нечеткие задачи. Предмет и требование в Z . находятся вне решателя.

Предметом внутренней Z служит некоторая имеющаяся в решателе модель, а её требованием, соответственно, модель требуемого состояния этой модели. Всякая внутренняя относительно произвольного решателя Z является идеально направленной. Внешние и внутренние относительно произвольного решателя Z в своей совокупности не исчерпывают множества Z ., рассматриваемых по отношению к этому решателю. Так, можно вести речь о решении субъектом Z ., компонентами которой являются некоторый внешний предмет и цель субъекта по отношению к этому предмету. См. также *Доопределение задачи*. Массовая Z является для определенного решателя рутинной (соответственно квазирутинной), если он обладает алгоритмом (соответственно квазиалгоритмом) ее решения и нерутинной, если он не обладает ни алгоритмом, ни квазиалгоритмом её решения. Индивидуальная $Z.M$ является рутинной (соответственно квазирутинной) для решателя Q , если одновременно выполняются два условия: 1) $Z.M$ принадлежит к типу Z ., соответствующему рутинной (соответственно квазирутинной) массовой Z .; 2) прямая информация об этом имеется в решателе Q либо операция, обеспечивающая отнесение $Z.M$ к указанному типу, для этого решателя эффективна (соответственно квазиэффективна). Если индивидуальная Z не является для рассматриваемого решателя рутинной или квазирутинной, то она нерутинна для него. $Z.M$ является для решателя Q : четкой, если прямая информация о том, решена ли эта Z ., находится в распоряжении решателя, или если Z контроля для $Z.M$ является рутинной; квазичеткой, если прямая информация о том, решена ли $Z.M$, с вероятностью, достаточно близкой к единице, находится в распоряжении решателя, либо если Z контроля для $Z.M$ квазирутинна; нечеткой, если она не является ни четкой, ни квазичеткой.

Отнесение Z к тем или иным из описанных выше типов позволяет охарактеризовать рассматриваемые Z в качественном отношении. В дополнение к этому вводятся количественные характеристики Z . Основные из них — уровень *трудности задачи* и уровень *сложности задачи*. Проблемные (нерутинные) Z можно охарактеризовать также уровнем проблемности, показывающим, в какой мере для решения Z решатель должен выйти за пределы находящихся в его распоряжении алгоритмов и квазиалгоритмов. Для нечетких Z можно указать и уровень нечеткости, отождествляемый с уровнем проблемности Z контроля для данной Z . Эти уровни характеризуют, собственно, не Z как таковую, а реальный или прогнозируемый процесс ее решения определенным решателем.

При анализе и построении процесса *обучения* требуется, наряду с описанными выше качественными и количественными характеристиками Z ., учитывать их функции в указанном процессе. С этой целью выделяются *учебные задачи*, решаемые *обучаемыми*, *критериальные задачи*, которые они должны научиться решать, и

дидактические З. (*задачи обучения*). Пользуясь понятиями об учебных и дидактических З., выделяют три осн. варианта использования компьютера в обучении. 1) Компьютер — средство решения только учебных (но не дидактических) З. Используется в качестве справочной системы, средства осуществления расчетов, моделирования и т.п. 2) Компьютер — средство решения только дидактических (но не учебных) З. Взаимодействует с преподавателем, выдавая ему результаты статистической обработки и анализа информации о процессе обучения, рекомендации об осуществлении тех или иных операций и т.д. 3) Компьютер — средство решения и учебных, и дидактических З. Взаимодействуя с обучаемым, непосредственно управляет его учебной деятельностью посредством соответствующей последовательности обучающих воздействий.

Г.А.Балл, А.М.Довгялло, Э.Л.Ивахненко.

ЗАДАЧА КОНСТРУИРОВАНИЯ — задача синтеза (компонования) некоторых сложных объектов из сравнительно более простых. Напр., при решении З.к. предложенный язык программирования (ЯП) в исходном состоянии предполагается наличие некоторого множества неформальных инструкций или запросов и описания формального языка. Требование задачи — получение множества адекватных неформальных предложений, корректных в формальном смысле. Процесс решения З.к. состоит в выборе операторов (*команд*) ЯП и конкретизации значений их аргументов за счет информации, содержащейся в множестве неформальных запросов. Наиболее распространенная форма процесса решения З.к. — диалоговое взаимодействие.

А.М.Довгялло.

ЗАДАЧА КОНТРОЛЯ — задача, в результате решения которой устанавливается, в каком отношении находятся между собой выявленные знания обучаемого и некоторая эталонная модель требуемых знаний. В теории задач З.к. — это задача, у которой $K_{\text{дкт}}^* = (A, \text{ЭТ}, M)$, где A — контролируемая система (объект); ЭТ — эталонная система; M — отношение соответствия (напр., совпадения, вхождения, пересечения) между A и ЭТ . Требование З.к. выражается прямым вопросом “Входит ли контролируемый объект A в объект ЭТ ?” либо императивом “Доказать, что A и ЭТ удовлетворяют некоторому отношению M ”.

А.М. Довгялло.

ЗАДАЧА ОБУЧЕНИЯ, дидактическая задача — *задача*, решаемая *обучающей системой* (педагогом) и направленная на достижение *целей обучения*. Предполагает изменение самого субъекта *учебной деятельности*, т.е. расширение и уточнение имеющейся в его распоряжении системы знаний, умений, навыков, сформированных умственных действий и т.д.

А.М. Довгялло.

ЗАДАЧА ПРОГРАММИРОВАНИЯ — задача, предметная область которой представлена универсальным множеством *операций* и состоящих из них *программ*. Исходное состояние З.п. содержит некоторый план решения *задачи* и отношение, определяющее наличие программы ее решения. *Модель* требуемого состояния предполагает получение программы решения этой задачи. Процесс решения заключается в выборе программы из множества существующих либо решения *задачи конструирования* ее в соответствии с планом. См. *Программированное обучение*.

А.М. Довгялло.

ЗАДАЧА УСОВЕРШЕНСТВОВАНИЯ ЗНАНИЙ, эпистемическая задача — *задача*, предметом которой служит *знание*. Представляет собой важнейший класс идеально направленных задач. В зависимости от отношения между предметом задачи и *решателем* выделяют такие классы З.у.з., как *познавательные задачи* и *коммуникативные задачи*.

Г.А. Балл.

ЗАКРЫТЫЕ И ОТКРЫТЫЕ ЗАДАЧИ — разновидности *познавательных задач*. *Задача* наз. *закрытой*, если для решения ее достаточно для каждого из *вопросов* задачи выбрать подходящий *ответ* из находящегося в распоряжении *решателя* набора вариантов. Другие познавательные задачи наз. *открытыми*. Решение *открытых проблемных задач* занимает ведущее место в творческой деятельности. В формулировках *закрытых задач* чаще всего используют *закрытые вопросы*. Но иногда задача *закрыта*, хотя и формулируется с помощью *открытого вопроса*. Примером может быть грамматическая задача, в которой для каждого из содержащихся в *тексте* существительных требуется указать род, число и падеж. Хотя их возможные значения не перечислены явно в формулировке задачи, т.е. все вопросы задачи являются *открытыми*, саму задачу естественно считать *закрытой*, если эти значения хорошо известны решающему ее учащемуся.

Чтобы *закрытая задача* оказалась *рутинной* (или *квазирутинной*), достаточно в распоряжение *решателя* предоставить *алгоритм* (соответственно, *квазиалгоритм*) поиска ответа (ответов) на вопрос

(вопросы) задачи. Поиск понимается при этом как процесс выделения элемента с заданными свойствами из некоторого конечного множества. Если, однако, приходится вести поиск среди большого числа возможностей, то процесс поиска может оказаться слишком громоздким, так что более целесообразным становится обращение к нерутинному способу решения задачи.

Г.А.Балл.

ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ЗУ) — устройство для приема и хранения информации, поступающей из других устройств электронной вычислительной машины или извне через устройства ввода—вывода информации. В ЗУ выполняются операции записи информации, поступающей из других устройств, или считывания (чтения) хранимой информации для передачи в другие устройства. ЗУ состоит из носителя информации и механизма, с помощью которого осуществляется запись — чтение информации. В зависимости от связи с процессором ЗУ подразделяют на внутренние и внешние. Внутренние ЗУ непосредственно связаны с процессором внутренним интерфейсом и представляют собой часть ЭВМ или процессора. ЗУ, связанные с ЭВМ внешним интерфейсом, являются внешними (накопителями). По возможности изменять и сохранять информацию внутренние ЗУ могут быть постоянными (ПЗУ) (храняемая в них информация в рабочем режиме ЭВМ не изменяется) или оперативными (ОЗУ) (информация может многократно изменяться). К внутренним ЗУ относятся буферные устройства, используемые для промежуточного хранения информации, т.наз. электронные диски. Внутренние ЗУ выполняются в основном на больших интегральных схемах (БИС).

ЗУ характеризуются емкостью и временем обращения. Емкость определяется количеством информации, которое может храниться на носителе ЗУ одновременно. Время обращения к ЗУ — это интервал времени между началом и окончанием записи информации в ЗУ или началом и концом чтения информации из ЗУ. В зависимости от этих двух характеристик внутренние ЗУ подразделяют на сверхоперативные (СОЗУ) и оперативные (ОЗУ), внешние ЗУ — на основные и массовые (табл.).

Запоминающее устройство	Емкость, бит	Время обращения	Запоминающий элемент
СОЗУ	$10^2 - 10^5$	1,0-100 нс	Регистр, интегральная схема
ОЗУ	$3 \cdot 10^4 - 10^7$	30нс-1,0мкс	Интегральная схема
Внешнее ЗУ основное	$10^6 - 3 \cdot 10^9$	1,0-20мс	Ферромагнитное покрытие, оптический диск

Запоминающее устройство	Емкость, бит	Время обращения	Запоминающий элемент
Внешнее ЗУ массовое	$10^8 - 10^{12}$	0,5-20 с	—

Различают ЗУ с произвольным, последовательным и параллельно-последовательным доступом к информации. ЗУ с произвольным доступом дают возможность по заданному адресу одинаково быстро извлекать информацию из любого места носителя информации. Такими ЗУ служат различные типы СОЗУ, ОЗУ и ПЗУ. ЗУ с последовательным доступом — это преимущественно различные типы внешних ЗУ. С параллельно-последовательным доступом обычно выполняют буферные ЗУ.

Внешние ЗУ различают по виду материала, запоминающей среды или принципа изготовления носителей информации, по разновидности самих носителей и сконструированных на их основе накопителей информации. В зависимости от материала, из которого изготовлены носители, и запоминающей среды носителей внешние ЗУ подразделяют на магнитные, полупроводниковые, оптические (диски с обратимой и необратимой записью информации). К магнитным внешним ЗУ относятся ЗУ на цилиндрических магнитных доменах (в виде интегральных схем), магнитных лентах и дисках. Из ЗУ на магнитных лентах наиболее распространены кассетные на магнитных лентах, имеющие малые габариты и небольшую стоимость. Наиболее широко используют внешние ЗУ с накопителями на гибких и жестких магнитных дисках.

В.П.Козлова, В.Л.Леонтьев, А.Рафф.

ЗАПРОС — форма представления информационной потребности пользователя (ИПП). Могут быть стандартными и нестандартными. В виде стандартных З. можно выразить лишь неизменяемую ИПП, характеризующуюся постоянством условий поиска, алгоритма обработки данных и состава задачи. Напр., приложениями, в которых достаточно иметь стандартные З., являются заказ авиа — и железнодорожных билетов, бронирование мест в гостинице и т.п. Для выражения изменяющейся ИПП используются нестандартные З., для которых перечень атрибутов, участвующих в условиях поиска, обработки и выдачи, заранее не специфицирован, как при стандартных З. По виду условий поиска нестандартные З. подразделяются на фактографические, документальные и смешанные. Фактографические З. предполагают поиск по атр бугам, отображающим формальные свойства объектов предметной области. Такими атрибутами могут быть, напр., ТАБЕЛЬНЫЙ-НОМЕР, ФАМИЛИЯ, ДОЛЖНОСТЬ, НАЗВАНИЕ-ЖУРНАЛА и т.п. Общее свойство фактографических атрибутов состоит в том, что множество их

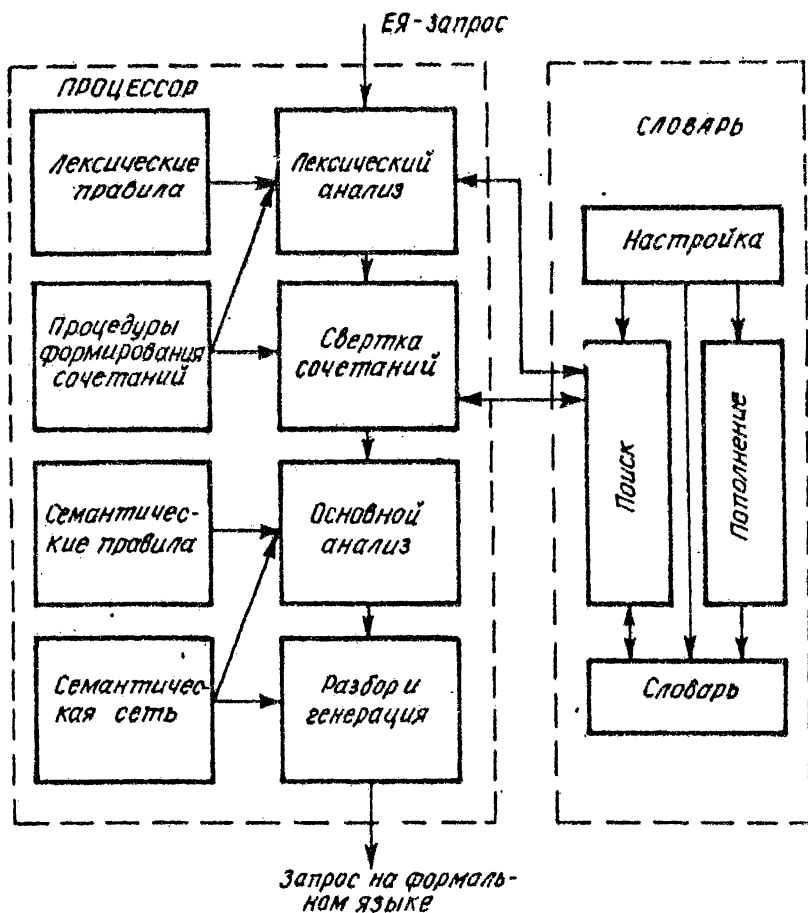
значений может быть заранее задано. Документальные З. предполагают поиск по атрибутам, отображающим неформальные, качественные характеристики объектов предметной области. Обычно эти характеристики выражаются в виде одного или нескольких связанных предложений на естественном языке. Такие атрибуты наз. содержательными, а поиск по их значениям — поиском по содержанию. Примером такого атрибута может служить атрибут СОДЕРЖАНИЕ-ПУБЛИКАЦИИ. Его значением может быть *текст* реферата или аннотации на естественном языке. В условиях поиска нестандартных смешанных запросов содержатся компоненты документальных и фактографических запросов.

Н.А.Власенко.

ЗАПСИБ (ЗАПрос к Справочно-Информационной Базе) — семейство *лингвистических процессоров* для естественного языкового взаимодействия с прикладными базами данных. Характерными особенностями младшей модели семейства — ЗАПСИБ-10 являются семантически ориентированный анализ (традиционная схема лингвистического анализа: *морфологический анализ* → *синтаксический анализ* → *семантический анализ* заменяется упрощенной схемой: *лексический анализ* → *семантический анализ*); использование восходящей (снизу—вверх) техники разбора; относительная простота алгоритмов анализа. Осн. макро модули ЗАПСИБ-10 — ПРОЦЕССОР и СЛОВАРЬ. Макромодуль ПРОЦЕССОР включает: модуль лексического анализа (здесь выполняется свертка исходного *запроса* в последовательность компонент с указанием типа компоненты); модуль свертки словокомплексов (формируются единые компоненты, соответствующие устойчивым словосочетанием); блок осн. анализа (интерпретируется описание языка общения, представленное в виде набора групп семантико-синтаксических правил продукционного типа); блок разбора и генерации (с помощью правил семантического согласования устраняются неоднозначности и генерируется запрос на формальном языке). Макромодуль СЛОВАРЬ располагает простыми средствами пополнения, корректировки словарных статей и средствами по настройке лингвистического процессора на другие *предметные области*. Экспериментальная версия ЗАПСИБ-10 реализована на языке программирования СЕТЛ, а производственная — на языке ПЛ/1 (рис.).

Н.А.Власенко, А.П.Ильяшенко.

ЗАПСИБ-ПРОЛОГ — *интегрированный пакет прикладных программ* для создания естественного языковых информационных систем путем настройки лингвистического процессора ЗАПСИБ на *предметную область* (ПО), реализованную средствами языка логического программирования *Пролог. Лингвистический процессор*



переводит сообщения (запросы) конечного пользователя с естественного языка на формальный, который отделен от базы данных (БД) и является языком интерфейса между лингвистическим процессором и БД. Особенности пакета ЗАПСИБ-ПРОЛОГ-ЕС можно проиллюстрировать на примере создания естественной языковой информационной системы МОНОГРАФИЯ.

Рассмотрим ПО, содержащую информацию о монографиях по языкам программирования. БД, соответствующая этой ПО, представляет собой таблицу, в строках которой находится информация о каждой монографии и которая на языке Пролог представляется

предикатом, имеющим структуру: +М(*АВТОР, *НАЗВ, *ГОР, *ИЗД, * ГОД), где *АВТОР — фамилия автора монографии; *НАЗВ — название монографии; *ГОР — город, где была издана монография; *ИЗД — наименование издательства; *ГОД — год издания монографии. Переменные *АВТОР, *НАЗВ, *ГОР, *ГОД в БД получают значения из множества объектов. Пример записей в БД:

+М(БЕЛЯЕВА,АЛГОЛ-60,ХАРЬКОВ,ВИЩА_ШКОЛА,1977).

+М(ГУРЬЯНОВ,АЛГОЛ-60_В_УПРАЖНЕНИЯХ,ЛЕНИНГРАД, ИЗДАТЕЛЬСТВО_ЛГУ,1977).

М(ЛЕСИК,АЛГОЛ-60,ХАРЬКОВ,ВИЩА_ШКОЛА,1977).

Конечный пользователь, зная структуру БД, может получить информацию, непосредственно обращаясь к созданной базе средствами языка Пролог. Напр., если пользователя интересует информация о монографиях Беляевой, то на языке Пролог этот запрос будет представлен так:

—М(БЕЛЯЕВА,*НАЗВ,*ГОР,*ИЗД,*ГОД)-OUTERM(*НАЗВ)-LINE., где встроенные процедуры Пролога OUTERM и LINE обеспечивают вывод на печать запрашиваемой информации. На этот запрос будет получен ответ:

АЛГОЛ-60

Чтобы облегчить обращение к этой и подобным реляционным БД (особенно при увеличении количества атрибутов базы), введем более простой и удобный язык запросов. Запрос на этом языке состоит из двух директив. Первая директива ВЫДАТЬ:... содержит список тех атрибутов БД, которые необходимо вывести на печать после обработки второй директивы языка ИЗВЕСТНО:..., где помещаются известные значения атрибутов, по которым производится поиск в БД. Предыдущий запрос о монографиях Беляевой на предлагаемом языке представляется так:

ВЫДАТЬ: НАЗВАНИЕ

ИЗВЕСТНО: АВТОР=БЕЛЯЕВА

Для обработки таких запросов необходимо реализовать средствами языка Пролог базу знаний (БЗ), включающую средства по переводу запроса приведенного типа в структуру:

—М(*АВТОР,*НАЗВ,*ГОР,*ИЗД,*ГОД).

Некоторые переменные в этой структуре получают значения, указанные во второй директиве ИЗВЕСТНО языка запросов.

Благодаря средствам языка Пролог легко расширить язык запросов, включая в ЕЗ понятия, не являющиеся атрибутами БД. Так, в нашем примере для обработки запроса:

ВЫДАТЬ: СОАВТОР

ИЗВЕСТНО: АВТОР=БЕЛЯЕВА,НАЗВАНИЕ=АЛГОЛ-60

необходимо ввести в БЗ аксиому СОАВТОР, которая описывает, каких авторов можно считать соавторами: авторов можно считать соавторами, если в строках БД у них совпадают значения атрибутов

*НАЗВ,*ГОР,*ИЗД,*ГОД. Введя в БЗ эту аксиому, в ответ на наш запрос получим:

ЛЕСИК

т.е. Лесик является соавтором Беляевой по монографии "Алгол-60". Формирование *ответа* на этот запрос является простейшим случаем логического вывода: получаем информацию, которая в явном виде в БД не присутствует.

Для реализации естественного обращения к БД необходимо иметь средства для перевода естественного запроса в некоторое формальное представление, в данном случае — в язык запросов с директивами ВЫДАТЬ и ИЗВЕСТНО. В качестве такого средства используется лингвистический процессор серии ЗАПСИБ. Система ЗАПСИБ-ПРОЛОГ-ЕС работает под управлением процессора ЗАПСИБ, а ПРОЛОГ вызывается как подпрограмма ЗАПСИБа. Пример работы программы МОНОГРАФИЯ.

РАБОТАЕТ СИСТЕМА ЗАПСИБ-ПРОЛОГ-ЕС

ЗАПРАШИВАЙТЕ:

КАКИЕ МОНОГРАФИИ ОПУБЛИКОВАЛ ГУРЬЯНОВ В 1977 ГОДУ?

АЛГОЛ-60 В УПРАЖНЕНИЯХ

ЗАПРАШИВАЙТЕ:

НАЗОВИТЕ МОНОГРАФИИ ИЗДАТЕЛЬСТВА ВИЦА ШКОЛА!

АЛГОЛ-60

ЗАПРАШИВАЙТЕ:

КОНЕЦ

СЕАНС РАБОТЫ ОКОНЧЕН

Н.А.Власенко, А.П.Ильшенко.

ЗНАК — предмет, выступающий в качестве представителя другого предмета (наз. денотатом, или предметным значением, знака). Как правило, З. не является *моделью* своего денотата, но воздействие его на активную систему, для которой он предназначен (рецепиента знака), с достаточно большой вероятностью приводит к формированию или активизации в этой системе некоторой модели указанного денотата. Напр., под воздействием слова, обозначающего к.-л. вещь, в сознании человека возникает ее образ. *Информация*, которую несет такая модель о денотате рассматриваемого З., наз. его смыслом, или смысловым значением, причем можно различать его смысл для конкретного рецепиента и нормативный смысл. См. также *Символ*.

Г.А.Балл

ЗНАНИЕ — система взаимосвязанных результатов познания некоторого объекта, содержащая достоверную (либо принимаемую за таковую) *информацию* о нём. Различают З. эмпирические,

получаемые на основе наблюдения объекта, проводимых над ним экспериментов и т.п., и теоретические, которые в соответствии с общими принципами той или иной науки раскрывают закономерности, существенные для рассматриваемого класса объектов. Всякое З., присутствующее в индивидуальном или общественном сознании, представляет собой идеальную модель. Для передачи З. от одного индивида, коллектива, поколения другому эта модель материализуется с помощью устной или письменной речи, изобразительной деятельности и т.п. В любом З. можно выделить не менее двух компонентов, каждый из которых также можно описать как идеальную модель. Простейшей формой З., как известно из логики, является суждение (напр., “все воробьи суть птицы”). *Понятия*, играющие роль субъекта и предиката суждения (в данном случае “воробей” и “птица”), выступают в качестве упомянутых компонентов-моделей.

З. о тех или иных объектах, а также о возможностях и способах их преобразования, служат *средствами решения задач*, в предметную область которых входят эти объекты. Важное значение, в особенности для творческой деятельности (напр., научной, инженерной, организаторской, педагогической), имеют так наз. методологические З., позволяющие субъекту самостоятельно вырабатывать эффективные *способы решения задач*. В процессе *обучения* формирование З. у *обучаемых* должно рассматриваться не как самоцель, а как составная часть вооружения их средствами решения тех задач, с которыми они столкнутся в последующей деятельности. См. также *База знаний, Задача усовершенствования знаний, Методы представления знаний, Представление знаний, Системы, основанные на представлениях знаний*.

Г.А.Балл.

ИГГА — вид деятельности, прямой продукт которой (см. *Продукты деятельности*) не имеет потребительской ценности и является личным приобретением ее участников. Появление И. связано с потребностью людей воссоздавать и усваивать общественный опыт, фиксированный в социально закрепленных способах осуществления предметных действий, в условных ситуациях. И. отражает нормы человеческой жизни и деятельности, поэтому анализ этого вида деятельности представляет достаточно сложную проблему. О сложности психологического анализа И. свидетельствует тот факт, что, несмотря на многочисленные исследования в этой области, понятийный аппарат еще не разработан: нечеткость статуса и соотношения таких *понятий*, как “игровая деятельность”, “игровой процесс”, “игровые методы” (в т.ч. — в *обучении*), игровые элементы, приемы и пр., очевидна. Поэтому существует точка зрения, согласно которой точное определение и отграничение И. от не-И. невозможно, а

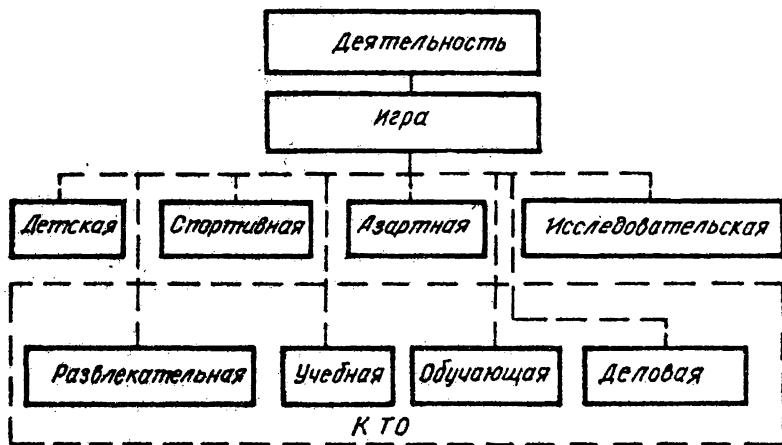
природа И. вообще не может быть сформулирована. Вследствие этого различные авторы оперируют (явно или неявно) рабочими определениями, которые (в зависимости от исследовательских целей) опираются на одно из возможных представлений И. и акцентируют те либо иные ее признаки или аспекты. Напр., для исследований в области детской, игры наиболее характерно ее представление как деятельности, опосредствующей познавательное, социальное развитие и формирование аффективно-эмоциональной сферы ребенка. Это вполне оправданно применительно к возрасту 3—7 лет, когда игровая деятельность оказывается ведущей, а также к И. развлекательного и спортивного плана, в которых цели, средства и продукты деятельности закрепляются в рамках самой игры. Если же И. содержится в иных видах человеческой деятельности, используются различные представления об И.: как о *модели* действительных и мнимых ситуаций, как о структуре социального действия, как об особом отношении человека к миру, к людям, к самому себе и т.д. Это же относится и к атрибутивным признакам И., фигурирующим в различных ее определениях. Чаще всего отмечаются: неутилитарный характер, самооценность процесса игровой деятельности, ее особая эмоциональная привлекательность, кооперативный характер И. с выделением случаев сотрудничества, конфликта, соревнования (см. *Соревновательная игра*), указания ролей-функций и др. (напр., *деловые игры*). При этом наблюдается тенденция к расширительному толкованию И., когда каждый игровой прием, соревновательный момент, введение в сюжет “игрового персонажа” (напр., изображение муравья, черепашки и пр., а иногда и сам факт использования компьютера) дают основания некоторым авторам судить о деятельности в целом как об игровой. Часто И. предполагает имитацию — “живое изображение основных черт реальности” В таких случаях говорят об *имитационной игре*, сочетающей признаки имитации (игровые роли, моделирование реальности) и признаки И. (стремление к цели, специфические правила и пр.). Соответственно, *учебная задача*, в качестве которой может выступить *учебная игра*, приобретает черты, характерные для учебных задач на имитацию и анализ конкретных ситуаций (некоторые авторы считают: целесообразным употреблять в таких случаях термин “имитационные обучающие игры”). Принципиальное значение имеет уточнение деятельностного представления И. Традиционные указания на самооценность процесса игровой деятельности, в результате которого происходит “сдвиг с продуктов на процесс” (т.е. основной смыслообразующей детерминантой игровой деятельности является процесс, а не результат), не уменьшают значимости ее результата. Часто встречаются дефиниции, в которых игра определяется как вид непродуктивной деятельности, где мотив лежит вне ее результата. Отрицание продуктивности игровой деятельности основывается в данном случае на том, что результаты не носят практически

полезного, утилитарного характера. Однако это не означает, что результата нет вообще и что он никак не мотивирует процесс. Это утверждение характеризует самого играющего, однако для других лиц продукты И. могут быть практически полезными. Результат И. неотделим от субъекта и не является, по выражению амер. ученых Дж. Неймана и О.Моргенштерна, трансферабельным, т.е. "передаваемым". Здесь под словом "результат" подразумевается прямой продукт, т.е. соответствующий основной, осознаваемой игроком цели своих действий. Однако утверждение о практической полезности процесса необходимо, иначе к игровой деятельности будут отнесены все виды деятельности, результаты которой неотделимы от субъекта (в т.ч. учебной, продукты которой заключаются в изменении самого субъекта деятельности).

В структуре И. целесообразно выделять три основных этапа: подготовку к И. (введение в И.); проведение самой И. (процесс И., тело И.); анализ И. и ее результатов (заключение И.). Такое членение имеет особое значение для учебных и деловых игр, поскольку позволяет локализовать воздействия внешней системы, в частности *обучающие воздействия* (в учебной И. они, как правило, имеют отстроченный характер — предъявляются после завершения И. либо отдельных ее фрагментов или, наоборот, предваряют И., актуализируя те или иные учебные цели. В И. различают также ее сюжет (модель реальной действительности, отражаемой в игре), сценарий (программа действий участников игры), который следует отличать от педагогического сценария, фабулу (совокупность основных сюжетных линий, "ядро сюжета"), содержание (*предметная область*, на материале которой осуществляется игра).

На практике целесообразно различать два основных класса И. — game и play, полагая, что в первом случае речь идет об И. с правилами, протекающей в рамках некоторой знаковой системы и направленной на достижение определенного результата, конституирующегося в этой же знаковой системе, а во втором — об И., протекающей в некоторой игровой среде (см. *Компьютерная учебная среда*) и не связанной с определенной целью (рис.21). Примером И. в первом смысле могут служить распространенные *компьютерные игры*, являющиеся имитацией различных конфликтов, во втором смысле — деятельность школьников в компьютерных учебно-игровых средах LOGO, Элкон и др. В дидактическом плане И. во втором смысле (т.е. деятельность в игровой среде) наиболее продуктивна. Учитывая больше возможности построения компьютерных игровых сред, следует обратить особое внимание именно на это направление *компьютерного обучения*.

Существует множество различных классификаций И. Напр., возможна классификация, основанием которой является ведущая цель (рис.). В *компьютерной технологии обучения* применяют следующие виды И.: развлекательные (приключенческие) для развития навыков



работы с клавиатурой, адаптации пользователей к новой технике, психологической разрядки; учебные для поддержки самостоятельной учебной деятельности пользователей в приятной эмоциональной среде, тренировки навыков; обучающие для неявного управления учебной деятельностью (см. Обучающая игра); деловые для поиска оптимальных управленческих решений и усвоения навыков работы в коллективе. В компьютерных И. одна из играющих сторон может быть реализована как программная модель.

Порождение И. связано прежде всего с двумя моментами. Первый — это отклонение от норм деятельности (напр., атрибутивным признаком учебной И. является наличие правила, “порождающего игру”, не заимствованного из контекста соответствующей учебной задачи, не формализованного в соответствии с процессом деятельности или ее продуктами и воспринимающегося как “условность”). Второй — аксиологический аспект деятельности, связанный с отсутствием или иллюзорностью санкций. Оба аспекта связаны между собой, поскольку новация в технологии (изменение норм деятельности) влечет за собой изменения и в аксиологическом плане, причем на первых порах контроль и оценка нового операционного состава деятельности носят искусственный или иллюзорный характер из-за отсутствия социально-нормированной технологии (хотя формально она может быть задана).

Рассмотрим И. с точки зрения задачного подхода к исследованию взаимодействия. Любая И. включает три взаимосвязанные стороны: игровое состояние, игровое общение и собственно игровую предметную деятельность. Рассмотрим более подробно деятельность в

компьютерной И. Такая деятельность с точки зрения играющего представляет собой определенную задачу — в ней имеются условия, цель и средства ее достижения.

Деятельность D_3 по решению задачи определяется как тройка

$$D_3 = (K, \Pi, O),$$

где K — предметная область или ее модель K' ; Π — множество целей (Π_i , $i=1, \dots, n$; $n \leq M$, $M \in N$); O — множество критериев оценки (O_j). Для каждой задачи Z_k , определенной на предметной области K

$$K = (K', K_{\text{акт}}, K'_{\text{тр}}),$$

где K' — модель предметной области; $K_{\text{акт}}$ — ее исходное состояние (условия задачи); K — модель требуемого состояния (результата решения задачи), а сам результат включается в K ($K \in K'_{\text{тр}}$). Нужно отметить, что $K'_{\text{тр}}$ формируется некоторой целью Π_i . Предметная область K представляет собой тройку:

$$K = (A, R, Q),$$

где A — множество всех объектов данной предметной области (A_i); R — множество отношений между объектами (R_j); Q — множество операций над отношениями и объектами (Q_j). Процессом решения задачи Z_k является переход $K_{\text{акт}} \rightarrow K'_{\text{тр}}$, выполняющийся посредством применения некоторого подмножества операций Q_3 из множества Q над исходным состоянием предметной области $K_{\text{акт}}$.

При рассмотрении одной из целей Π_i как перехода $K_{\text{акт}} \rightarrow K'_{\text{тр}}$ получено более узкое определение деятельности:

$$D_3 = (Z_k, O),$$

т.е. деятельность представляет собой задачу Z_k и оценку эффективности ее решения (имеется в виду деятельность одной решающей системы, которой можно считать человека, программу на компьютере, коллектив). Некоторые типы критериев, характерных для компьютерных И.: временные, когда учитывается время решения задачи; эталонные, когда имеется эталон результата решения; балловые, когда эффективность задачи оценивается некоторой суммой баллов. Кроме того любая деятельность целенаправлена, т.е. её характер определяется множеством целей. Структура множества целей и определяет вид деятельности. Так, если в множестве целей Π превалирует некотора Π_u — получить результат K_u (знания, умения и навыки), то такая деятельность является учением. Для игровой деятельности характерны два типа целей: получить игровой результат K_i (решить игровую задачу) либо эмоциональный результат от участия в игре (удовольствие, наслаждение). Для И. характерно взаимодействие как минимум двух сторон, поэтому необходимо анализировать пару целей взаимодействия (см. *Конфликтное взаимодействие, Кооперативное взаимодействие*).

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

ИГРОВАЯ ЗАДАЧА — задача, решаемая в игре по правилам. Правила игры накладывают ограничения на средства решения задачи. С точки зрения теории задач, И.з. помимо фактуального и требуемого состояния предметной области, обязательно включает оценку деятельности и действий по достижению игрового результата. Это обусловлено наличием как минимум двух игроков, связанных взаимодействием по решению И.з. Они либо соревнуются — кто лучше или быстрее решит данную И.з., либо создают сопернику помехи для решения И.з., либо сочетают первые два способа взаимодействия (см. Конфликтное взаимодействие, Соревновательная игра). Критерии оценки И.з.: сумма баллов, время решения, минимум израсходованных ресурсов и др. Оценка решения И.з. каждого игрока осуществляется по одному или нескольким критериям. Значение оценки определяет победившего игрока, который из игровой предметной области получает специфический объект — поощрение (моральное или материальное). И.з. часто включает учебную задачу (см. Учебно-игровая задача).

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.С трижак.

ИГРОВАЯ МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ — модель предметной области, содержащая игровые элементы (см. Игра). Часто в качестве И.м.п.о. служит имитационная модель реальной или воображаемой действительности. Однако в ней происходит искажение некоторых норм, т.е. имеется хотя бы одно условное правило, порождающее игру (см. Учебная игра). Примером И.м. п.о. может служить модель предметной области в игре “морской бой”, в которой играющие осваивают понятие “система координат”. И.м.п.о. обеспечивает социальную интерпретацию соответствующих учебных операций и действий, что и является образующим фактором игры.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.С трижак.

ИЗЛУЧЕНИЕ электромагнитное — процесс образования свободного (самоподдерживающегося) электромагнитного поля. Электрические и магнитные поля не существуют обособленно друг от друга: изменения одного из них вызывают изменения в другом. Электромагнитное поле заряда включает в себя как собственное поле, находящееся непосредственно вблизи заряда, так и уходящее вдаль от заряда поле И. Согласно классической теории И. — это испускание электромагнитных волн при неравномерном движении электрических зарядов либо при протекании переменного тока. Поскольку скорость протекания электромагнитных волн — конечная величина (в вакууме $c = 3 \cdot 10^8$ м/с), то изменение положения заряда вызовет изменение электромагнитного поля в расположенной на расстоянии R точке не мгновенно, а спустя некоторое время $t=R/c$. Поэтому возможно существование свободного, поддерживающегося независимо от источ-

ника И. электромагнитного поля. Взаимное влияние движущихся зарядов можно оценить только с учетом параметров электромагнитных полей, сформированных ими. Уносимая И. энергия пропорциональна векторному произведению напряженности электрического и магнитного полей. Электромагнитное поле дополнительно характеризуется скалярным и векторным потенциалами. По теории относительности составляющие векторов образуют тензор электромагнитного поля, компоненты которого при переходе к другой системе отсчета видоизменяются в соответствии с преобразованиями Лоренца. В зависимости от физической природы или причин, вызывающих изменение скорости движения заряда, возможны различные типы И., имеющие свои конкретные способности (дипольное, тормозное, синхротронное и т.д.). Квантовая теория рассматривает И. как рождение фотонов (квантов электромагнитного поля) при изменении состояния квантовой системы (ядра, атома). Внутренняя энергия квантовой системы может иметь только определенные дискретные состояния. Изменение состояния системы происходит скачкообразно и сопровождается испусканием (поглощением) фотона.

Источником И. при использовании видеотерминалов являются радиоэлектронные элементы конструкции и люминофоры. Численные значения параметров И. и их влияние на здоровье персонала в настоящее время изучены недостаточно. Существуют лишь оценочные значения параметров поля И., измеренные непосредственно на стенках видеотерминалов: напряженность электростатического поля — 5 — 300 кВ/м; напряженность низкочастотного электромагнитного поля ($< 1\text{МГц}$) — 10 — 500 В/м; напряженность высокочастотного электромагнитного поля ($> 1\text{МГц}$) — 5 — 100 В/м; поток инфракрасного излучения — 0,1 — 10 Вт/м; тепловой поток — 10 — 100 Вт/м; световой поток — 10 — 100 кд/м; поток ультрафиолетового излучения 0,1 - 10 Вт/м; мощность дозы излучения — 30 — 150 мкР/ч.

М.Н.Коротенко.

ИЗОМОРФИЗМ (от греч. *ἴσος* — равный одинаковый и *μορφή* — форма) — отношение между двумя системами, состоящее в идентичности их структур, точнее, в наличии взаимно однозначных соответствий между компонентами систем и отношениями, связывающими эти компоненты. Напр., изоморфны молекула некоторого вещества и его структурная формула (если она верна). Если системы изоморфны, то каждую из них наз. также изоморфным образом другой. См. также *Гомоморфизм, Модель*.

Г.А.Балл.

ИМИТАЦИОННАЯ ИГРА — игровое взаимодействие с имитационной моделью некоторой системы или процесса. Подвидом И.и.

являются *деловые игры*. К И.и. относятся также ролевые и организационно-деятельностные *игры*. Важный элемент И.и. — понятие роли, охватывающее существенные черты мотивов деятельности личности, группы, коллектива, участвующих в игре. Особенностью И.и., реализованных на базе компьютерных имитационных моделей, является возможность имитационного сжатия времени (год можно сжать до часов или минут, что позволяет оценивать множество принятых решений и последствий от непринятых решений).

И.и. в целом воспроизводит атрибутивные характеристики имитируемой реальной ситуации. Если целью предъявления И.и. является получение учебного эффекта (усвоение *обучаемым* соответствующих норм деятельности, овладение способами действий, формирование необходимых *знаний, умений и навыков*), игра наз. имитацией *обучающей*.

Игру, в которой воспроизводятся существенные признаки моделируемого объекта, но в замещённой форме (напр., схемы и знаки замещают составные части моделируемых объектов), можно назвать моделирующей.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

ИМИТАЦИОННАЯ МОДЕЛЬ — искусственная система, имитирующая изучаемое явление или реальную систему. Используется для анализа возможных реакций объекта исследований на управляющие и возмущающие воздействия. В И.м. выделяют такие составляющие, как компоненты, параметры, функциональные зависимости, ограничения и целевые функции. И.м. применяют в экономических исследованиях, военном деле, *экспертных системах, обучении на тренажерах* и имитаторах и др. В тренажерах и имитаторах И.м. представляет собой *модель* объекта управления, с которой взаимодействует тренируемый, находящийся в обстановке, имитирующей его реальное *рабочее место*.

В частном случае под И.м. понимают специальный вид матем. модели, когда не существует ее аналитического описания в приемлемой форме. Процесс составления И.м. наз. *имитационным моделированием*.

С.Т.Адрианов.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ — вид моделирования процессов, происходящих в исследуемых системах, с воспроизведением входных сигналов и получением количественных и качественных оценок функционирования систем. Применяется для анализа и оптимизации сложных систем в тех случаях, когда не удается создать приемлемых аналитических *моделей*. Сущность И.м. заключается в разработке *имитационной модели* процесса функционирования реальной системы с необходимым уровнем детализации и

проведении имитационных экспериментов с помощью целенаправленных изменений параметров модели. При этом обеспечивается более быстрое, дешевое или безопасное выявление сущности моделируемых систем, чем при экспериментировании с самими системами. И.м. стало бурно развиваться в связи с распространением и совершенствованием компьютеров... Это обусловило разработку большого числа систем программирования, включающих специализированные языки программирования (GPSS, SIMSCRIPT, SIMULA, SIMPL/1, SOL, СЛЭНГ и др.), позволяющих разработчикам уделять большое внимание собственно моделированию, а не техническим деталям реализации моделей.

В контексте компьютерной технологии обучения И.м. рассматривается в более узком смысле. Так, в специальных имитаторах, тренажерах, интегрированных обучающих тренажерных системах И.м. призвано обеспечить имитацию процессов, происходящих в реальных системах, навыки управления которыми отрабатывает тренируемый. Входными сигналами здесь являются действия тренируемого, а цель его деятельности заключается в приведении имитационной модели в одно из обусловленных (целевых) состояний. Ко входным сигналам относятся также вводные инструктора, которые интерпретируются как возмущающие воздействия на имитационную модель. Текущие характеристики моделируемого объекта поступают на средства отображения, в которых они преобразуются в сообщения, показания контрольноизмерительных приборов, различных индикаторов и табло.

С.Т.Адричнов.

ИМПЕРАТИВНЫЙ ЯЗЫК — алгоритмический язык, предоставляющий средства для детального определения последовательности действий компьютера, необходимых для решения задачи. В основе И.я. лежит операционная модель процесса вычислений, согласно которой выполнение программы заключается в преобразовании состояния посредством оператора присваивания. По технологической направленности к И.я. относится большинство известных языков программирования (Фортран, Паскаль, Ада, Смолток и др.), а также некоторые языки, используемые для спецификации (SETL) и проектирования (PDL) программ. По уровню проблемной ориентированности И.я. делятся на проблемно-ориентированные (Кобол, Тьютор, язык описания курсов и др.), универсальные (Ада, Паскаль, Смолток и др.) и машинно-ориентированные (ассемблера язык, Смолток для Смолток-машины).

В.А.Третьяк.

ИНДИВИДУАЛИЗАЦИЯ ОБУЧЕНИЯ — организация учебного процесса, при которой методы, способы, приемы, темп обучения

выбирают на основе учета индивидуальных различий *обучаемых* (уровня развития их способностей к *учению*, подготовленности и пр.).

Различают индивидуальное, адаптивное и индивидуализированное обучение. Первое понятие характеризует организационную сторону обучения, второе — обучение, учитывающее возрастные и индивидуально-психологические особенности обучаемых, третье — обучение, которое основывается на *модели обучаемого* и выдает *обучающие воздействия* с учетом этой модели. Очевидно, И.о. является высшим уровнем адаптации, позволяет организовать обучение наиболее эффективно, что достигается за счет оптимального распределения учебного времени (дифференциация режимов, темпов выполнения заданий), организации учебного материала в соответствии с особенностями обучаемого (дифференциация заданий по цели, сложности, трудности и т.д.), выбора адекватного метода обучения и особенностей управления *учебной деятельностью* обучаемого. Задачи, которые в условиях традиционного обучения чаще всего лишь декларируются, при наличии компьютера можно реально решить. Возможны три режима И.о.: при первом из них выбор обучающих воздействий полностью определяется компьютером, при втором — наиболее предпочтительный способ управления выбираю сами обучаемые, при третьем — предлагается смешанное управление (напр., при неэффективных действиях обучаемого управление “берет на себя” компьютер).

В условиях применения компьютера И.о. осуществляется на основе динамической модели личности и деятельности обучаемого, обеспечивающей адекватный выбор обучающих воздействий. Модель обучаемого представляет собой частный случай модели партнера, являющейся обязательным компонентом структуры коммуникации в *общении человеко-машинном* и отражающей ожидания в отношении человека-коммуниканта, что необходимо для корректного понимания системой сообщений последнего. Модель обучаемого должна учитывать интеллектуальные, эмоционально-личностные и мотивационно-потребностные характеристики.

В множестве индивидуальных особенностей, существенных в ситуации обучения, можно выделить четыре осн. типа характеристик, располагающихся на шкале изменчивости по возрастающей, — от стабильных до определяемых преимущественно сиюминутным воздействием окружающей среды: личностные факторы, когнитивные (познавательные) стили, *стратегии* учения, субъективная структура *знаний*. К личностным факторам относятся интеллект, экстраверсия/интроверсия, тревожность, креативность. Параметрами когнитивного стиля выступают прежде всего полнезависимость, внушаемость, критичность, образность (вербальность, гибкость), ригидность, операциональное/“понимающее” (концептуальное) усвоение. Стратегии учения, формируемые у обучаемых, классифицируются по

признакам эвристичности/систематичности, сериализма/холизма. Индивидуальная структура знаний включает правила вывода, схемы, семантическую сеть. Некоторые из этих характеристик столь существенно влияют на процесс обучения, что требуют адаптации к ним обучающей системы (см. *Адаптивность системы обучения*). Различение типов *пользователей* возможно и по соотношению и сочетанию параметров когнитивного стиля. Включение этих характеристик в психологический портрет пользователя позволяет оптимизировать *структуру диалога*, поскольку экспериментально подтверждено, что когнитивные стили надежно прогнозируют учебное поведение в системе "учащийся-компьютер". Адаптация содержания и структуры учебного материала к индивидуальным особенностям и уровню подготовки обучаемого дает положительные сдвиги прежде всего в обучении слабых учащихся. Не умея выстроить оптимальную последовательность заданий самостоятельно, они иногда показывают даже лучшие результаты, чем сильные учащиеся, если процесс обучения проходит под управлением компьютера и спроектирован в соответствии с *принципами обучения*. В некоторых ситуациях *компьютерного обучения* уровень интеллекта вообще не влияет на результаты (напр., поиск информации в *базе данных*), хотя при обучении без компьютера различия несомненны. С другой стороны, при изучении *языков программирования* обычно отмечается значительная *корреляция* между показателями деятельности и уровнем интеллекта. Но при этом даже слабые обучаемые демонстрируют определенный уровень креативного поведения в использовании языков программирования, обнаруживая подчас те способности, которые в условиях традиционного обучения не удастся раскрыть.

И.о. — важный показатель качества компьютерного обучения, характеризующий степень приспособления комплекса технических, программных, учебно-методических средств КТО, в частности *автоматизированных обучающих систем* (АОС), к условиям реализации конкретной учебной процедуры и индивидуальным особенностям обучаемого. Она представляет собой осн. цель функционирования адаптивной АОС и составляет главное условие организации гибкого учебного процесса, предполагающего усвоение знаний, умений и навыков в удобном для обучаемого темпе, с глубиной изложения зависящей от способностей или заинтересованности пользователя. В КТО процесс обучения можно считать индивидуализированным, если: а) каждый обучаемый имеет собственное *автоматизированное рабочее место*, оборудованное интеллективным *терминалом*; б) *программы учебного назначения* позволяют реализовать сложные *алгоритмы* многокритериальной оценки знаний, умений и навыков, диагностику состояния и поведения обучаемого и включают развитый инструмент для принятия решения по управлению учением (см. *Экспертные обучающие системы*); в) лингвистические средства позволяют обеспечить приближенный к

естественному диалог с обучаемыми по изучаемой предметной области и др.

Е.Ю.Комиссарова, Е.Д.Маргулис, Л.Е.Михеева, В.Н.Сороко.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА для представления знаний — программные средства представления и обработки знаний. Существует множество И.с. манипулирования знаниями, применимых при разработке интеллектуальных систем — от универсальных языков программирования (ЯП), таких, как *Лисп, Паскаль и Фортран*, до универсальных систем представления знаний. Осн. подмножества И.с.: универсальные ЯП; инструментальные системы; универсальные языки представления знаний.

Манипулирование знаниями использует различные программные парадигмы: логическое программирование (Пролог); функциональное программирование (Лисп); объектно ориентированное программирование (SMALLTALK, FLAVORS). Каждый ЯП является языком представления знаний. При этом в качестве декларативных знаний выступают описания, а в качестве процедурных — последовательности операторов, реализующие тот или иной алгоритм (см. *Методы представления знаний*). Универсальные ЯП неудобны для представления знаний, т.к. построить такое представление — весьма трудоемкий и сложный процесс.

Вместо того, чтобы каждый раз с самого начала разрабатывать новую систему манипулирования знаниями, можно позаимствовать некоторые из ранее построенных систем. В результате этой стратегии появилось несколько новых программных средств, напр., EMYCIN или EXPERT (полученная из CASNET), которые можно охарактеризовать как инструментальные системы. Ныне все чаще используются инструментальные системы, опирающиеся на знания. Существуют два подхода к их разработке. В рамках первого из них инструментальная система погружается в имеющееся программное обеспечение, в рамках второго — в базовую технологическую систему. Примером первого подхода может служить система СТЕНД, которая развивается в настоящее время как технологический комплекс, предназначенный для конструирования достаточно широкого класса систем, использующих знания. В качестве базового инструментального ЯП в системе СТЕНД используется язык SETL. Пример реализации второго подхода к разработке инструментальных систем — комплекс генерации программного обеспечения систем общения с архивами и базами данных — ПОСОХ. Центральным моментом системы является комплекс средств для представления лингвистических знаний. Исходная программа, специфицируя словарную информацию и анализатор, преобразуется во внутреннее представление, которое может быть скомпилировано в объектную программу.

Одной из категорий программных средств, используемых для построения баз знаний (БЗ), являются универсальные языки представления знаний (ROSIE, C.SS, RLL, HEARSA-3 и ФОРТЕ, используемый для конструирования сетевого представления знаний). Применение их по сравнению с инструментальными системами требует меньше ограничений, поскольку они позволяют использовать более широкое множество управляющих структур. Следовательно, эти языки применимы к более широкому классу задач, хотя сам процесс их использования может оказаться более сложным, чем в случае инструментальных систем.

При разработке обучающих систем целесообразно использовать инструментальные системы, обеспечивающие механизм вывода и предоставляющие конкретные методы представления знаний по предмету обучения, его методике, средствам диагностики ошибок, т.к. использование универсальных ЯП и универсальных языков представления знаний может сопровождаться значительными трудностями. С увеличением объема БЗ потери в эффективности обработки знаний, связанные с декларативными и однородными представлениями, становятся все более ощутимыми. Особое значение приобретает разработка И.с., позволяющих повысить эффективность процесса обработки знаний. К средствам, обеспечивающим эффективность БЗ, относятся: использование специализированных структур знаний, компиляция знаний и преобразование знаний. Структуризация знаний сказывается на скорости процесса извлечения информации. Как правило, эти структуры построены так, чтобы совместно используемые знания совместно и запоминались, а часто используемые знания извлекались достаточно быстро. Выбор структур знаний зависит от предположений относительно того, как эти знания будут использоваться. Распространенным предположением относительно специальных структур знаний является то, что они должны быть полны в отношении некоторых конкретных связей. Напр., в структурах знаний, предназначенных для представления гипотез о молекулах, сегменты молекул соединены тогда, когда они связаны и в структуре знаний. Большое значение имеет компиляция знаний — процесс преобразования одного представления знаний в другое, допускающее более эффективное использование. Этот процесс преобразования может включать как оптимизацию, так и приведение представлений в соответствие с конкретным множеством операций.

В.В. Колос.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ВИДЕОКОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — набор программных средств, обеспечивающих полноценное функционирование процесса обучения с применением видеотехники. В состав И.с.в.т.о. входят программные обеспечения *видеокomпьютерного рабочего места преподавателя*

(ВРМП), *обучаемого* (ВРМО) и видеогиды (*информационной системы*). Программное обеспечение ВРМП включает: авторскую систему создания учебных программ; специализированный видеокомпьютерный модуль, допускающий вызов любой авторской системы, любого фрагмента уже разработанной учебной программы после минимальной и элементарной редакции; драйверы управления контроллерами в ВРМП.

Авторскую систему поддерживает набор редакторов, обеспечивающих нормальное функционирование системы и создание обучающих курсов — *шрифтовой редактор; текстовый редактор; графический редактор* (для свободного рисунка, статистической графики, точных чертежей и др.); *связывающий редактор* (проектор); *видеокомпьютерный редактор* (модуль); *авторский редактор* для создания обучающих курсов. Эта система должна отвечать следующим требованиям: а) работа с одним или двумя *дисплеями* (графическим и текстовым), которые, наряду с обеспечением возможности работы на профессиональном *рабочем месте*, автоматизации инженерного труда, позволяют разработку интерактивных (с точки зрения *преподавателя*) программ; с помощью *меню* и подменю, вызываемых на текстовом дисплее, преподаватель выбирает необходимую *информацию* или *алгоритм* её представления, адаптируясь к интеллектуальному уровню и знаниям аудитории; б) совместимость с языками высокого уровня (см. *Уровень алгоритмического языка*) и возможность демонстрации программных пакетов; в) доброжелательный и упрощенный *диалог* (обеспеченный вспомогательной информацией), облегчающий работу массового потребителя.

Специализированный видеокомпьютерный модуль разработан в двух вариантах: 1) авторском, состоящем из редактора управления и маркирования видеофрагментов видеокассетофона и редактора вызова цифровых и аналоговых слайдов; 2) потребительском, состоящем из программных драйверов управления и обратной связи о состоянии видеокассетофона; управления автоматическим диапроектором, управления видеопроекционной системой; воспроизводящего драйвера, обеспечивающего связь между модулем и обучающим курсом или редакторами. Этот модуль не поддерживает списки, таблицы и т.д. о требуемом фрагменте или диaposитиве и удовлетворяет следующим требованиям: резидентность установки; вызов посредством программного прерывания языка высокого уровня; совместимость и передача параметров языка высокого уровня.

Для подключения к рабочему месту преподавателя и обучаемого различных видеоустройств необходимы драйверы, обеспечивающие одинаковый формат передаваемых параметров, определяющих данную функцию. Это обеспечивает унификацию модуля к *программам*, написанным на разных языках высокого уровня, т.е. различные (созданные или создающиеся) авторские системы, редакторы или обучающие курсы не связаны с типом используемого устройства.

Технические и программные драйверы рекомендуется организовать так, что во время поиска определенного фрагмента процессор мог бы работать по определенной в данной цели программе. Это даст возможность заполнить время ожидания для обнаружения требуемого видеофрагмента с нужной информацией — аннотацией, описанием, техническими данными, рекламами и др. в информационных системах.

Программные драйверы управления контроллерами в ВРМП осуществляют связь между программным обеспечением, написанном на языках высокого уровня, и непосредственным управлением компонентами в ВРМП посредством периферийных схем. Поскольку время выполнения некоторых функций является критическим, драйверы написаны на *ассемблера языке*. Они разные для каждого контроллера и для каждой модели видеоприбора. Так можно удовлетворить специфическим требованиям любого протокола обмена между контроллером и видеоприбором. Общее между различными видами драйверов — однотипный способ осуществления связи с языками высокого уровня. Для этой цели используется программа прерывания. Обмен параметрами осуществляется с помощью регистров *микропроцессора*. Этот способ соответствует концепции, по которой разрабатываются программные драйверы управления периферийными устройствами в компьютерах. Программист, пользующийся этим драйвером, может абстрагироваться от различий в периферийном оснащении. Таким образом программное обеспечение, создаваемое на языках высокого уровня, может унифицироваться.

По структуре и алгоритму работы драйверы одинаковы. После старта начинается тестирование контроллера и корректной связи видеомагнитофона. При неисправности выводится сообщение, и программа переустанавливается. Если все в порядке, вектор прерывания направляется на точку входа драйвера, после чего управление предоставляется *операционной системе* ДОС, причем программа обработки прерывания остается резидентной. Она состоит из программ, вызываемых в зависимости от содержания регистра. Первая из них является программой первоначального инициализирования. Она включает определение направления обмена регистров АВВ, режимы и коэффициенты отсчета таймера, возврат видеопленки в первоначальное положение и инициализирование рабочих ячеек, которые драйвер использует для сохранения текущей информации. Последующие программы управляют режимами работы видеокассетфона. Другая функция драйвера заключается в поддержке информации о местонахождении пленки. Это осуществляется путем вычислений на базе данных со счетчиков таймера. Позиционирование пленки на определенном кадре происходит посредством аппаратного заказа на перерыв, генерируемого таймером. Предварительное инициализирование, как и обработка аппаратного прерывания, осуществляется *подпрограммами*, размещенными в резидентной части

драйвера. На этом же принципе работает и программа воспроизведения видеофрагмента до определенного кадра. Последние программы связаны с управлением видеопроекционной системой и диапроектором. Они одинаковы для всех драйверов. Текущее состояние видеомagneфона и драйвера отражается флажками, которые посредством регистров доступны для программ высокого уровня.

Если при создании обучающих курсов размещение видеофрагментов осуществляется аналогично алгоритму обучающего курса, то во время обучения работающий драйвер и обучающая программа прогнозируют, насколько это возможно, следующий шаг и заранее подводят пленку к необходимому фрагменту, так что при необходимости он сразу воспроизводится. Если это невозможно или программа имеет более низкую степень вероятности, поиск и воспроизведение необходимого фрагмента происходит при возникшей необходимости, причем отставание видеоалгоритма на один шаг, т.е. обнаружение нужного фрагмента, осуществляется за десятки доли секунды. Модуль должен сопровождаться дополнительной текстовой программой уже выбранных видеофрагментов и аналоговых или цифровых слайдов.

Г.Ж.Желев.

ИНСТРУМЕНТАЛЬНЫЙ ТЕХНИЧЕСКИЙ КОМПЛЕКС — программно-техническая компьютерная система для разработки, сопровождения, совершенствования, хранения, тиражирования *автоматизированных обучающих систем и автоматизированных учебных курсов*. Комплекс может быть использован и как вычислительное средство в компьютеризированных учебных заведениях с развитой *компьютерной технологией обучения* (КТО). Отличительные особенности И.т.к.: более высокие вычислительные ресурсы по сравнению с учебными компьютерами или комплексами в части объема оперативной и внешней памяти, разнообразие *носителей информации* и устройств длительного её хранения, *интерфейсов* для связи с внешними устройствами, *устройств ввода — вывода информации*.

Обычно И.т.к. создаются под определенные задачи. Так, для разработки элементов и систем КТО исследований в этой области, ориентированных на применение персональных ЭВМ, выбирается наиболее мощная ЭВМ этого класса с развитой периферией, базовым и специальным *программным обеспечением*, сетевыми интерфейсами и другими средствами. В состав технических средств многомашиного И.т.к. для исследования и обучения компьютерным технологиям входят ЭВМ серии СМ и ПЭВМ ДВК, Корвет, Ямаха, функции которых обеспечиваются локальными вычислительными комплексами (мини-сетями) СМ-ДВК, СМ-КУВТ "Корвет", СМ-КУВТ "Ямаха". На И.т.к. можно осуществлять перенос *файлов* и *текстов* программ из ПЭВМ на внешние носители СМ ЭВМ и наоборот, загрузку

ПЭВМ локальной сети (КУВТ) от СМ ЭВМ, другие операции, обеспечивающие эффективную организацию учебной работы.

И.А.Емченко, В.Н.Никулин.

ИНТЕГРИРОВАННАЯ БАЗА ДАННЫХ — база данных (БД), позволяющая одновременно и совместно использовать несколько разнотипных БД как единое целое. Интегрированная совокупность различных БД по отношению к прикладной программе должна логически выглядеть как единая БД. Интегрированная система управления базами данных (СУБД) требует совместного использования всех соединенных в ней БД. Интегральное представление этих БД не есть их простая сумма. Требуется не только вводить однородное представление данных, давать общие средства манипулирования ими, позволяющие устранять различия в исходных моделях данных, но и создавать абстракции данных более высокого порядка, которые в большей степени соответствовали бы условиям совместного использования данных.

Осн. идея интеграции БД заключается во введении воображаемого представления данных, в которое должна быть отображена каждая из интегрируемых БД. Этому уровню представления данных соответствует вполне определенная модель данных, в которую эффективно можно преобразовать модели данных произвольных СУБД. Такая модель данных наз. *концептуальной моделью данных* системы интеграции. Концептуальные модели данных, поддерживаемые реальными СУБД, по отношению к этой общей модели выступают в качестве внутренних моделей. Для обеспечения условий интеграции произвольных БД концептуальная модель данных системы интеграции должна включать средства, позволяющие работать одновременно как со структурированными, так и с неструктурированными данными. Над концептуальной моделью данных системы интеграции можно надстраивать модели данных внешнего уровня. Включающие языки программирования и соответственно прикладные программы опираются непосредственно на концептуальный уровень системы интеграции.

В архитектуре системы интеграции БД отражаются принципы построения многоуровневых систем стремление к макс. независимости представлений данных на различных уровнях и изоляция одного уровня от другого с тем, чтобы ослабить влияние изменений в представлении данных более высоких уровней; использование представлений данных и средств манипулирования данными низлежащих уровней при реализации функций манипулирования данными каждого уровня; обеспечение независимости реализации функций каждого уровня от реализации функций других уровней. Главной целью введения многоуровневой архитектуры является превращение системы интеграции в открытую систему, позволяющую включать в

все произвольные СУБД и БД на основе должным образом выбранной системы интерфейсов.

А.П. Ильинченко, М.Е. Козлов.

ИНТЕГРИРОВАННАЯ ОБУЧАЮЩАЯ ТРЕНИРУЮЩАЯ СИСТЕМА — совокупность *автоматизированной обучающей системы и тренажеров*, предназначенная для формирования и закрепления *знаний, умений и навыков*. Обычно подготовку специалиста с помощью тренажера можно разделить на два осн. этапа — предтренажерную и собственно тренажерную. Осн. задачей предтренажерной подготовки является изучение операторами эргатических (человечно-машинных) систем общетеоретических дисциплин, конструкций и компоновок систем, принципа и логики их функционирования в штатных и нештатных режимах работы. Осн. задача тренажерного этапа подготовки — овладение навыками и умениями в управлении сложными системами в различных режимах работы. Тренажерная подготовка операторов сложных эргатических систем обычно проводится на дорогих и сложных комплексных тренажерах, поэтому целесообразно большую часть подготовки возложить на предтренажерный этап. Для реализации этой возможности и предназначены И.о.т.с. Использование их позволяет осуществить следующие формы обучения: лекции, *программированное обучение*, практические занятия (тренаж) по отработке задач управления отдельными подсистемами эргатических систем, подготовка учебных курсов и заданий на тренаж, регистрация *данных о процессе обучения*. Достоинства И.о.т.с.: комплексный подход к проблеме приобретения теоретических знаний и практических навыков и умений, сведение к минимуму разрыва между ними; повышение наглядности обучения, прочности усвоения как теоретического, так и практического материала. Наиболее эффективно такие системы можно реализовать на базе современных компьютеров в комплексе со специализированными техническими средствами (аудиовизуальными устройствами, натурными и полунатурными стендами и т.п.).

С.Т. Адрианов.

ИНТЕГРИРОВАННЫЙ ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ — совокупность *пакетов прикладных программ (ППП)*, целью создания которых является объединение “под одной крышей” наиболее часто используемых прикладных программ. Преимущества И.п.п.п. состоят прежде всего в том, что они удобны для *пользователя* и решают класс задач, для которого были созданы. Система предоставляет удобные средства доступа к различным пакетам. При интеграции пакетов важнейшие требования — достижение требуемого соответствия типов данных языка программирования (ЯП), удовлетворение соглашений о межпрограммных связях.

Недостаток интегрированных пакетов по сравнению с изолированными — увеличение занимаемого объема оперативной памяти ЭВМ и времени работы. Напр., для использования “чистого” ЯП Пролог в машинах серии ЕС необходима оперативная память около 300 К, а для И.п.п.п. Пролог-АДАБАС — уже около 800 К. Весьма существенным фактором — являются некоторые ограничения на функциональные возможности по сравнению с лучшими образцами в своём классе.

Наибольшую популярность среди И.п.п.п. в классе программных средств на персональных электронных вычислительных машинах имеют Lotus 1-2-3, Symphony, Framework. В классе программных средств, используемых на ЕС ЭВМ, известны такие пакеты, как ЗАПСИБ-Пролог, Пролог-АДАБАС, АОС-ВУЗ-ПРОЛОГ и др.

А.П.Ильяшенко, М.Е.Козлов.

ИНТЕРАКТИВНАЯ МАШИННАЯ ГРАФИКА — вид машинной графики, с помощью которой пользователь может в реальном масштабе времени динамически управлять содержанием изображения, его формой, размерами и цветом на экране дисплея. Для компьютерной технологии обучения представляет особый интерес. В частности, среди средств машинной графики автоматизированных обучающих систем (АОС) И.м.г. занимает центральное место. По существу, И.м.г. АОС — новая форма взаимодействия обучаемого, преподавателя, автора автоматизированного учебного курса (АУК) с компьютером, обслуживающим учебный процесс, и собственно с программным обеспечением АОС.

И.м.г. АОС позволяет объединить лучшие черты интерактивной текстовой связи (посредством клавиатуры терминала) и графического взаимодействия (посредством двухмерного рисунка) с учебным программным обеспечением и инструментальными средствами АОС. Это дает возможность существенно активизировать познавательную деятельность обучаемых, упростить взаимодействие преподавателя и автора АУК с АОС. Кроме статистических изображений, И.м.г. АОС поддерживает динамически изменяющиеся изображения. Это позволяет представлять в графическом виде изменяющиеся во времени изучаемые явления (как реальные, так и абстрактные). Динамика движений обеспечивает визуализацию перемещений объектов относительно наблюдателя, их вращение. И наоборот, фиксируя объект, средства И.м.г. допускают панорамирование объекта, увеличение и уменьшение степени его детализации и т.д. С помощью И.м.г. АОС можно также передавать динамику изменений учебного объекта, показывая фактическую эволюцию его формы, цвета и др. свойств.

В.И.Отенко.

ИНТЕРАКТИВНЫЕ МЕТОДЫ ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЯ — совокупность *процедур*, обеспечивающих формирование изображения на экране, его модификацию при *диал. зовой работе* в графическом режиме. Формирование изображения на экране *дисплея* представляет собой процесс создания (или модификации) *модели* объекта, показываемого на экране. Базисные операции в этом процессе — определение точек и отрезков, а также перемещение, вращение и масштабирование объектов. Манипулирование объектом осуществляется путем косвенного манипулирования его изображением на экране.

Для рисования отрезков чаще всего используется метод резиновой нити, суть которого заключается в том, что при входе в режим резиновой нити фиксируются координаты текущего положения экранного курсора как координаты начальной точки и затем рисуется отрезок, соединяющий начальную точку с текущей позицией экранного курсора. При перемещении экранного курсора движется и конечная точка отрезка. Выход из режима резиновой нити приводит к фиксации координат конечной точки отрезка, каковыми являются координаты точки, в которой в момент выхода находится экранный курсор.

Одним из наиболее распространенных методов перемещения объектов является буксировка, представляющая собой динамическое перемещение выбранного *символа* из одной позиции в другую под контролем локатора. Для начала буксировки обычно нажимается одна клавиша. При этом происходит “захват” символа, на который указывал экранный курсор. Перемещение экранного курсора приводит к перемещению выбранного символа на экране дисплея. Завершение буксировки и “освобождение” символа от курсора происходит при нажатии другой клавиши. Буксировка дополняется динамическим масштабированием и вращением объектов для установления желаемых размеров и ориентации. Для облегчения ориентации некоторых сложных трехмерных объектов используется специальный указатель — *гномон*, показывающий ориентацию главных осей. Гномон упрощает вращение объектов в трехмерном пространстве. При формировании рисунков часто желательно начать рисовать новый отрезок в конечной точке существующего отрезка или на нем. Позиционирование экранного курсора с такой точностью может быть затруднительным. Эту трудность преодолевают, используя метод, наз. *гравитационным полем*. Суть его заключается в том, что отрезок окружается окрестностью, попадая в которую, экранный курсор устанавливается в точности на одну из точек, принадлежащих отрезку.

Основа эффективности И.м.ф.и. — наличие немедленной динамической обратной связи. Эта связь может быть обеспечена только гарантированной доступностью услуг компьютера. Персональные компьютеры позволяют осуществить динамическую обратную связь

с достаточной степенью эффективности. Если же используется компьютер, работающий в *режиме разделения времени*, то организация динамической обратной связи весьма усложняется.

В.И.Отенко.

ИНТЕРПРЕТАТОР (от лат. interpretatio — истолкование, перевод) — *языковой процессор*, реализующий операторы программы, записанной на некотором языке программирования, путем перевода с этого языка на внутренний язык ЭВМ (интерпретации) и исполнения каждого оператора (в отличие от компилятора, переводящего тексты программ с одного языка программирования на другой). Может реализовываться как программными, так и аппаратными средствами. Как правило, время выполнения программ на И. больше, чем на компиляторе, однако И. удобнее компиляторов при написании небольших программ и при отладке программ. Благодаря этому при изучении программирования используются в основном языки интерпретирующего типа (*Бейсик, LOGO* и др.).

Многие проблемно-ориентированные языки и прикладные системы реал ованы в виде И. или содержат И. (напр., инструментальные системы программирования автоматизированных учебных курсов СПОК, СПОК-ВУЗ, АОС-ВУЗ; системы управления базами данных dBASE и FoxBASE). Некоторые универсальные языки программирования высокого уровня реализованы в виде И. (APL, Forth), для других языков созданы как интерпретаторы, так и компиляторы (Бейсик). В некоторых программных системах (например, в инструментальной системе программирования автоматизированных учебных курсов АОС-М) и языках программирования (Смолток) компиляторы и интерпретаторы используются одновременно и выполняют различные функции.

В.А.Бардадым.

ИНТЕРФЕЙС (англ. interface) — совокупность технических и программных средств, обеспечивающих логическое или физическое взаимодействие между устройствами и/или программами в *электронной вычислительной машине*. Физический И. определяет тип стыка, уровни сигналов, импеданс, синхронизацию и другие параметры канала связи, программный И. — совокупность допустимых процедур или операций и их параметров, список общих переменных, областей памяти ЭВМ и др. Внутренний (внутрисистемный) И. осуществляет связь между отдельными компонентами ЭВМ в синхронном и асинхронном режимах. В ЭВМ с синхронным внутрисистемным И. тактирование всех блоков и узлов производится единой тактовой частотой. В асинхронных — каждое устройство работает в своем темпе и взаимодействует с другими устройствами в режиме диалога. В этом случае передающее устройство посылает

сигнал и удерживает его до тех пор, пока не получит ответ от принимающего устройства о том, что сигнал принят, после чего посылает следующий сигнал и т.д. Аналогично внутри одного устройства И. *периферийных устройств* (внешний И.). Как внутренний так и внешний И. бывают последовательными и параллельными. Последовательные И. производят передачу и приём *данных* последовательно *бит* за битом. Параллельные И. осуществляют передачу или прием данных *байтами* или словами. Они обладают большим быстродействием, но в отличие от последовательных И., где одна линия связи, для них необходимо много линий связи (как минимум на одну больше, чем количество бит в передаваемом слове; эта дополнительная линия связи служит для передачи синхросигналов).

В.Л. Леонтьев, В.И. Семихов.

ИНТЕР-ЭКСПЕРТ — инструментальная система, предназначенная для создания прикладных программ и систем, в которых наравне с обычными вычислительными алгоритмами могут применяться методы экспертной оценки и вывода. *Экспертные системы* (ЭС), разработанные в среде И.-Э. основываются на продукционных базах знаний (см. *Продукционные методы представления знаний*). Система И.-Э. создана советско-болгарским научно-исследовательским институтом "ИНТЕРПРОГРАММА".

И.-Э. обладает широкими функциональными возможностями благодаря наличию языка команд высокого уровня (см. *Уровень алгоритмического языка*); встроенного *текстового редактора*; средств интерактивного конструирования *баз данных* (БД) и *баз знаний* (БЗ); средств логического вывода в среде продукций; подсистемы управления цветовым отображением; подсистемы построения диаграмм и графиков. Язык команд системы обеспечивает широкий набор действий и позволяет выполнить *операции*, типичные для языков высокого уровня: присвоение значения (let); цикл (while... do... endwhile); оценка истинности условия (if...then...else...endif); множественное ветвление процесса вычислений (test... case'... case p... endtest); ввод и вывод *данных* на экран *дисплея* (input, output). Имеется ряд других команд и встроенных функций, исполняющих специальные действия. Есть возможность определять *макрокоманды* и использовать *процедуры*, как с параметрами, так и без них. И.-Э. позволяет строить последовательность команд (в виде командного *текстового файла*) и выполнять их как программу, а также обеспечивает возможность интерактивного покомандного исполнения действий, указанных *пользователем*. Допускаются следующие типы значений переменных: числовые (действительные и целые); символьные; логические. Можно описывать и использовать переменные неизвестного типа (unknown). Текстовый редактор И.-Э. позволяет вводить и корректировать различные виды документов, наборы

правил, процедуры, описания форм и *макроопределения*. Построение и применение БЗ по некоторой *предметной области* и БД, содержащих фактический материал, является одной из осн. задач, для решения которых и была создана система.

Знания в И.-Э. представляются в виде правил продукций, имеющих следующую структуру: ЕСЛИ справедливо утверждение Р, ТО выполнить действия A_1, \dots, A_N . Последовательность, в которой проверяются условия и планируются действия, может быть произвольной. Следовательно, знания специалиста-эксперта отображаются в виде множества правил продукций, выполняющихся в некоторой последовательности в зависимости от данных, характеризующих конкретную задачу, и от *стратегии* выбора очередного правила. Существенным удобством для пользователя, обращающегося к ЭС, разработанной в И.-Э., является возможность с основания, предусмотренная форматом правил продукций. Каждое правило имеет специальное поле ПРИЧИНА, в котором приведено содержательное и понятное для пользователя описание причин выбора и последствий его применения. Эта информация не используется системой при обработке БЗ, но когда последовательность правил построена, выполнена и привела к некоторому результату, то пользователь, желающий убедиться в обоснованности результата, может выдать *запрос*, в ответ на который ему будет представлена последовательность сообщений из поля ПРИЧИНА, объясняющих действия системы.

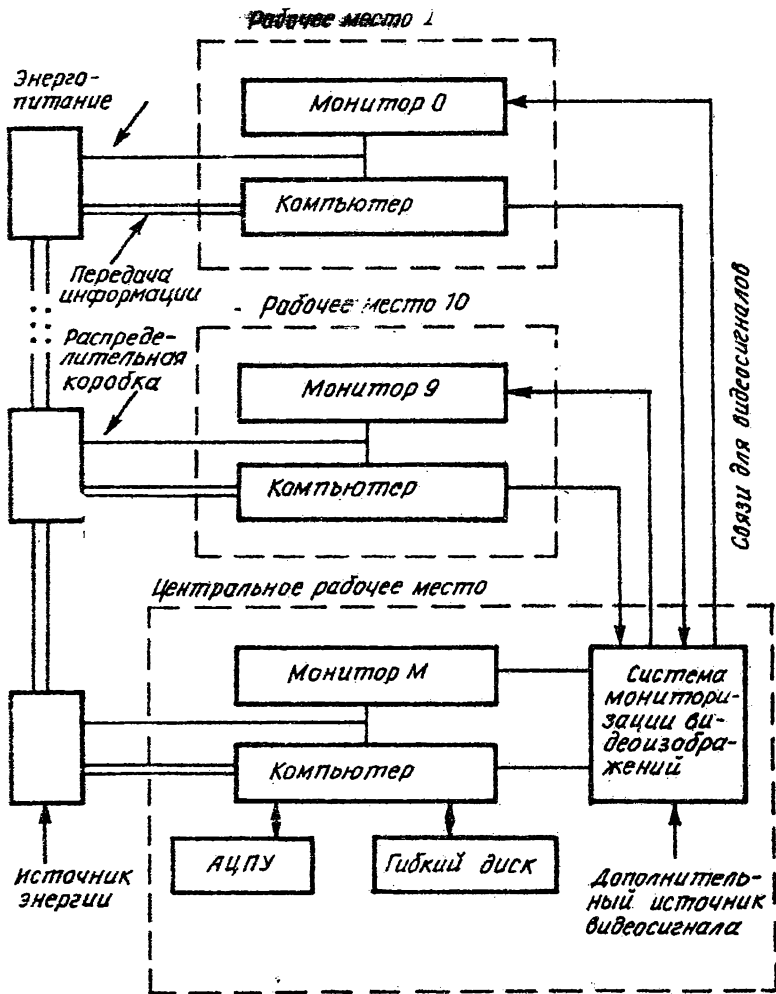
В процессе логического вывода при обработке ситуаций с элементами неполной определенности используются факторы уверенности. При отображении знаний специалиста-эксперта в набор правил часто приходится сталкиваться с неточной (нечеткой) информацией, т.е. с неоднозначностью значений или неоднозначностью выбора и применения того или иного правила. Факторы уверенности используются для определения степени достоверности данных и знаний. В процессе работы ЭС уверенность в некоторых сведениях может возрастать или уменьшаться, что учитывается при получении результата. Следовательно, применение факторов уверенности позволяет использовать не вполне определенные знания или данные и оценивать достоверность выводов, сделанных экспертной системой. И.-Э. функционирует в среде *операционной системы* MS DOS на персональных компьютерах с оперативной памятью ЭВМ не менее 640 К при наличии *винчестерского накопителя*.

М.В.Лёгкий, Н.А.Петрова.

INFOBUS — передвижной класс — лаборатория для *автоматизированного обучения* информатике. Спроектирована и разработана в Румынии в 1988. Смонтирована в автобусе с целью немедленного удовлетворения потребности в практическом обучении в различных организациях, предприятиях, школах и т.д. В этой

лаборатории оригинально решены и обеспечены условия, необходимые для выполнения современного процесса обучения на основе средств вычислительной техники, средств для звукоизображения видеопленки, средств для звукоизображения, информационных программ обучающего и профессионального характера. INFOBUS позволяет использовать учебную технологию, в рамках которой информационные средства применяются для обеспечения эффективной связи между индивидуальным и групповым (в котором непосредственно участвует преподаватель) обучением.

В состав комплекса INFOBUS входят следующие подсистемы: транспортное средство, в котором смонтирована лаборатория; вычислительные устройства; библиотека информационных программ; подсистема для приема и передачи видеоизображений; подсистема для передачи звука; подсистема климатизации; подсистема электропитания. Эти подсистемы спроектированы так, чтобы обеспечить единообразный характер для комплекса. Каждый компонент играет определенную роль в процессе обучения. INFOBUS является в высшей степени автономным и обеспечивает оптимальные условия для процесса обучения, основанного как на информатизационных учебных средствах (компьютерах, программах), так и на классических учебных средствах (видеоизображения, звуки и т.д.). Подсистема транспортного средства состоит из автобуса типа 111-RDT (выпускает бухарестское предприятие "AUTOBUZUL"), имеющего следующие технические характеристики: длина — 11 м, ширина — 2,5 м, высота — 3 м, мотор в 192 лошадиные силы (2100 вращений в 1 мин). Внутренняя площадь автобуса разделена на 3 части: доступная площадь, площадь для лаборатории с поверхностью 3 x 2.4 м и утилитарная площадь для складирования устройств во время транспортировки. Автобус используется в двух рабочих режимах: в качестве лаборатории по информатике; для этого устроены 10+1 рабочих мест для 20+1 пользователей; как транспортное средство для 29 человек. Подсистема вычислительных устройств состоит из 11 румынских персональных компьютеров HC-85 (совместимых с ZxSpectrum и CP/M), взаимосвязанных в рамках специализированной локальной сети (рис.). Сравнительно невысокая стоимость устройств, соединенных в сеть, обеспечивает удовлетворяющее целям лаборатории соответствие между стоимостью и техническими качествами. Эти компьютеры смонтированы как рабочие места (центральное рабочее место предназначено для преподавателя и 10 рабочих мест используются обучаемыми; каждый компьютер предназначен для работы двух обучаемых). Стандартное оснащение лаборатории содержит вычислительные устройства: персональный компьютер типа HC-85 с гибким диском, алфавитно-цифровое печатающее устройство (АЦПУ), цветной монитор и интерфейс связи с центральным рабочим местом; 10 персональных компьютеров типа HC-85 с черно-белым монитором и интерфейсом связи с другими рабочими местами. Библиотека информационных прог-



рамм расположена на дисковом центре центрального рабочего места и доступна с каждого рабочего места командам BASIC или CP/M в зависимости от селектированного рабочего места. Оригинальный способ проектирования позволяет использовать в лаборатории и другие виды вычислительных устройств (профессиональные микро-ЭВМ или мини-ЭВМ с терминалами, минироботы и др.). Вычислительные устройства, которые можно смонтировать, были

представлены в рамках концепции “автоматизированного учебного модуля”, предложенного в качестве типового решения по оснащению учебных заведений вычислительной техникой. Такой модуль является интегрированным набором технических и программных средств, удовлетворяющим требованиям учебной технологии, основанной на информационных средствах. Подсистема для приёма и передачи видеоизображений основана на электронной системе мониторинга видеоизображений (VMS — Video Monitoring System), специально спроектированной для комплекса INFOBUS и смонтированной для работы центрального рабочего места. VMS принимает видеоизображение от одного источника и передает его на мониторы, находящиеся на рабочих местах. Такое изображение может исходить от любого из 11 компьютеров, используемых для рабочих мест, или от внешнего дополнительного источника (напр., от телевизионной камеры, видеокассетфона, приемника телевизионных изображений, принимаемых от спутника или от наземных станций). VMS обеспечивает рабочие режимы: локальный — на каждом мониторе появляется изображение, генерированное с помощью подсоединенного к нему рабочего компьютера; дистанционный — на мониторе появляется изображение, исходящее от дополнительного источника или от любого из остальных 10 рабочих мест; в крайнем случае изображение-источник можно дублировать на всех мониторах лаборатории; автоматическое инспектирование — на мониторе центрального рабочего места кругообразно дублируется (по 20 с) изображение остальных 10 рабочих мест; селекционное инспектирование — на мониторе центрального рабочего места дублируется изображение, имеющееся на мониторе селекционированного рабочего места. Эта подсистема дает возможность реализовать учебный процесс с помощью компьютера, визуально контролировать, как работает каждый ученик на своем рабочем месте, производить демонстрацию пакетов программ и т.д. С помощью подсистемы передачи звука обеспечивается принятие звука от источника (микрофона, магнитофона), его распространение и передача в рамках лаборатории. Подсистема климатизации обеспечивает сохранение нормальных условий работы летом и зимой. Кроме нагревательной и вентилирующей установок, существующих в стандартном оснащении автобуса, подсистема климатизации содержит два кондиционера для летней погоды и два электрических калорифера для зимней погоды. С помощью подсистемы электропитания обеспечивается необходимая для функционирования лаборатории электроэнергия. Электропитание производится от внешнего трехфазного источника 220 Вт с частотой 50/60 Гц. Для связи с внешним источником используется гибкий кабель длиной 50 м. Распределительная панель подает энергию на каждое рабочее место.

Педагогический эффект комплекса INFOBUS определяется возможностью обмена информацией (программами или видеоизобра-

жениями) между рабочими местами и, особенно, между центральным и остальными рабочими местами; простотой управления библиотеками обучающих и профессиональных программ, имеющихся на диске центрального рабочего места; глобальной асинхронизацией между операциями обработки, производимыми на уровне рабочих мест; сочетанием индивидуального рабочего метода с коллективным (групповым) рабочим методом, что возможно благодаря созданию сети компьютеров и наличию подсистемы мониторинга видеозаписей (VMS); приспособлением к разным обучающим моделям путем использования библиотек, имеющих специфичные обучающие программы для взаимодействия/ общения между обучаемым и машиной, двумя обучаемыми, обучаемым и преподавателем; созданием нового профессионального стиля работы благодаря доступу к библиотекам профессиональных программ, находящихся на диске центрального рабочего места. INFOBUS применяется для преподавания информатики и других школьных дисциплин с помощью компьютеров; практических работ по программированию, проводимых индивидуально или под контролем преподавателя; практических демонстраций с помощью информационных технологий; онлайн-проверки знаний; динамического контроля развития процесса обучения; сбора и обработки статистической информации о процессе обучения; организации лагерей и конкурсов по информатике и др.

Н.Бадеа-Динке, М.Житару, Р.Тудорике.

ИНФОРМАТИКА — наука, изучающая вопросы, связанные с поиском, сбором, хранением, преобразованием и использованием информации, механизм воздействия информации на человеческую деятельность и усиление этого воздействия с помощью вычислительной техники. Современная И. — результат бурного развития науки и техники во 2-й половине XX в. Но можно сказать, что возникла она одновременно с первыми попытками автоматизировать умственную деятельность человека. Первые эффективные применения И. связаны с использованием *электронных вычислительных машин* для расчетов в области физики. С развитием компьютеров И. стала использоваться в экономике и управлении производством с целью поиска оптимальных решений. Ныне И. проникает и в науки, еще до недавнего времени считавшиеся не формализованными, т.е. не доступными для точных количественных методов исследований.

Иногда И. отождествляют с теорией информации. Особенно широкое распространение получила трактовка И. как науки, изучающей структуру и общие свойства научной информации. Распространено также толкование И. как науки о вычислительном процессе, или науки об ЭВМ ("Computer Science"). Обычно "Computer Science" отождествляют либо с кибернетикой, либо с вычислительной технологией (включая сюда все её элементы: компьютеры, носители

информации, матем. обеспечение, организацию вычислений, кадровое обеспечение, *интерфейс*), трактуэт как дисциплину по чисто машинной обработке информации, целью которой является ответ на вопрос, как должны выглядеть системы *искусственного интеллекта*. Узость такого подхода очевидна. И. — не вычислительная технология (технология обработки *данных*) в “чистом виде”, а вычислительная технология, ставшая элементом конкретной социальной среды, органически “встроенная” в эту среду и преобразовывающая её как в соответствии с требованиями теории информации и вычислительной техники, так и с потребностями развития самой среды (производственной, управленческой, научной и т.д.). Комплекс воздействия вычислительной технологии на среду применения — главный элемент И., её “ядро”. Именно поэтому существует И. как особая научная дисциплина наряду с “Computer Science”.

Предметом И. как науки выступают специальные машинно-информационные системы (управленческие, научные, производственные, учебные, медицинские и др.) в их взаимодействии со средой. Эти системы являются воплощением новой технологии сбора, переработки, передачи информации — технологии, переводящей практику управления, регулирования материального производства, научных исследований, образования, медицины и др. областей человеческой деятельности на принципиально новый, индустриальный уровень. Предметом И. является также вычислительная технология как социально-исторический (производственный, управленческий, познавательный) феномен. Это пограничная область, охватывающая как вычислительную технологию, так и конкретную (исторически возникшую) социальную информационно-коммуникативную практику.

Проблемы “встраивания” компьютерной технологии в общественную практику, глубины и широты её воздействия на информационно-коммуникативные процессы в обществе тем сложнее, чем выше поколения электронной техники и шире масштабы её распространения. Поэтому значимость И., её социальных функций растет по мере перехода к новым поколениям компьютеров.

И. применяется во всех важнейших подсистемах общества: материальном производстве, планировании, управлении, административно-хозяйственной сфере, науке, образовании (см. *Учебная информатика*), проектном деле, обслуживании, обороне, медицине, охране окружающей среды, криминалистике и т.д. Все шире будет применяться она непосредственно в индивидуальной работе, повседневной жизни людей, что связано с использованием персональных и домашних компьютеров, соответствующих средств *программного обеспечения*, упрощения *машинно-ориентированных языков*. И. обеспечила переход от разрозненного использования новых методов и инструментов планового управления (применение матем. методов, моделирование, машинный эксперимент, оптимизация, автоматизи-

зированной обработка данных и др.) к целостной, принципиально новой технологии информационно-организационных процессов.

В.Я.Валах, В.И.Гриценко.

ИНФОРМАЦИОННАЯ КУЛЬТУРА — составная часть содержания общего и специального образования. Подход к определению понятия И.к. в *обучении* у различных ученых разный. Напр., у А. П.Ершова и В.М.Монахова он во многом сводится к целям и задачам введения в учебные планы курса “Основы информатики и вычислительной техники”, содержание которого включает: умение грамотно поставить *задачи*, возникающие в практической деятельности, для их решения с помощью компьютера; навыки формализованного описания поставленных задач, элементарные знания о методах матем. моделирования и умение строить простые матем. модели поставленных задач; знание осн. алгоритмических структур и умение применять эти знания для построения *алгоритмов* решения задач по их матем. моделям; понимание устройства и функционирования компьютеров и элементарные навыки составления *программ* для компьютеров по построенному алгоритму на одном из *языков программирования* высокого уровня (см. *Уровень алгоритмического языка*); навыки квалифицированного использования осн. типов *информационных систем* (и *пакетов прикладных программ*) общего назначения для решения с их помощью *познавательных задач* и понимания осн. принципов, лежащих в основе функционирования этих систем; умение грамотно интерпретировать результаты решения несложных практических задач с помощью компьютера и применять эти результаты в практической деятельности. Ряд исследователей включает в содержание И.к. либо только овладение учебным материалом курса “Основы информатики и вычислительной техники”, либо дополнительно некоторыми сведениями из школьных программ по математике, связанных с *информатикой* (понятие об алгоритмах, типология алгоритмов и др.). Несколько иной подход к структуре И.к. у В.Кайвина и Н.Устинович, которые считают, что структура И.к. должна включать не только умение работать с компьютером, но и общение учащихся друг с другом. Однако содержание И.к. и здесь рассматривается несколько сужено, охватывая фактически не И.к., а компьютерную культуру. В понятие И.к. целесообразно включить также комплекс общеучебных умений, т. е. таких умений, которые не привязаны к конкретному предметному содержанию, а формируют умение учиться. Исходя из этого, в понятие И.к. должны входить понятия общеучебной культуры, культуры логического мышления и умственных действий, культуры общения друг с другом и компьютерной культуры.

Общеучебная культура — комплекс общеучебных знаний и умений, внедрение которых приводит к успешному усвоению И.к.

Этот комплекс входят умения: читать и сканировать *текст* с разной скоростью; составлять простой и сложный планы текста или собственного изложения; реферировать и конспектировать учебные и научно-популярные тексты; составлять аннотации и резюме, отбирать *ключевые слова* и дескрипторы (единицы языка, отвечающие определенному понятию) текста; применять информационную технику.

Культура логического мышления и умственных действий. Соответствующий комплекс включает следующие умения: а) планирование структуры действий, необходимых для достижения заданной цели при помощи фиксированного набора информационных средств; б) составление информационной структуры для описания объектов и систем; в) организация поиска *информации*, необходимой для решения поставленных задач; г) правильная, четкая и однозначная формулировка мысли в понятной для собеседника форме и правильное понимание текстов информационных сообщений.

Культура общения друг с другом. Закономерности *взаимодействия человеко-машинного* в большей или меньшей степени опираются на обобщение опыта человеческих коммуникаций, в частности *диалога*. Культура общения, её теор. аспекты важны и для обоснования человеко-машинного взаимодействия. В этой связи важны умения выслушать чужую точку зрения, адекватно к ней относиться, представлять информацию в любой форме; уметь излагать свою точку зрения и доказывать свою правоту; находить общие решения и составлять программы совместной деятельности для достижения общих целей.

Компьютерная культура. Учащиеся, овладевшие ею, в своей профессиональной деятельности постоянно используют компьютер; за годы обучения накапливается достаточный опыт такой работы. Общеизвестно, что овладение основами *компьютерной грамотности* будет со временем перенесено на более ранние, чем ныне, сроки — к 6—8-м классам образовательной школы, а также в начальную школу. Целесообразно так располагать в процессе усвоения “некомпьютерные” составляющие И. к., чтобы те из них, которые должны обслуживать школьную информатику, формировать компьютерную грамотность, соответственно изучались раньше, чем по ныне действующим учебным планам.

И.к. относится к системным, многоаспектным понятиям, не ограничивающимся рамками информатики, *дидактики* и педагогической психологии. Иногда наряду с термином “И.к.” используют близкий по смыслу термин “информационный режим”, причем предполагается, что этот режим включает создание мотивов для восприятия информации, формирование приемов *адаптации* к оптимальному темпу её восприятия, способствует пониманию и усвоению знаний и навыков, относящихся к И.к. Грубые отклонения от разумного информационного режима неизбежно приводят к росту

вероятности информационных болезней, которые могут быть следствием как информационного голода, так и информационных перегрузок. Информационные болезни представляют значительную опасность в условиях ведения учебно-воспитательного процесса. Здесь, как и в случае медицинской и технической диагностики, своевременные профилактические меры более целесообразны, чем соответствующее лечение. Предстоит изучить и сформулировать оптимальные методики в условиях новой информационной технологии, обосновать методику и организационные формы преемственного, "сквозного" овладения содержанием И.к., распределить учебный материал, составляющий содержание И.к., между возрастными уровнями (содержанием отдельных предметов). Нужны координированные усилия преподавателей, педагогов-исследователей, чтобы реализовать задачу формирования компьютерной культуры, И.к. в целом.

Н.М. Розенберг.

ИНФОРМАЦИОННАЯ СИСТЕМА — система, обеспечивающая предоставление информации для принятия решений и других применений. Включает языково-алгоритмические и технические средства для хранения, поиска и выдачи необходимой информации по запросам конечных пользователей. До настоящего времени развитие И.с. определялось развитием технического обеспечения (компьютеры, периферийное оборудование, средства передачи данных и т.п.), программного обеспечения и информационного обеспечения. Техническое обеспечение играло ведущую роль в развитии И.с. За короткий срок сменилось три поколения компьютеров. Смена поколений технических средств вызвала изменения в программном обеспечении и частично в средствах информационного обеспечения. Именно со сменой поколений компьютеров связывают и этапы развития И.с. Первый этап характеризуется применением компьютеров первого поколения для реализации отдельных расчетов по индивидуальным программам, составленным в кодах машины. Распространение компьютеров второго поколения привело к смене концепций в развитии И.с. Для неё характерны выделение алгоритмических процессов обработки данных и обособленное их выполнение в специальных службах (вычислительных центрах). В области программного обеспечения усилия по снижению трудоёмкости программирования обусловили создание операционных систем, процедурных языков высокого уровня (см. Уровень алгоритмического языка) и трансляторов к ним. Для третьего этапа развития И.с. наиболее характерным моментом является создание систем управления базами данных, обеспечившее независимость процессов ведения массивов информации от прикладных программ. Они стали важнейшей предпосылкой повышения гибкости И.с. —

...явилась возможность организовать работу И.с. в режиме "запрос-ответ" (*информационно-справочный режим*).

Ныне отсутствует четкое разделение И.с. на информационно-поисковые, информационно-справочные, информационно-логические и др. Выделяют две различные концепции построения И.с.: первую обозначают термином "*информационно-поисковые системы (ИПС)*", вторую — "*банк данных*". ИПС с самого начала ограничивались функциями хранения и поиска информации. Они обычно ориентируются на определенную *предметную область*, чаще всего область научно-технической информации. При построении ИПС большое внимание уделяется анализу информации, а их технология строится в явной связи с ее содержанием. ИПС появились на ранних этапах развития И.с. Большинство из них реализовано на компьютерах второго поколения. Концепция "*банк данных*" сформировалась на третьем этапе развития И.с., отразив изменившуюся роль процессов хранения и поиска данных. Под банком данных обычно понимают И.с., выполняющую функции хранения данных, поступающих из многих источников, а также поиска и выдачи их для многих *пользователей*. Банк данных предполагает наличие одной или нескольких *баз данных* и СУБД, осуществляющей управление ими. И.с., строящиеся на базе инструментальных пакетов *ЗАПСИБ-ПРОЛОГ* и *ЛИПС*, рассматриваются в рамках концепции "*банк данных*".

Н.А.Власенко.

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ информационных систем — система *представления знаний*, содержащая элементарные сведения об основных параметрах *предметной области*. Это могут быть цифровые или иные *данные*, характеризующие сроки, объемы, транспортные средства и направления перевозок некоторых видов сырья или промышленной продукции. В структуре инструментального пакета *ЛИПС* роль И.о. играет *интеллектуальный банк данных*.

Н.А.Власенко.

ИНФОРМАЦИОННО-ОПЕРАЦИОННАЯ ЛОГИЧЕСКАЯ СЕТЬ — двудольный ориентированный граф, отображающий логические связи между информационными (И) и операционными (О) элементами системы. I-элементы обозначают *фрагменты знаний*, а 0-элементы — *операции*. И.-о.л.с. строят для различных видов *обучения* (изучение, упражнения, решение проблемы, тестирование, моделирование); их можно интерпретировать с двух позиций — формирования знаний и организации операций. Напр., при описании процесса изучения 0-элементы представляют собой мыслительные действия, которыми *обучаемый* овладевает в процессе обучения. Входные I-элементы являются исходными знаниями (*образы, понятия, утверждения*), необходимыми для выполнения определенных умственных операций,

выходные I-элементы — результатом этих действий. 0-элементы являются учебными действиями, I-элементы — разделами учебного материала.

Ю.И.Лобанов.

ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА (ИПС) — *информационная система, обеспечивающая автоматический поиск информации. Построение ИПС связано с решением ряда семантических проблем и прежде всего с разработкой представления знаний о предметной области (см. Модель предметной области), что весьма трудно осуществить, используя традиционные модели баз данных: иерархическую, сетевую и реляционную. Развивается информационно-логический подход к моделированию баз данных (БД), в котором вначале строится концептуальная модель предметной области, а затем происходит отображение в модель БД. Информационно-логический подход связывает проблемы моделирования в БД с проблемами представления знаний. ИПС делятся на документальные и фактографические. Документальные ИПС используют для поиска научно-технических документов, содержащих необходимую информацию (записями этих систем являются различные неформализованные документы — статьи, книги, письма и т.п. Фактографические ИПС применяются для получения ответов на запросы по фактам к.-л. одного вида (БД этих систем состоит из формализованных записей — картотек, каталогов и т.п.).*

С.П.Кудрявцева.

ИНФОРМАЦИОННО-СПРАВОЧНАЯ СИСТЕМА (ИСС) — *информационная система, обеспечивающая регистрацию, переработку и хранение справочной информации. Примерами ИСС могут служить справочные службы (в т.ч. и в учебных центрах), диспетчерские службы, библиографические отделы. ИСС должны иметь средства представления и обработки знаний различного уровня — от манипулирования фрагментами текстов до различных методов представления знаний (фреймовых, сетевых, продукционных). ИСС присутствует в любой автоматизированной обучающей системе как отдельная подсистема (режим взаимодействия обучаемого с системой).*

С.П.Кудрявцева.

ИНФОРМАЦИОННО-СПРАВОЧНЫЙ РЕЖИМ в структуре автоматизированного учебного курса — режим, при котором обучаемый получает информацию справочного характера, не включенную в основное содержание обучения (напр., различные табличные данные (удельные веса, удельные сопротивления различных материалов и др.); расшифровку кодов завершения заданий для операционной

системы компьютера, коды и сообщения *трансляторов* и т.д.). Простейшей реализацией И.-с.р. является встраивание в тело компьютерной *программы* курса заранее определенной необходимой для данного курса справочной информации, которая выводится на экран дисплея по *запросу* обучаемого. Недостатки такой реализации очевидны, т.к. всякие изменения справочной информации (удаление устаревших *данных*, пополнение новыми данными и т.п.) влекут за собой переделку всей программы курса, что не всегда можно осуществить с малыми трудозатратами и достаточно быстро. Эффективность И.-с.р. зависит от того, на сколько они удовлетворяют предъявляемым к ним осн. требованиям — открытости, простоте формирования запроса на поиск информации, времени поиска и визуализации информации. Исходя из свойства открытости, в качестве И.-с.р. можно использовать некоторую типовую *систему управления базами данных* (СУБД), интегрированную с *автоматизированным учебным курсом*. Однако такой подход редко применяется из-за универсальности, присущей большинству СУБД, вследствие чего усложняется *процедура* запроса информации и время ее поиска, что не приемлемо с точки зрения удобства для обучаемого. В то же время структура данных И.-с.р. курса достаточно проста, что позволяет реализовывать в курсе открытые И.-с.р. с простым доступом к информации и малым временем реакции. Основу организации открытых И.-с.р. составляет фиксированная *база знаний*, к которой обучаемые могут обращаться в соответствии с *алгоритмом* обучения курса и, кроме того, по собственному усмотрению.

В.Д.Рынгац.

ИНФОРМАЦИЯ (лат. informatio — разъяснение, осведомление) — сведения, которыми располагает активная система (общество, человек-индивид, компьютер и т.п.) о том или ином объекте. И. об объекте представлена в его разнообразных *моделях*, в т.ч. знаковых. Несущий И. *знак* или система знаков наз. сообщением, совокупность сообщений — *данными* о соответствующих объектах. Одну и ту же И. можно передавать с помощью различных сообщений (напр., на разных языках), и, наоборот, одно и то же сообщение, по-разному интерпретированное, может передавать различную И. Возможности общества в использовании И. существенно обогатились благодаря специальным машинным технологиям ее сбора, переработки и передачи, т.е. методом *информатики*.

В основе матем. теории И. лежит трактовка И. как меры уменьшения неопределенности (энтропии) и способ измерения ее количества, предложенный в 1948 на основе такой трактовки амер. ученым К.Шенноном. За единицу количества И. принимается макс.

И., которую может нести сообщение, состоящее из одного двоичного знака. Такая единица наз. *битом*.

В теории И. рассматриваются составляющие информационных процессов — прием, передача, кодирование, декодирование, запоминание, хранение, извлечение, отображение И. и т.д. Одна из задач теории И. — нахождение наиболее экономичных методов *кодирования*, позволяющих передавать заданную И. посредством миним. количества знаков. Изучаются также требования, которым должны удовлетворять каналы связи, чтобы передавать И. с миним. задержками и искажениями.

И., которую несет к.-л. модель о моделируемой системе, может быть охарактеризована объемом, уровнем адекватности и уровнем полноты, который при прочих равных условиях тем больше, чем больше объем и уровень адекватности рассматриваемой И., и тем меньше, чем сложнее моделируемая система. Частным видом полноты И. можно считать, напр., информативность вторичного документа, выражающую меру адекватного воспроизведения в нем осн. элементов содержательной и формальной структуры первичного документа.

Пусть система B , состоящая из подсистем B_1, B_2, \dots, B_n , есть модель системы A , состоящей из подсистем A_1, A_2, \dots, A_n , причем, каждая подсистема B_i ($i = 1, 2, \dots, n$) есть модель соответствующей подсистемы A_i . В этом случае И., которую система B несет о подсистеме A_i системы A , не исчерпывается той И., которую несет об этой подсистеме соответствующая ей подсистема B_i системы B . Эту последнюю И. можно назвать прямой, а остающуюся часть И., которую система B несет о подсистеме A_i , — косвенной И. о ней. Напр., изображение или описание к.-л. персонажа произведения живописи или литературы несет об этом персонаже прямую И. для зрителя или читателя, а другие компоненты произведения, так или иначе связанные с этим изображением или описанием, — косвенную И. об упомянутом персонаже.

Г.А.Балл, В.Я.Валах.

ИОНИЗИРУЮЩЕЕ ИЗЛУЧЕНИЕ — *излучение*, взаимодействие которого с веществом приводит к ионизации (образованию электроположительных и электроотрицательных ионов) его атомов и молекул. Ионизирующей способностью обладают заряженные и нейтральные частицы (альфа-частицы, протоны, позитроны, нейтроны и др.), а также потоки электромагнитных квантов. Видимый свет и ультрафиолетовое излучение не относятся к И.и. Количественные характеристики И.и.: плотность потока, флюенс (отношение числа частиц, проникающих в элементарную сферу, к малому элементу площади сечения этой сферы), энергетический спектр излучения, мощность экспозиционной дозы (МЭД). В настоящее время, согласно

НРБ-76/87, регламентируются дозовые характеристики источников И.и. с энергией квантов ≥ 5 кэВ. Ионизирующая способность характеризуется удельной ионизацией, т.е. количеством ионов обеих знаков, создаваемых на единицу пути. Заряженные частицы ионизируют атомы вещества в результате непосредственных столкновений. При прохождении нейтральных частиц (нейтронов, фотонов) ионизация происходит за счет вторичных частиц, образованных при взаимодействии первичного потока излучения со средой. Удельная ионизация зависит от плотности среды, энергии и заряда ионизирующей частицы. Осн. виды ионизации: столкновительная (соударения с атомами, ионами, электронами), фотоионизация (передача энергии квантов электромагнитного излучения), поверхностная (в результате взаимодействия излучения с поверхностью твердых тел), полевая (ионизация в сильных электрических полях).

Источниками И.и. в компьютерах являются видеотерминалы с электроннолучевыми трубками (ЭЛТ). Испускаемые катодом ЭЛТ электроны бомбардируют покрытый люминофором экран, вызывая его люминесценцию. Плотность потока и энергия излучения с экрана ЭЛТ зависит от состава люминофора и энергии электронов (величиной приложенного к ЭЛТ напряжения). В некоторых типах современных ЭЛТ мощность дозы излучения вплотную к экрану может достигать значительных величин (> 100 мкР/ч). Однако вследствие малой энергии ($\sim 5 - 20$ кэВ) значения МЭД на *рабочем месте* оператора (25-35 см от экрана) не превышают, как правило, фоновых значений. МЭД И.и. дополнительно можно уменьшить как за счет совершенствования конструкции ЭЛТ (подбор люминофоров, применение дополнительных прозрачных экранов и др.), так и за счет оптимизации места расположения оператора и режима его работы.

М.Н.Коротенко.

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ — 1) Искусственная система, способная выполнять действия, имитирующие интеллектуальную деятельность человека. 2) Часть *информатики*, занимающаяся разработкой методов создания таких систем. Исследования в области И.и. ведутся по четырем направлениям. 1. Разработка теории эвристического поиска, создание *решателей* задач, имитирующих творческие процессы (доказательство теорем, сочинение музыкальных произведений, шахматные программы). Именно это направление привело к появлению термина "искусственный интеллект". 2. Разработка теории и принципов построения интеллектуальных роботов, систем обработки изображений (интеллектуальных сенсоров). 3. Интеллектуализация компьютеров, т.е. создание программно-технических средств, позволяющих конечным *пользователям* использовать компьютер в своих практических целях (см. также *Системы*,

основанные на представлении знаний). 4. Аппаратная поддержка целого ряда программ (внутренняя интеллектуализация компьютера). Это приводит к появлению машин баз данных и машин баз знаний, ПРОЛОГ — машин, аппаратно реализованных лингвистических процессоров, логических информационно-вычислительных систем.

С. П. Кудрявцева.

ИСКУССТВЕННЫЙ ЯЗЫК — специально созданная семиотическая система. В отличие от естественного языка, являющегося универсальной знаковой системой, или естественного языка ограниченного, как подмножества естественного языка, И.я. имеет узкоспециальный характер. Основой построения И.я. служит естественный язык. Напр., И.я. международного общения (эсперанто, идо и др.) создавались на основе лексики, общей для наиболее распространённых европейских языков. Их отличает простота словообразования и грамматики. В последнее время создаётся огромное количество И.я., ориентированных на требования отдельных задач общения с компьютером, напр. для записи алгоритмов и программ; для управления работой вычислительной системы; для представления различных видов фактографической, документальной, графической информации, подлежащей вводу, хранению и обработке в компьютере; для записи запросов на поиск, обработку и выдачу информации. (см. Алгоритмический язык, Язык описания курсов, Язык представления знаний, Язык программирования, Язык проектирования, Язык спецификаций).

Н. А. Власенко.

ИСЧИСЛЕНИЕ ЛОГИЧЕСКОЕ — способ задания множества формул (теорем исчисления) путем указания исходных формул (аксиом) и конечного набора правил вывода, каждое из которых определяет, как строить новые формулы из исходных и уже построенных. Задаётся следующими компонентами: логико-математическим языком, задающим множество формул F ; языком исчисления, на котором записываются схемы формул; набором правил вывода вида $\frac{\Phi}{S}$ (или $\Phi \vdash S$), где Φ — конечное множество схем формул (посылки правила), а S — схема формулы (заключение правила). Правило вывода с пустым множеством посылок наз. схемой аксиом. Язык исчисления должен быть таким, чтобы существовали конструктивные правила конкретизации любой схемы формул S , порождающие примеры схемы S — формулы из F_m . Любая конкретизация схемы аксиом наз. аксиомой. Говорят, что формула F выводима из набора формул F_1, \dots, F_n по правилу вывода $R: \frac{\Phi}{S}$, если существует такая одновременная конкретизация всех схем

формул из множества $\Phi \cup \{S\}$, при которой F превращается в $\{F_1, \dots, F_n\}$, а S — в формулу F . Суть любого И.л. F заключается в конструктивном определении бинарного отношения выводимости \vdash между множествами формул и формулам из F_m . Выводом

формулы F из множества формул M наз. такая конечная последовательность формул F_1, \dots, F_n , в которой F_n совпадает с F , а каждая формула F_i , $i=1, \dots, n-1$, либо является аксиомой, либо принадлежит M , либо получена из некоторого подмножества предшествующих формул по одному из правил вывода исчисления J . Говорят, что формула F выводима из множества посылок, предположений) M в исчислении J и пишут $M \vdash F$, если существует

вывод F из M в J . Формула F наз. теоремой исчисления J , если она выводима в J из пустого множества посылок (т.е. выполнено $\emptyset \vdash F$, что сокращённо записывается в виде $\vdash F$ или даже $\vdash F$, если

ясно, о каком исчислении идет речь). Обычно И.л. используются для описания свойств предметных областей (ПО). При этом тем или иным образом (обычно неконструктивным) определяют понятие истинности формулы F в модели \mathcal{M} этой ПО. В этой связи важнейшими характеристиками исчислений являются непротиворечивость, адекватность и полнота. Исчисление J наз. непротиворечивым, если не всякая формула из F_m выводима в J_k ; адекватным модели \mathcal{M} , если всякая теорема исчисления истинна в \mathcal{M} ; полным для модели m , если всякая формула из F_m , истинная в \mathcal{M} , является теоремой исчисления. Важнейшей задачей, связанной с любым И.л. считается задача поиска вывода теорем; если существует алгоритм построения вывода в исчислении J , то с его помощью можно автоматически доказывать утверждения о ПО, описываемой исчислением J . (см. *Логическое описание предметной области*). Одно из самых популярных в искусственном интеллекте — резолюционное исчисление, позволяющее доказывать теоремы классической логики первого порядка и являющееся основой логического программирования (см. *Пролог*).

В.М. Антимиров.

КАБИНЕТ АВТОМАТИЗИРОВАННОГО ОБУЧЕНИЯ — помещение, оснащенное техническими и программными средствами для организации *автоматизированного обучения*. Главным компонентом технических средств является учебная вычислительная техника, в качестве которой используются комплексы учебно-вычислительной техники, терминальные комплексы достаточно мощной ЭВМ или разумное сочетание обоих видов техники. К техническим средствам К.а.о. относятся также проекционный цветной телевизор или

электронная доска, позволяющие демонстрировать компьютерные слайды и вести занятия по безмеловой технологии; телекамера и видеоманитофон, управляемые от ЭВМ, для организации *программируемого обучения* с использованием видеозаписей.

В.Н.Семихов.

КАРУС — система активизации творческой деятельности, творческий тренинг, основанный на *стратегиях* конструирования — комбинирование (К), аналогизирование (А), реконструирование (Р), универсальной (У), случайных подстановок (С). Разработана докт. психол.наук В.А.Моляко. Стратегии как доминирующей тенденции в интеллектуальном поведении решающего *задачу* субъекта подчинены все его действия — подготовительные (понимание условия), планирующие (формирование замысла) и реализующие (проверка замысла, эксперимент). Реализуется стратегия с помощью конкретных действий, сочетание которых составляет определяющую мыслительную тактику. Определяют стратегию доминирующие, основные, принципиальные для конкретной деятельности действия. Если это действия связанные с поисками аналогов, реализуется стратегия аналогизирования; соответственно речь будет идти о стратегии комбинаторных действий, стратегии реконструирующих действий. Если же существует примерное равенство, гармоническое сочетание названных действий, то эту стратегию можно считать универсальной. При хаотическом, нецеленаправленном поиске, когда субъект не имеет доминирующей тенденции, действия ведутся без плана, стратегия наз. стратегией случайных подстановок.

К. опирается на использование различных научных и технических проблем. Чаще всего используются технические задачи по проектированию кинематических цепей, построенные по принципу “черного ящика” (известны параметры входа и выхода системы, следует сконструировать механизм — содержимое “ящика”). Практическое применение К. предполагает диагностический, обучающий и собственно тренинговый этапы. На диагностическом этапе выясняется, как решает данный субъект творческие задачи, каковы приемы, методы и т.д. На обучающем этапе ему предлагают решить новую серию задач с использованием осн. инструкций, направленных на *обучение* поиску аналогов технических элементов, комбинированию технических элементов, их реконструированию, адекватному применению стратегий. Затем осуществляется тренинг по каждой из стратегий.

Тренинг в усложненных условиях — важная психолого-педагогическая характеристика К. Для его реализации разработан ряд специальных приемов (методов). Метод временных ограничений основывается на том положении, что фактор времени существенно влияет на интеллектуальную деятельность человека.

При лимите времени решение в некоторых случаях существенно упрощается (субъект ограничивается использованием того, что он лучше всего знает), а также существенно деформируется. Метод внезапных запретов состоит в том, что субъекту на том или ином этапе решения запрещается использовать в своих построениях конкретные механизмы и процедуры. Этот методический прием весьма эффективен, поскольку практически исключает штампы, стереотипизацию. Метод новых вариантов заключается в требовании решить задачу по-иному, найти новые варианты решения. Это требование вызывает дополнительную активизацию деятельности субъекта, прямо нацеливает на творчество. Метод скоростного эскизирования — от решающего требуется как можно чаще рисовать все то, что он в данный момент представляет себе мысленно. Метод информационной недостаточности — исходное условие представляется с явным недостатком необходимых для начала решения данных. Метод информационной перенасыщенности основывается на включении в исходное условие задачи заведомо лишней информации, не имеющих существенного значения для решения. Метод абсурда заключается в построении заведомо невыполнимой задачи.

К настоящему времени под руководством В.А.Моляко проведено много исследований, направленных на использование К. для активизации творчества на различном материале, в т.ч. в условиях решения задач с помощью компьютера.

М.Л.Смульсон.

КАЧЕСТВО ИЗОБРАЖЕНИЯ — показатель соответствия изображения на экране монитора эргономическим и гигиеническим требованиям. Выделяют следующие характеристики К.и.: яркость символов (изображения); контрастность *символов*; отсутствие отражений и искажений изображения; частота регенерации; линейные и угловые размеры матрицы знака; межстрочное расстояние; межзнаковое расстояние; количество символов в строке; количество строк на экране и другие. В *дисплеях* используются два типа изображения — позитивное (тёмные символы на светлом фоне) и негативное (светлые символы на темном фоне). В большинстве *персональных электронных вычислительных машин*, имеющих монохромные мониторы, используется негативное изображение.

Позитивное изображение при отсутствии мерцаний и обеспечении его устойчивости имеет ряд преимуществ перед негативным: условия работы *пользователя* с экраном и с исходным документом не требуют переадаптации глаз пользователя; снижается чувствительность к дискомфорту от *блескости*; меньше чувствительность пользователя к *световому потоку*, идущему от окон. По данным Всемирной организации здравоохранения, пользователи видеотерминалов с эк-

равами на электронно-лучевой трубке (ЭЛТ) могут испытывать дискомфорт, вызванный недостаточной яркостью символов, контрастностью, мерцанием изображения, бликами, блескостью. Проблема стабильности изображения является одной из главных. Устойчивость изображения рекомендуется обеспечивать за счет выбора люминофора со значением времени послесвечения, находящимся в пределах от среднего до короткого, обеспечения частоты регенерации для негативного изображения для различных люминофоров от 50 до 60 Гц, а для позитивного — минимум 70 Гц. При негативном изображении обычно нормируется яркость символов, для позитивного — яркость фона. Различны и требования к контрасту между символами и фоном.

Количество растровых линий — один из осн. факторов, определяющих разрешающую способность дисплеев. Чем больше имеется растровых сторон, тем выше разрешающая способность дисплея. Для алфавитно-цифровых дисплеев рекомендуется от 8 до 12 растровых строк на символ. При этом следует применять горизонтальную ориентацию строк раstra. Символы в большинстве дисплеев на базе ЭЛТ строятся на основе *матрицы точек*. В связи с этим первостепенное значение имеет размер точек и расстояние между ними. Близлежащие точки символа должны создавать впечатление непрерывной линии. Соотношение между миним. размером точки и расстоянием между точками должно составлять 0,7. Рекомендации относительно размера символа зависят от расстояния наблюдения. Высота символа может составлять около 20 угловых минут, т.е. около 4мм при расстоянии наблюдения 600мм. Для обеспечения *хорошей читаемости* информации, выводимой на экран, лимитируется межстрочное и межзнаковое расстояние. Вариации строки и столбца на экране — менее 1%. Неустойчивость изображения символа на экране менее 0,7 угловой минуты (0,1мм на 50см). Геом. искажения символов по ширине и высоте не должны превышать 10%.

В.М.Бондаровская.

КВАЗИАЛГОРИТМ (от лат. quasi — как будто и *алгоритм*) — предписание, не являющееся алгоритмом, но близкое к нему по своим характеристикам. Точнее — предписание о выполнении активной системой *процедуры*, обладающей следующими свойствами: 1) состоит из *операций*, эффективных и/или квазиэффективных для системы; 2) при разветвлении процедуры то, какая именно операция осуществляется вслед за данной, с вероятностью, равной или достаточно близкой к единице, определяется выполнением некоторых четко установленных условий. Как и в случае алгоритма, по меньшей мере одна из предусматриваемых К. операций является родовой. Для К. часто можно указать эталонный для него алгоритм, отличающийся

от К. тем, что квазиэффективные операции заменены соответствующими эффективными, а вероятность, упомянутая в п.2, обязательно равна единице.

От общего понятия К. следует отличать понятие о квазиалгоритме решения массовой задачи. Для такого К. эталонный алгоритм является алгоритмом решения указанной задачи. К. решения задач (как и другие модели *способов решения задач*) могут находиться в распоряжении *решателя* в различной форме. Рассматривая решение задач человеком, следует различать такие случаи.

1. Указанная модель представлена вовне в виде развернутого предписания (инструкции), устанавливающего содержание и последовательность подлежащих выполнению операций.
2. Вовне представлена только свернутая модель способа решения задачи, но субъект при этом владеет способом перехода от нее к развернутому предписанию. Так, формулу $(a+b)^2 = a^2 + b^2 + 2ab$ можно считать свернутым представлением алгоритма возведения двучлена в квадрат.
3. Субъект помнит предписание и пооперационно воспроизводит его под контролем сознания.
4. Последовательность операций, предусмотренная предписанием, сформирована на уровне навыка.

При выяснении вопроса о том, является ли некоторое предписание К. (как и того, является ли оно алгоритмом), следует учитывать характеристики не только самого предписания, но и системы, которая должна выполнять его. Пусть, напр., ученик хорошо знает требуемую последовательность операций при перемножении чисел “в столбик”. Но таблицу умножения он как следует не усвоил, так что не обеспечена высокая вероятность безошибочного выполнения указанных операций. Поэтому реализуемое учеником предписание не является для него ни алгоритмом, ни К., несмотря на то, что внешне совпадает с “настоящим” матем. алгоритмом.

Г. А. Балл.

КВАЛИФИКАЦИОННЫЕ ХАРАКТЕРИСТИКИ — нормативные документы Государственного комитета по народному образованию, в которых обусловлены квалификационные требования к подготовке специалистов высшей квалификации. Являются юридическими документами, предусматривающими проведение в жизнь единой государственной политики аттестации качества подготовки специалистов по всем специальностям вузов страны и по отношению к каждому студенту. В К.х. отражены условия взаимной ответственности вуза и предприятия за подготовку и последующее использование специалистов. Следовательно, К.х. — это документы, отражающие нормативно-правовую основу и обуславливающие систему мер, обеспечивающих усиление взаимной заинтересованности и ответственности вузов и предприятий. К.х. предназначены для высших учебных заведений, государственных органов, министерств, ведомств при

прогнозировании потребности планирования подготовки и проведении аттестации специалистов. Они утверждаются совместно со средствами диагностики конечных целей подготовки специалистов.

А.Е.Денисов.

КЛАВИАТУРА (нем. Klaviator, от лат. clavis — ключ) — устройство оперативного ручного ввода данных, представляющее собой комплект расположенных в определенном порядке клавиш. Обеспечивает связь между оператором и электронной вычислительной машиной нажатием клавиши. К. может быть выполнена в виде

Спец	1/	2/	3#	4\$	5%	6^	7&	8*	9(0)	=-	+ <	Стр	
Tab	q	w	e	r	t	y	u	i	o	p	[]	печ	
Утр		а	с	р	ф	г	н	ж	л	:	")	Ввод	
Верх		э	х	с	в	н	м	<	>	?	,	:	Верх	
Альт	Лат	Р/Л	Имп	Пробел								Р/Л	Р/С	Икс

отдельного блока, связанного с ЭВМ кабелем связи, или встроена в блок процессора в некоторых персональных ЭВМ. Наиболее распространены К. герконовые (с механическими контактами в вакууме), конденсаторные (две пластины с изменяемым расстоянием между собой), пленочные (металлизированная эластичная пленка через трафарет из диэлектрика контактирует с контактным полем), сенсорные (использующие электрическую емкость тела человека) (рис.).

В.Н.Семихов.

КЛАВИАТУРНЫЙ ТРЕНАЖЁР — тренажёр для обучения работе с клавиатурой. Пробразом такого типа тренажеров можно считать машину Паска образца 1959 SAKI, которая представляла собой электромеханическую систему, использовавшую при выдаче очередного задания оптимальные стратегии в смысле теории игр, что позволяло значительно сократить время подготовки обучаемого. Ныне в связи с широким распространением ЭВМ, в частности персональных электронных вычислительных машин, создано большое количество К.т. на их базе. Обычно они представляют собой адаптивные тренирующие программы, предназначенные для выработки и развития навыков работы с клавиатурами различных типов у операторов и других пользователей ЭВМ.

С.Т.Адрианов.

КЛЮЧЕВОЕ СЛОВО — слово или словосочетание, обозначающее важные с точки зрения смыслового содержания элементы объектно — понятийной структуры *предметной области*. При анализе естественных языковых запросов в *информационных системах* К.с. считаются названия и значения атрибутов *базы данных* (БД) в различных грамматических формах (или их синонимы) для нечисловых значений атрибутов. Пример естественного языкового запроса к системе “МОНОГРАФИЯ” (см. ЗАПСИБ-ПРОЛОГ):

НАЗОВИТЕ МОНОГРАФИИ БЕЛЯЕВОЙ.

К.с. в этом запросе: МОНОГРАФИИ (название атрибута в БД), БЕЛЯЕВОЙ (значение атрибута БД).

Н.А.Власенко.

СОАСН (англ. Constructed Arithmetic and Controlled Help конструирующее арифметическое и контролирующее средство) — *генерирующая автоматизированная обучающая система*, предназначенная для генерации простых арифметических задач. Оценивает *ответы* обучаемого и оказывает при необходимости помощь, *учитывая* при этом *предысторию обучаемого*; определяет тип генерируемой задачи и осуществляет выбор *требуемой стратегии* обучения. Арифметика была выбрана как хорошо структурированная область, с легко определяемыми оценками *уровней сложности задач*.

При *обучении* с помощью С. обучаемый вначале тестируется для выбора необходимого обучающего курса, затем определяет уровень сложности для каждого курса цифрами от 1 до 8 (в системе 8 осн. уровней сложности). После этого ему предоставляется ряд задач. Если обучаемый, напр., выбрал все четыре типа задач первого уровня, он решает сгенерированную последовательность типовых задач этого уровня. После каждого ответа обучаемого выдается сообщение о том, насколько правильно его решение. При *неправильном* ответе обучаемый получает несколько подсказок (включая первоначально простейшую). Предложенная ранее задача предоставляется вновь, и если данный ответ снова неверен, обучаемый в виде подсказки получит правильный ответ. Пока данный тип задач не будет освоен, программа сохраняет представленную задачу. Обучаемый может прервать работу с программой, запомнить *требуемую информацию* на диске и в дальнейшем начать работу именно с этой точки. Кроме того, обучаемый может в процессе обучения изменить набор тем или уровень сложности путем изменения параметров, определенных для каждого проблемного типа. Запись *диалога* между обучаемым и программой фиксируется на протяжении всей работы.

Стратегия обучения формируется на основе анализа текущего усвоения *знаний* обучаемых путем автоматического изменения уровня сложности генерируемой задачи. Задачи внутри одного уровня

разбиты на ряд рангов. Правильное решение двух сгенерированных задач определенного ранга обуславливает постепенное усложнение задач, в противном случае ранг должен быть понижен, и обучаемому предлагаются более легкие задания. Вопросы, на которые не были получены ответы, можно задать обучаемому вновь при получении правильных ответов на задачу с более низким рангом. Если задачи всех рангов данного уровня решены, то обучаемый переходит к решению задач следующего уровня сложности. Если предъявляемые задачи успешно решены, используется ускоренный метод обучения, при котором фактор ускорения автоматически увеличивает ранг или уровень обучения. Если же окажется, что обучаемый не может решить ни одной задачи предложенного ранга, он постепенно понижается вплоть до миним. значения. Язык реализации СОАСН — расширенный Алгол.

Л.Н.Гецко.

КОБОЛ (от англ. COmmon Business Oriented Language — универсальный язык, ориентированный на коммерческие задачи) — язык программирования, ориентированный на решение задач обработки данных (в основном коммерческой информации). Разработан в 1960 организацией КОДАСИЛ. Ныне используется стандарт языка К., принятый в 1977.

К. был первым языком, в котором появились развитые средства обработки файлов с различной организацией (последовательной, индексно-последовательной, относительной), поддержки сложных типов данных — записей, в т.ч. имеющих иерархическую структуру, оформления входных и выходных данных в виде таблиц. Стандарт К. предусматривает возможность обработки таблиц, сегментации программ, межпрограммной связи, обмена данными с телекоммуникационными каналами связи, работы с библиотеками, использования средств отладки программ. Особенность К. — использование слов английского языка в качестве обозначений операторов; операторы К. напоминают предложения английского языка. Такие обозначения были приняты для облегчения изучения К. специалистами, не имеющими матем. подготовки. В советском стандарте предусматривается использование русских слов для обозначения операторов.

К. широко использовался на компьютерах IBM-360, IBM-370, ЕС ЭВМ. В настоящее время из-за распространения систем управления базами данных, электронных таблиц, интегрированных пакетов К. стал использоваться значительно реже.

В.А.Бардадым.

КОД — конечная последовательность *символов* (букв), принадлежащих единому конечному алфавиту, составленная на основе правил ее однозначной интерпретации. Такая последовательность наз. также кодовым словом или просто словом. Алфавит совместно с правилами интерпретации образуют систему *кодирования*. Количество символов алфавита наз. основанием К. или основанием системы кодирования. Основание К. не может быть менее 2. Количество символов в кодовом слове наз. его длиной или разрядностью. Напр., шестиразрядный К. — это слово из шести букв. К. могут быть непозиционными и позиционными. В непозиционных К. значение каждого символа алфавита единственно и неизменно. Примером непозиционного К. может служить слово человеческой речи, представленное в алфавите звуков (произнесенное) или букв (написанное). В позиционных К. каждому символу алфавита присуще множество значений, не пересекающееся ни с одним из множеств других символов данного алфавита, т.е. никакие два символа одного алфавита ни при каких условиях не могут принимать одно и то же значение. Конкретное значение символа, одно из множеств, определяется по местонахождению (позиции) символа в кодовом слове в соответствии с правилами данной системы кодирования. Во всех позиционных системах счисления числа представляют собой позиционные К., основание которых равно количеству цифр системы счисления, их алфавит — это набор её цифр, а правила интерпретации таковы:

$$a_{n-1} a_{n-2} \dots a_1 a_0 = a_{n-1} g^{n-1} + \dots + a_1 g^1 + a_0 g^0,$$

где g — основание системы счисления (кодирования); $a_i = 0, 1, 2, \dots, g-1$ — набор ее цифр (алфавит); n — длина кода. Представление чисел в непозиционной системе счисления громоздко; работать с ними неудобно, и обработка их с приемлемой точностью в цифровой вычислительной машине технически невозможна. Цифровые ЭВМ работают с двоичными позиционными К., представляющими собой двоичные числа (числа в двоичной системе счисления). Основание этих К. равно 2, т.е. алфавит содержит две буквы (цифры) 0 и 1. Выбор двоичной системы счисления для цифровых ЭВМ обусловлен тем, что электронные элементы ЭВМ, имеющие два состояния, более надежны в работе и при этом проще в изготовлении и дешевле, чем элементы, у которых более двух состояний.

Количественное выражение (значение) каждой цифры в двоичном числе находят из выражения

$$a_m a_{m-1} \dots a_3 a_2 a_1 a_0 = a_m 2^m + \dots + a_2 2^2 + a_1 2^1 + a_0 2^0,$$

где $a_i = 0$ или 1; a_i наз. разрядом или битом (a_0 — нулевой разряд, a_5 — пятый разряд).

При вводе информации в ЭВМ или выводе из ЭВМ применяются восьмиричные (основание 8) и шестнадцатеричные (основание 16) К.; они легко переводятся в двоичные, с ними удобно работать.

Сообщения, хранимые на *носителях информации* и передаваемые на большие расстояния, часто бывают представлены в виде последовательности самокорректирующихся К. Такие К., в случае возникновения ошибки, могут быть исправлены с высокой степенью достоверности.

И.А.Емченко, В.Л.Леонтьев.

КОДИРОВАНИЕ — представление сообщения (*информации*) последовательностью *кодов*. Последовательности *символов* и кодов составляются на основе правил их однозначной интерпретации. Алфавит и правила однозначной интерпретации кодов и кодовых последовательностей образуют систему К. Перевод сообщения из одной системы в другую также наз. кодированием.

Если сообщение дискретно, т.е. состоит из последовательности отдельных элементов, то каждый элемент сообщения можно закодировать отдельным кодом. Если сообщение непрерывно, то его перед К. разбивают на отдельные элементы в соответствии с необходимой точностью представления, а затем кодируют как дискретное.

Системы К. бывают равномерными и неравномерными. При равномерном К. все кодовые слова имеют одинаковую длину, а при неравномерном — различную. В алфавите систем неравномерного К. должно содержаться не менее трех букв, одна из которых служит исключительно для разделения слов (этой буквой может быть пробел). Системы неравномерного К. часто применяют, если статистическая структура сообщений известна. При К. более часто встречающихся элементов сообщения словами меньшей длины количество символов, затраченных на сообщение, получается минимальным, что позволяет экономить *память ЭВМ* при хранении и время при передаче сообщений. К системе неравномерного К. относится человеческий язык. Системы К., применяемые в программировании, наз. *языками программирования*. Внутренний язык ЭВМ представляет собой систему двоичного равномерного позиционного К. См. также *Ассоциативное кодирование*, *Трансформационное кодирование*, *Цветовое кодирование изображений*.

И.А.Емченко, В.Л.Леонтьев.

КОЛЛЕКТИВНАЯ ОРГАНИЗАЦИЯ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ — одна из организационных форм учебного процесса. В *компьютерной технологии обучения* компьютер рассматривается как средство организации совместной деятельности *преподавателя* и *обучаемых*, самих обучаемых и обеспечивает такие формы их взаимодействия: а) разделение действий и *операций* при решении *учебных задач* между разными участниками и их кооперацию; б) взаимный контроль и оценку действий и операций обучаемых;

в) совместное моделирование задаваемых преподавателем схем преобразования объекта; г) отображение одним обучаемым способа решения задачи, осуществленного другим. Систематическое применение в учебном процессе К.о.у.д., опосредованной компьютером, способствует овладению элементами *компьютерной грамотности*, усвоению программного материала, а также в ряде случаев ведет к возникновению кооперирующего эффекта, способствующего улучшению межличностных отношений между обучаемыми, повышает эффективность совместного решения учебных и практических задач. Особенности применения К.о.у.д. в условиях компьютерной технологии обучения разрабатываются психологом В.В.Рубцовым и его сотрудниками.

Е.Д.Маргулис.

КОМАНДА (фр. *commande* — приказ) — код, определяющий операцию вычислительной машины и ее операнды. Состав, назначение, размер и расположение отдельных частей (полей) К. наз. форматом К. и задается при конструировании ЭВМ исходя из архитектуры ЭВМ и специфики К. Код операции содержится в операционной части К., информация об операндах (значения, адреса или указатели на адреса) — в адресной части, состоящей из нескольких полей. В зависимости от количества операндов К. могут быть безадресными, одно-, двух- или трехадресными. К. также иногда называют операторы языков программирования (как правило, машинно-ориентированных). В диалоговых системах под К. подразумевают одну из разновидностей элементарного акта диалога, задаваемую человеком и определяющую операции диалоговой системы и их операнды.

В.А.Бардадым.

КОММУНИКАТИВНАЯ ЗАДАЧА (от лат. *communicatio* — сообщение, передача) — решаемая участником общения задача усовершенствования знания, которым обладает партнер по общению (реципиент). Сущность этого усовершенствования состоит в приведении знания реципиента в соответствие с относительно совершенным знанием, которым владеет решатель. Применительно к реципиенту-человеку решение К.з. предполагает организацию решения реципиентом соответствующей познавательной задачи (или обеспечение этого решения, если сам реципиент поставил перед решателем данную К.з.).

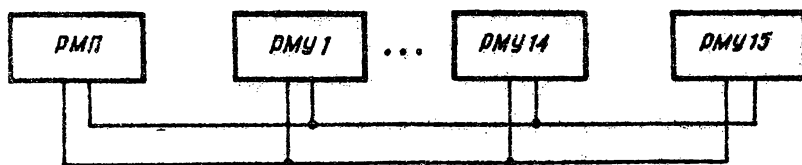
Г.А.Балл.

КОМПИЛЯТОР (англ. *compiler* — составитель) — транслятор, входным языком которого является язык высокого уровня, а объектным — машинный или близкий к нему язык. Выполняет

функции: ввод программы; лексический анализ (выделение и классификацию лексем — элементов входного языка, образованных цепочками символов его алфавита); синтаксический анализ; семантический анализ; синтез промежуточной формы (представление программы на внутреннем языке некоторой абстрактной машины, удобное для дальнейшей обработки); оптимизацию (преобразование промежуточной формы, целью которого является улучшение характеристик программы); генерацию программы на объектном языке. Кроме того, К. сигнализирует обо всех обнаруженных при обработке исходной программы формальных ошибках. В отличие от интерпретаторов, применение К. приводит к тому, что выполнение программы в К. возможно только после полного завершения преобразования исходной программы. Существует К. для языков Фортран, Паскаль, Ада, Бейсик и др.

В.И.Отенко.

КОМПЛЕКС УЧЕБНОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (КУВТ) — совокупность технических и программных средств для организации массового и профессионального обучения. Обычно КУВТ включает рабочее место преподавателя и до 15 рабочих мест обучаемых, объединенных локальной сетью учебного назначения. В качестве учебной вычислительной техники используются ПЭВМ, отвечающие требованиям эффективной организации процесса обучения (рис.).



Каждый компьютер имеет аппаратно реализованный интерпретатор языка Бейсик. Рабочее место преподавателя оснащено двоячным накопителем на гибких магнитных дисках и печатающим устройством для получения протоколов и распечатки программ. Все компьютеры имеют возможность подключения накопителей на магнитной ленте. К программным средствам КУВТ относятся: учебная операционная система с обеспечением работы локальной сети; текстовый редактор; графический редактор; учебный язык программирования; учебная база данных; набор учебных игр; библиотеку

обучающих программ. Используют КУВТ "Корвет", КУВТ УКНЦ, КУВТ-86 (бывш. СССР), YAMAHA MX1 и MX2 (Япония).

В.Н.Семихов.

КОМПЬЮТЕР — то же, что и *электронная вычислительная машина.*

КОМПЬЮТЕРИЗАЦИЯ — внедрение компьютеров в различные виды человеческой деятельности. Является основой совершенствования технологии и средством оптимальной (с позиции рациональности и экономичности) организации общественного производства. Соответственно повышается значимость ценностного аспекта как прерогативы человеческого подхода, возникают вопросы координации иерархии ценностного и рационального, целей и средств, выгоды и последствий. С другой стороны, существует опасение относительно преобразующего влияния компьютерной технологии, которая будет отражаться на самой методологии познания и художественного творчества, а также процесса потребления результатов этой деятельности. Наиболее серьезная проблема — влияние компьютера в плане замены предмета человеческой деятельности, которая утрачивает свой предметный характер и становится гносеологической, т.е. деятельностью не с предметами, а с различными формами их знакового отображения, что может привести к искажению смысла деятельности, лежащего в сфере бытия. С К. тесно связаны *информатика*, субстратом которой является компьютер, и *кибернетика*, теоретические концепции которой положены в основу принципа действия компьютера.

Осн. функция компьютера в учебном процессе — *помощь преподавателю в управлении обучением, воспитании и развитии обучаемых.* Исследования по проблеме К. в сфере педагогического управления, где компьютер выступает как инструмент решения задач, направлены на выяснение вопроса о функциях, которые можно возложить на компьютерную технику на разных иерархических уровнях управления. Выделяют следующие уровни решения проблем: 1-й — управление *учением* и развитием отдельного обучаемого; 2-й — управление учебным процессом в рамках одного учебного заведения; 3-й — управление работой группы родственных учебных заведений; 4-й — управление работой учебных заведений, объединенных по территориальному признаку; 5-й — управление системой народного образования всей страны.

Задачи управления на 1-м уровне (*индивидуализация обучения, управление процессом обучения, контроль за качеством знаний, развитие познавательных и личностных качеств обучаемого*) совпадают с задачами *компьютерного обучения.* Компьютер позволяет

существенно изменить способы управления учебной деятельностью, напр., погружая обучаемого в определенную игровую ситуацию. По сравнению с традиционными формами изменяется контроль за деятельностью обучаемого путем обеспечения гибкости управления учебным процессом. Применение компьютера даст возможность педагогам качественно изменить методы и организационные формы своей работы: полнее сохранять и развивать индивидуальные способности обучаемого; осн. внимание уделять формированию познавательных способностей, развитой учебной деятельности, поддерживать и развивать стремление к самосовершенствованию; усилить междисциплинарные связи в обучении, комплексность, обеспечить неразрывные взаимосвязи между естествознанием и техникой, гуманитарными науками и искусством; осуществлять постоянное динамичное обновление организации учебного процесса, форм и методов его осуществления, обеспечить де бюрократизацию учебных заведений, их постоянную адаптацию к изменяющимся внешним условиям и обновляющемуся контингенту обучаемых, дать им возможность активно участвовать в подготовке этих изменений. Общение с компьютером в процессе учебной деятельности дает возможность преодолеть пассивность обучаемых, усиливает их мотивацию, удовлетворяет потребность в самоконтроле, самокорректировке, позволяет лучше усвоить практические знания и выработать тот уровень интеллекта школьников, который развивается обычно на 3-4 года позже.

На 2—5-м уровнях компьютер используется для автоматизации, обработки данных о процессе обучения, об учебной работе преподавателей, администраторов, для составления расписания занятий, экономического планирования работы школы, системы школ и вузов, для определения и прогнозирования количества требуемых специалистов и уровня их подготовки. Что касается задач собственно педагогического управления (2-го уровня), направленного на совершенствование многоплановой учебно-воспитательной деятельности в учебных заведениях разного типа, то по объективным причинам (временной нехватки компьютерной техники или же стихийно сложившегося парка компьютеров с разным уровнем их технического совершенства) эти вопросы решаются эпизодически. Наиболее разработаны в теоретическом отношении и апробированы на практике проблемы управления на 3-м уровне: автоматизированное управление работой общеобразовательных школ в системе просвещения, ПТУ в системе профтехобразования, вузов и техникумов в системе высшего и среднего специального образования. На этом уровне автоматизированные системы управления позволяют существенно повышать качество финансовых расчетов и управленческих решений, относящихся к текущему и перспективному планированию контингента учебных заведений, определению потребностей в педагогических кадрах, рациональному размещению учебных заведений разного типа,

прогнозированию выпуска специалистов и т.д. Управление работой учебных заведений, объединенных по территориальному признаку (4-й уровень) остается сложной трудноразрешимой задачей прежде всего из-за ведомственной разобщенности учебных заведений разного типа, а также различной степени готовности к преобразованиям, связанным с информатизацией школ, вузов, детских садов, ПТУ, средних специальных учебных заведений. В связи с этим проблема автоматизации управленческой деятельности на этом уровне требует предварительного решения организационных вопросов, связанных с координацией различных по уровню, профилю и ведомственной принадлежности образовательных учреждений того или иного региона. Автоматизированное управление системой народного образования всей страны (5-й уровень) — пока еще постановочная, прогностическая задача, поскольку и в организационном, и, особенно, в педагогическом отношении предстоит большая работа по обеспечению функционирования целостной системы непрерывного образования с четко налаженным механизмом взаимодействия и преемственности всех ее звеньев. Решить задачу управления народным образованием на всех уровнях можно, планомерно внедряя новую информационную технологию, позволяющую максимально интегрировать информационные процессы путем перевода их на единый процесс обработки. Для нее характерны: работа *пользователя* в режиме манипулирования данными (пользователь должен “видеть” и “действовать”, а не “знать” и “помнить”); сквозная информационная поддержка на всех этапах прохождения *информации* на основе *интегрированной базы данных*, предусматривающая единую унифицированную форму представления, хранения, поиска, отображения, восстановления и защиты данных; безбумажный процесс отработки документа, при котором на бумаге фиксируется лишь окончательный вариант; интерактивный (диалогический) режим с широкими возможностями для пользователя; возможность коллективного исполнения документов на основе группы *персональных электронных вычислительных машин*, объединенных средствами коммуникации; возможность адаптивной перестройки формы и способа представления информации.

При К. психолого-педагогических исследований использовать компьютерную технику можно в таких направлениях: сущность и специфические особенности педагогических законов и закономерностей, способы их объективизации, процедуры выявления и формулирования (только путем машинной обработки значительных по объему массивов опытно-экспериментальных данных — опросов, анкетирования, поведенческих показателей, физиологических параметров и т.п. — можно выявить и теоретически интерпретировать статистически устойчивые признаки, раскрыть причинно обусловленные, закономерные связи и отношения разнообразных педагогических явлений, дать качественный содержательный анализ соответствующих регулярностей); оперативная обработка педагогической информации

(количественная обработка эмпирических данных с помощью компьютера способствует объективной содержательной характеристике выявленных закономерных связей, позволяет поднять на качественно более высокий уровень процесс их поиска и научного обоснования, усиливая прогностические функции, а следовательно, и реальные возможности применения в педагогической практике); процесс отбора содержания обучения, построения квалифицированных характеристик, учебных планов и программ, систематизация и структурирование материала в учебниках и учебных пособиях (эти задачи допускают множество различных вариантов решений, а выбор наиболее оптимального связан с перебором и соответствующей оценкой альтернативных решений; моделируя на компьютере возможные последствия принимаемых решений, изменяя те или иные переменные, влияющие на конечные результаты, можно найти наиболее приемлемый вариант решения с учетом заранее обусловленных критериев оптимальности); внедрение результатов теоретических и опытно-экспериментальных разработок в педагогическую практику, что возможно за счет создания постоянно действующей автоматизированной *информационно-поисковой системы*, обеспечивающей потребителей надежной информацией о результатах научно-исследовательских работ для оперативного использования этой информации в повседневной педагогической деятельности; информационное обслуживание учителей (примером такого обслуживания может служить экспериментальная информационная система «Учитель—родители», позволяющая классному руководителю получить разнообразную информацию в области воспитательного воздействия школы и семьи: компьютер после заполнения карточки-запроса выдает информацию относительно ситуации, которая интересует учителя; подобные автоматизированные информационно-поисковые системы позволяют приблизить педагогическую науку к практике, преодолеть разрыв между результатами научно-педагогических исследований и их оперативным внедрением в практику с учетом реальных запросов каждого конкретного учителя; на их базе создаются условия для проведения сложной и трудоемкой исследовательской работы по диагностированию реальных запросов учителей, по выявлению затруднений в их учебно-воспитательной деятельности); реализация педагогикой своей прогностической функции, заключающейся в своевременном и оперативном улавливании назревающих тенденций, педагогической интерпретации и избирательном оценивании разнохарактерной информации, сведений о передовом педагогическом опыте, выдаче соответствующих прогностических данных различным потребителям для принятия соответствующих плановых и управленческих решений.

По сравнению с традиционными формами психологических исследований компьютер предоставляет ряд преимуществ: стандартное предъявление заданий, конфиденциальность автоматизированного

тестирования; однозначность и точность фиксирования ответов одновременно с быстрым получением результатов тестирования; освобождение исследователя от трудоемких рутинных действий, фиксирование большого количества дополнительной информации самим компьютером; возможность проведения адаптивных экспериментов, рандомизированных, массовых исследований в сжатые сроки; использование мощного матем. аппарата при обработке и обобщении полученных данных и снижение вследствие этого стоимости проведения эксперимента. При анализе научной деятельности, опосредованной компьютером, отмечают явление психологического барьера к применению компьютера у некоторых пользователей (см. *Психология компьютеризации*). В исследовательской образовательно-педагогической и психологической деятельности применение компьютера оказывает существенную помощь в оптимизации управленческих решений глобального и частного характера, повышает качество творческих решений, способствует оперативному внедрению результатов научных экспериментов в массовую практику.

Компьютер, являясь универсальным орудием производственной, научной, управленческой, игровой, учебной и других видов деятельности, оказывает влияние на жизнь общества и индивидуума. Применение вычислительной техники позволило резко повысить эффективность многих форм труда, расширить возможности человека в различных областях деятельности, сделав рентабельным то, что ранее было нерентабельным, и достижимым то, что ранее было недостижимо. Производимый компьютерами "информационный ресурс" общества служит повышению интеллектуальности деятельности, преобразует характер производства и познавательной практики людей, позволяет разрабатывать новые методы научного исследования, формирует новый стиль мышления. Создаваемые на компьютерной основе новые технологии дают возможность сделать производство менее энерго- и материалоемким, более экологичным. Применение компьютера вытесняет тяжелый физический или монотонный механический труд, поскольку эти функции передаются машинам и механизмам. Теряет также смысл дифференциация видов труда по признаку пола исполнителя.

К. порождает новые отношения между специалистами, создает новые производственные и организационные структуры. Преобразуются функции учреждений, происходит непрерывное перераспределение работников, пополняющих информационную сферу за счет производственной. Такая тенденция будет сохраняться и впредь. В развитом технологическом обществе большая часть населения будет занята в информационной сфере, эффективность которой существенно возрастает за счет безбумажной информатики. Специфичным для данного этапа является изменение характера труда, разрушение традиционного единства (пространственного и временного) субъекта и трудового процесса, считавшегося его неременной характерис-

тикой. Производство, действующее в автоматическом режиме, не требует совпадения во времени производственного процесса и обеспечивающей его человеческой деятельности. Равным образом интеллектуальный характер деятельности человека снимает необходимость ее пространственной определенности. Ликвидируются существенные различия между свободным и рабочим временем. С этим связана и тенденция к отмиранию многочисленных производственных коллективов и децентрации населения в городах. Реализация компьютером некоторых интеллектуальных процедур изменяет не только деятельность, но и мышление специалиста, что отражается в постановке задачи, логике мышления и пр. Автоматизация значительной части процедур высвобождает человека для творческой работы.

Новые формы деятельности существенно изменяют психологию и сознание человека, предъявляя новые требования к его познавательным, операциональным, эмоциональным характеристикам, модифицируя его мотивационную сферу. Эти изменения носят противоречивый характер, включая как позитивные, так и негативные аспекты. С одной стороны, понимание знаковых структур требует развития предметно-ориентированного мышления; происходит раскрепощение личности, повышение ее самостоятельности в *принятии решений*; обусловленный автоматизацией производства рост свободного времени и интеллектуальность "орудия" (компьютера) способствуют развитию человеческой личности; возникают новые способы общения, новые формы культуры. С другой стороны, немало проблем порождено самой К. Так, стандартизация знаний и мыслительных процессов приводит к содержательным и смысловым потерям. Массовое отчуждение профессиональных знаний от специалистов с передачей их в пользование другим влечет за собой деперсонализацию знаний. Трудности в передаче философских и гуманитарных форм мышления, ориентация на алгебр. стиль мышления, оперирующий дискретными последовательностями *символов*, абстрагированных от предметного мира, потеря живых контекстов деятельности и мышления создают опасность дегуманизации мышления вследствие доминирования его логических, формальных компонентов. Ощущение человеком безграничности своих возможностей в компьютерной среде, обеспечивающее психологическую комфортность, чреватого погружением в компьютерную реальность, ослаблением социальных связей с внешним миром. Чтобы избежать возможных трансформаций деятельности и общения в иллюзорно-компенсаторные формы, следует определить должное место компьютера в контексте предметно-практической деятельности и живого общения.

С проникновением компьютера в сферу обучения происходит определенный кризис школьного обучения. Массовое применение *обучающих систем* лишает школу ее роли "главного орудия"

обучения, осн. функцией школы становится воспитательная. При компьютерном обучении исчезают качественные различия между обучением знанию и обучением труду, между обучением и игрой. Следовательно, информатизация и К. должны рассматриваться не только как проявления или предпосылки научно-технического процесса. Это социальное явление, способствующее преобразованию познавательной практики людей, формированию нового мышления, созданию новых подходов и методов научного исследования, важный фактор формирования индивидуального и общественного сознания.

Е.Г.Коберник, Е.Ю.Комиссарова, Е.Д.Маргулис.

КОМПЬЮТЕРИЗИРОВАННОЕ ОБУЧЕНИЕ, обучение с помощью компьютера — использование преподавателем компьютерной техники в качестве одного из значимых средств обучения. Если компьютерное обучение осуществляется полностью с помощью компьютеров в системе “компьютер — обучаемый”, без активного участия педагогов (собственно обучающая программа, составленная до начала обучения, во внимание не принимается), то К.о. проводится с участием педагогов в системе “преподаватель — компьютер — обучаемый”. Упоминание об участии компьютера в обучении фактически означает, что, кроме него, используются и другие средства обучения. Речь идет, напр., о системах “преподаватель — компьютер — учебник — обучаемый” или “преподаватель — компьютер — учебный кинофильм — обучаемый” и многих других.

Н.М.Розенберг.

КОМПЬЮТЕРНАЯ ГРАМОТНОСТЬ — совокупность интеллектуально-практических средств, необходимых для успешного общения с компьютером, использования его в своей деятельности. При анализе проблематики, связанной с понятием “К.г.”, различают две осн. линии. Одна из них относится к уточнению ее содержания, другая — к способам формирования соответствующих знаний, умений и навыков. В содержании важны такие характеристики, как динамичность и конкретно-историческая обусловленность. Напр., в учебниках “Основы информатики и вычислительной техники”, изданных в середине 80-х г., отражена соответствующая этому периоду точка зрения на содержание понятия “К.г.” и способы его усвоения. Критика в адрес этих учебных пособий, различные дополнения и предложения по переносу акцентов (отход от основ алгоритмизации как ядра изучаемого предмета, снижение внимания к изучению языка программирования или алгоритмического языка в пользу изучения операционных систем, в частности текстового редактора, процессора электронных таблиц, графопостроителей, широкое применение новых информационных технологий в учебном процессе и т.д.), изменению и расширению возрастных категорий

обучаемых, изучающих новый предмет, свидетельствовали об изменении представлений о содержании К.г. на рубеже 90-х гг. и позволяли судить о перспективах его дальнейшего развития.

Наряду с динамичностью важное значение имеет такая характеристика понятия "К.г.", как его отнесенность к конкретному контингенту обучаемых: содержание К.г. должно быть связано с профессиональной или социальной группой, на которую оно ориентировано. Так, должны быть существенные различия между К.г. школьников, инженеров и педагогов. Кроме того, К.г. нужно рассматривать как сложное системное образование, имеющее иерархическую структуру и функционирующее (и формируемое) на различных уровнях. Напр., педагогам — предметникам и классовадам начальных классов требуются неодинаковые знания в области *компьютеризации* обучения. Существуют и более дифференцированные различия — между применением компьютера в учебном процессе вуза, в профессиональном обучении и общеобразовательной школе, между *компьютерным обучением* в рамках дисциплин естественно-математического и гуманитарного циклов и т.д. Поэтому выделять ряд уровней К.г. учителей: 1) обязательный, содержащий основы К.г. (рассматриваемой как компонент общеобразовательной и профессиональной подготовки учителя); 2) пользовательский, обеспечивающий практическую реализацию компьютерной технологии обучения (предназначен для тех, кто использует компьютер в учебном процессе и должен знать, как это делать эффективно); 3) уровень разработчика (ориентирован на тех, кто сам принимает участие в создании педагогических сценариев, проектов и учебного обеспечения компьютерных систем обучения). Здесь, как отмечает украинский психолог Е.И. Машбиц, необходима специальная подготовка в области теории и технологии *проектирования, применения и оценки обучающих программ*.

Ситуация, возникшая в середине 80-х гг., характеризовалась, с одной стороны, актуализацией проблемы компьютерного обучения, осознанием того, что отставание в этой области недопустимо, а с другой — недостаточностью средств для решения ее в полном объеме. Наиболее остро это проявилось в сфере материально-технического обеспечения. В условиях временного дефицита ЭВМ в школе приходилось идти путем так наз. безмашинного варианта изучения основ *информатики и вычислительной техники*, подключать школьные кабинеты информатики и вычислительной техники к большим ЭВМ, принадлежащим базовым предприятиям и учреждениям, широко использовать в практике обучения программируемые микрокалькуляторы, рассматривая это как подготовительный этап и прообраз работы обучаемых с *персональными электронными вычислительными машинами* и т.д.

В деле использования вычислительной техники неправомерны как недооценка возможностей, которые она открывает, так и преу-

величение их, сводящееся порой к утверждению, что компьютеризация якобы решает все проблемы. Поскольку она только начинает развиваться, необходим объективный, взвешанный подход к оценке достигнутого. Решающей же силой всех перемен был и остается человек, поэтому прежде всего с повышением роли умственного фактора связано успешное решение задач научно-технического прогресса в социально-экономическом развитии страны. В связи с этим очевидна необходимость овладения К.г. педагогическими работниками. Уже сейчас преподавателям не только естественно-матем. наук, но и гуманитарных — филологам, историкам, географам, а также учителям начальных классов необходимо знакомиться с проблемами компьютеризации, осваивать практические навыки владения вычислительной техникой. Ныне уже существует немало кружков компьютерного обучения для младших школьников, и классовод должен использовать новые возможности для усвоения и закрепления программного материала, знать имеющийся в школе фонд программ и применять его в своей педагогической практике. Очевидно, адекватные представления о возможностях компьютера как средства обучения и воспитания школьников должны стать важной предпосылкой и неотъемлемым компонентом профессиональной подготовки учителей. В этом направлении ведется большая работа, прежде всего — в системе повышения квалификации учителей путем коррекции учебных программ в педагогических институтах, организации специальных курсов для учителей в различных вузах. Результативность такой подготовки зависит от эффективности решения психолого-педагогических проблем применения компьютера в обучении. В этом — предпосылки успешного использования новых информационных технологий, которые должны войти в арсенал профессиональной деятельности учителя.

Овладение К.г. с психологической точки зрения предполагает формирование нового вида деятельности, требующего особых средств (компьютеры, матем. и программное обеспечение, знания о способах решения и средствах решения задач и т.д.). При обучении информатике и использованию ЭВМ сложились два осн. подхода. Один основывается на изученных психологической наукой и апробированных в педагогической практике различных модификациях проблемного обучения и программированного обучения, при другом — на первый план выдвигаются психологические проблемы диалога с компьютером, процесс обучения с помощью компьютера рассматривается как особым образом организованное общение с ним. В работах сов. ученых были описаны осн. механизмы порождения и преодоления проблемных ситуаций в условиях традиционного обучения. Эти результаты должны быть переосмыслены и творчески использованы в новых условиях, с учетом компьютеризации учебного процесса. Это особенно важно, поскольку формирование интеллектуальных компонентов К.г., т.е. соответствующий уровень логичес-

кого мышления, овладение *алгоритмической культурой*, освоение осн. этапов процесса решения задач с помощью компьютеров можно обеспечить при проблемной организации обучения информатике, включению в учебный процесс проблемных ситуаций. В программированном обучении отход от дискредитировавших себя принципов линейных программ, построенных на бихевиористической модели подкрепляемого поведения (по сути, отождествляющих деятельность человека и поведение животных) и сводивших цели обучения к повышению вероятности правильных ответов, развитие технологии компьютерного обучения в "сторону человека", т.е. в направлении все более гибких программ, позволили сохранить жизнеспособность такого метода обучения. Хотя бескомпьютерный вариант программированного обучения не оправдал возлагавшихся на него в свое время надежд, отказываться от его применения нецелесообразно. Объективно любой вид компьютерного обучения является, как отмечает академик Н.Ф.Талызина, обучением программированным (т.е. обучением с опосредствованным управлением: его осуществляет не непосредственно *преподаватель*, а заранее разработанная *программа, программированный учебник* или техническое устройство, в частности компьютер). Поэтому при работе использовать уже известные по исследованиям программированного обучения результаты: требования к выбору оптимального размера шага *обучающей программы*, к *программированию* учебного материала, подлежащего усвоению, к реализации *обратной связи в обучении*, обеспечивающей оценку деятельности обучаемого и позволяющей строить адекватные *обучающие воздействия*, и т.д. Ныне многие учебные курсы по различным дисциплинам (в частности — обучение информатике и основам вычислительной техники) строятся как программированные. Компьютеризированное программированное обучение имеет свою специфику, прежде всего в связи с резко возросшими возможностями ветвления и адаптивности программ (см., напр., *Индивидуализация обучения, Учение*).

К.г., являясь компонентом интеллектуальных достояний, оказывает влияние на развитие личности обучаемого, на его мышление, внимание, память и другие психические процессы. Интеллектуальные компоненты К.г. предполагают умение обучаемых формулировать цели деятельности, анализировать условия задачи и средства ее решения, строить алгоритм, работать в диалоговом режиме, организовывать поиск и структурирование *информации*. В совокупности эти интеллектуальные умения образуют структурно-модульный рефлексивный стиль мышления, характеризующийся точным определением нужного результата, полным анализом исходных условий, выделением и координацией иерархической системы подцелей, построением алгоритма, предусматривающего наличие формального

исполнителя, сформированностью *стратегий* эффективной коррекции алгоритма на основе обратной связи о результатах его выполнения.

При овладении К.г. важно, чтобы механические действия, формальное исполнение алгоритмов не становились определяющими — творческое, продуктивное мышление должно преобладать над рутинным, непродуктивным. Уже разработаны психологические требования к такому содержанию К.г., которое способствует развитию творческого мышления, готовности обучаемых к творческому труду. Особенно важно повысить внимание к начальным, творческим этапам постановки задачи, которая решается с помощью компьютера. Акцент в содержании обучения переносится с навыков и умений составления алгоритма и программы работы компьютера на соответствующие умения грамотной постановки *задачи* и ее формализованного описания. Основа К.г. в таком понимании — глобальное умение решать задачи с помощью компьютера, состоящее из следующих взаимосвязанных умений: а) вовремя находить и вычленять проблемные ситуации; б) правильно ставить задачу, выбирая соответствующее знаковое оформление; в) строить формализованное описание задачи или *предметной области* с учетом сервисных и рабочих средств используемой системы, соотносить модель задачи, отражающую ее понимание *пользователем*, со средствами решения, представляемыми компьютером, — *языком программирования*, сервисными возможностями, объемом *памяти ЭВМ*, структурой *данных* и др.; г) намечать и реализовать эффективную стратегию решения, уметь адекватно использовать имеющиеся средства программного обеспечения и из них конструировать алгоритм решения.

Е.Д.Маргулис.

КОМПЬЮТЕРНАЯ ИГРА — игра, реализованная на компьютере, выступающем либо в качестве одного из участников игры ("игра с компьютером"), либо как средство игровой деятельности ("игра с помощью компьютера"). Различают два осн. вида К.и. — игры типа *game* и *play*. Большинство К.и. относятся к первому типу. Второму типу соответствует понятие "игровая среда" (см. *Компьютерная учебная среда*).

Один из наиболее продуктивных способов представления игры по правилам (типа *game*) связан с ее трактовкой как диалоговой программы, реализующей *алгоритм* некоторой игры и обеспечивающей взаимодействие игрока с компьютером.

К.и. состоит из следующих осн. компонентов: *модели предметной области*, (состоит из множества *моделей* объектов, множества отношений между объектами и множества операций, заданных над отношениями и объектами; см. *Учебная игра*), множества целей и критериев оценки игровой деятельности.

Основу К.и. составляет сюжет (сценарий), которым является связная совокупность всех игровых событий (ситуаций). При проектировании и реализации К. сюжет разделяется на фон и динамические объекты. Фон на протяжении всей игры остается неизменным и не зависит от развития сюжета. Однако некоторые объекты фона связаны отношениями с динамическими объектами и могут вступать во взаимодействие с ними. Развитие сюжета определяет только фон, на котором в данный или следующий момент происходит действие сюжета, т.е. сюжет определяет смену доступных для игрока фрагментов фона. Сюжет включает совокупность объектов (так наз. персонажей), связанных между собой функциональными отношениями. Здесь выделяются объекты, связанные с действиями игрока и не зависящие от них (соответственно управляемые и неуправляемые объекты сюжета). Управляемые объекты (герои игры) перемещаются по фрагменту фона в зависимости от управляющих воздействий играющего, передаваемых игровой программе посредством клавиш управления курсором и небольшим количеством дополнительных функциональных клавиш, первые из которых управляют перемещением объекта, а вторые — его поведением (прыжок, выстрел, поворот и т.п.). Перемещение и поведение неуправляемых объектов на данном фрагменте фона описывается функциональными отношениями именно с этим фрагментом фона или со всем фоном, причем функции управления объектом не зависят от воздействий игрока. Нужно только отметить, что аргументами этих функций могут быть координаты управляемого объекта. Тогда неуправляемые объекты становятся косвенно управляемыми, т.е. либо сокращается расстояние между управляемым и косвенно управляемыми объектами (преследование героя), либо расстояние увеличивается (герой преследует других персонажей). Переходы от данного фрагмента фона к другому осуществляются в соответствии с результатами действий игрока на данном фрагменте. Поэтому для осуществления переходов с фрагмента на фрагмент необходимо выделить типовые функциональные зависимости, определяющие схемы переходов, характерные для игр. Сложность схемы переходов и критерии для выбора каждого конкретного перехода должны быть увязаны с множеством целей данного (игрового) вида деятельности.

Программа, реализующая алгоритм игры, состоит из двух частей: первая реализует ее логическую (внутреннюю) структуру, вторая отображает процесс игры на экране терминала — интерпретирует знаковые модели объектов и связей в графические (см. *Учебно-игровая программа*).

Существует ряд классификаций К.и. Среди наиболее распространенных — сюжетно-тематическая и функционально-психологическая классификации. Первый вариант классификации основан на содержании задачи, решаемой пользователем в игре. Так, выделяются игры-головоломки, настольные интеллектуальные игры в компьютер-

ном варианте, азартные шансовые игры, управленческо-экономические, спортивные, военные, игры-единоборства, игры преследования-избегания, авантюрные, конвейерные, конструктивно-динамические, диалоговые познавательные, учебно-технологические игры и т.д. Второй вариант классификации основан на выделении осн. функций игры: социально-приспособительной, рекреативной, функции самоиспытания, психотренинга, соревнования, образования, мотивационного управления учебной или производственной деятельностью, функции объекта конструктивной деятельности и имитационного тренинга.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

КОМПЬЮТЕРНАЯ ОСВЕДОМЛЁННОСТЬ — начальный уровень компьютерной грамотности. Предполагает: общее представление об осн. устройствах, входящих в состав компьютера; знание осн. возможностей и сфер применения компьютеров; наличие опыта простого диалога с вычислительной машиной. К.о. помогают формировать средства массовой информации, научно-популярные издания, фильмы, видеопередачи, выставки, клубы, фестивали и другие формы пропаганды.

Н.М.Розенберг.

КОМПЬЮТЕРНАЯ ПСИХОДИАГНОСТИКА (от греч. ψυχη — душа и διαγνωσις — распознавание) — направление внедрения новой информационной технологии, связанное с применением компьютера для диагностики состояний субъекта, его психологических свойств, качеств, процессов. К.п. — это технология классификации и ранжирования людей по психологическим и психофизиологическим признакам с использованием компьютеров, наука и практика постановки психологического диагноза с помощью компьютеров. Осн. цель современной К.п. — фиксация, диагностирование и описание в упорядоченном виде психологических индивидуальных внутригрупповых и межгрупповых различий. Психологическая диагностика с применением компьютеров осуществляется с помощью разнотипных инструментальных и диагностических средств, которые можно объединить в понятие компьютеризированных диагностических методик (компьютеризированных тестов, анкет и др.).

Компьютеризация учебного процесса в средней и высшей школе позволила ставить и решать многие проблемы, которые в условиях традиционного обучения лишь декларировались, но практически не решались на должном уровне. Показательной в этом отношении является проблема индивидуализации обучения и дифференцированного подхода к обучаемым. Педагогическая диагностика, цель которой — установление обратной связи в обучении, требует от преподавателя значительного объема рутинной работы, связанной с

контролем и оцениванием, а также оперативного использования этой информации для управления процессом обучения. Еще сложнее задачи психодиагностики, цель которых — установление обратной связи для получения данных о формировании психических качеств обучаемых, об особенностях протекания психических процессов в учебной деятельности, о мотивации и эмоциях, памяти и т.д. Все это, как правило, даже самые опытные преподаватели учитывают лишь на интуитивном уровне. Психологический "портрет" каждого обучаемого строится на основе несистематизированных наблюдений, отрывочных сведений и в большинстве случаев оказывается неполным и даже неадекватным. В результате вопросы психодиагностики оказываются на периферии учебного процесса, практически не решаются, поэтому и принцип индивидуализации обучения не может быть эффективно и последовательно реализован.

Применение компьютера в учебном процессе открыло реальные пути для практической реализации психодиагностических методик и их автоматизированной интерпретации, нивелировало проблемы, связанные с затратами времени, необходимостью специальной подготовки, трудоемким сбором, хранением и обработкой экспериментальных данных. Существенно и то, что если раньше психодиагностические комплексы разрабатывались для больших ЭВМ, то с начала 80-х гг. появление микро-ЭВМ позволило использовать психодиагностические процедуры в любой организации, в частности в школе. Вместе с тем возник ряд новых проблем. Одна из них — корректное использование традиционных психодиагностических методик при работе с компьютером. Нередко существующие диагностические процедуры автоматически переносятся на компьютер, причём на него возлагается не только обработка данных, но и, напр., предъявление обучаемым стимульного материала. Это перенесение неправомерно: проведение методики с использованием компьютера — это, по сути, новая психодиагностическая процедура, требующая традиционных способов валидации, стандартизации, проверки надёжности и т.д. Лишь немногие типы методических средств (напр., опросники и их обработка) можно перенести на компьютер без дополнительной доработки и адаптации. Существенно и то, что для каждого типа компьютера и периферийного оборудования, если они используются для предъявления обучаемым стимульного материала, требуется отдельная проверка (напр., результаты методик, использующих цветность, могут существенно различаться из-за различных оттенков цветов, реализованных в данном мониторе, разрешающей способности экрана и т.д.). Вторая проблема связана с повышением доступности психодиагностических методик для непрофессионалов, которые могут неправильно применять такие методики и неадекватно интерпретировать их результаты, разглашать полученные данные и т.п. Поэтому целесообразно различать методики с профессиональными ограничениями (П-методики) и открытые методики (О-методики).

В.М.Бондаровская, Е.Д.Маргулис, Н.И.Повякель.

КОМПЬЮТЕРНАЯ УЧЁБНАЯ СРЕДА — учебная среда, реализуемая средствами компьютерной технологии обучения. Представляет собой связанную по целям взаимодействия совокупность аппаратных, программно-информационных средств планирования, организации и формирования учебной деятельности. К аппаратным средствам относятся дисплей, процессор, дисковая память ЭВМ и другие компоненты компьютера. Программно-информационные средства включают различные режимы взаимодействия, модели предметной области, описания учебных задач и средства работы с этими моделями и описаниями, специальные пакеты прикладных программ, обеспечивающие диалоговое взаимодействие и реализацию изучаемых моделей и процессов. Учебно-методические средства представлены в виде инструкций, рекомендаций, словарей, справочников, пособий, учебников и т.п. Связность указанных средств К.у.с. позволяет расширить информационную структуру компьютерного обучения. Это достигается за счет подкрепления курса учебным пособием с более широким перечнем учебных задач, т.е. в К.у.с. обучаемый знакомится с базовыми понятиями курса и учится решать типовые классы задач. Однако при недостаточно четком понимании некоторого понятия пользователем система отсылает его к учебному пособию (вспомогательной ветви обучающей программы), где производятся соответствующие примеры и аналогичные задачи. Целесообразно различать четыре уровня проектирования К.у.с. — проблемно-, сюжетно-, объектно- и процедурно-ориентированный (в зависимости от того, осуществляется ли проектирование на уровне решения научно-практической проблемы, разработки сюжета, описания системы объектов, фигурирующих в игре, или процедур, реализующих операции над объектами).

Примерами К.у.с. могут служить системы АФРОДИТА и ЭЛКОН. Напр., в среде системы АФРОДИТА реализуется обучение правилам описания конкретных типов данных языка Фортран, а в учебном пособии приводится детальное изложение учебного материала с примерами. Фиксируя действия, которые обучаемый применяет к объектам учебной задачи, К.у.с. осуществляет контроль за правильностью их использования. При необходимости К.у.с. отсылает обучаемого к пособию. В системе ЭЛКОН обучаемому предлагается сюжет его учебно-игровой деятельности и объекты, реализованные в среде системы. Обучение осуществляется в процессе построения композиции из имеющихся в распоряжении обучаемого объектов и функций. Учебное пособие в данном случае подкрепляет информационную модель, представленную в среде системы объектов и учебных задач.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

КОМПЬЮТЕРНОЕ ОБУЧЕНИЕ, обучение на базе компьютера — использование компьютера в качестве средства обучения, т.е.

компьютерная реализация *обучающих программ* (для формирования знаний, умений, приобретения опыта творческой деятельности и т.д.). Первые попытки использования компьютера в учебно-воспитательном процессе относятся к концу 50-х — началу 60-х гг. Технически это сводилось к осуществлению *диалога* “человек-компьютер”, причем предлагаемые компьютером вопросы печатались преимущественно на телетайпе, а ввод *ответов* осуществляется с помощью *клавиатуры*. Опыт обучения с помощью компьютера в те годы имел ограниченный, лабораторный характер. Существует преемственность между понятиями компьютерного и *программированного обучения*. Сохранив основные характеристики программированного обучения (*индивидуализация обучения, усиление роли самостоятельного поиска новых знаний, управление по принципу обратной связи в обучении*), компьютер обеспечивает такие новые характеристики, как автоматизация вычислительных процедур, выполнение логических и информационно-поисковых заданий, предъявление *информации* в виде графиков, цветных движущихся изображений и др. Появляются также новые сервисные возможности, позволяющие осуществлять *решение задач с помощью компьютера* на качественно ином уровне (это, напр., использование *текстового редактора*, автоматизированное выявление и локализация ошибок, удобства в структурировании и обработке *информации* и т.д.). По сути К.о. создает некую сложную надстройку над программированным обучением.

Н.М.Розенберг.

КОМПЬЮТЕРНЫЕ ВИРУСЫ — специально созданные и, как правило, небольшие *программы*, способные заражать другие программы включением в них своей, возможно модифицированной, копии (которая сохраняет способность к дальнейшему размножению). Программа зараженная К.в., может рассматриваться как автоматически созданная троянская программа, скрытым модулем которой является тело К.в. Кроме заражения К.в. можно выполнять и другие несанкционированные действия, от вполне безобидных до крайне разрушительных (напр., уничтожение *данных* на зараженном диске). В этом случае К.в. может рассматриваться как логическая мина. Несанкционированные действия могут быть обусловлены наступлением определенной даты (такие троянские программы в какой-то степени аналогичны минам с часовым механизмом) или определенного количества рамножений, либо сочетанием определенных условий, напр. записи зараженной программы на *винчестерский накопитель* (такие троянские программы аналогичны бесконтактным минам). При этом комбинация условий может быть достаточно сложной, чтобы затруднить ее определение (как, напр., в К.в., известном под названием “Пинг-понг”).

Упрощенно процесс заражения программных *файлов* можно представить следующим образом. Код зараженной программы обычно изменен так, чтобы К.в. получил управление первым, до начала работы программы-вирусоносителя. При передаче управления вирусу он каким-то способом находит новую программу и вставляет собственную копию в начало либо добавляет ее в конец этой, обычно еще не зараженной, программы. В последнем случае он корректирует код программы, чтобы получить управление первым. Для этого несколько первых *байтов* запоминаются в теле вируса, а на их место вставляется команда перехода на начало вируса. Получив управление, вирус восстанавливает "спрятанные" первые байты, а после отработки своего тела передает управление программе-вирусоносителю, и она нормально выполняет свои функции. Иногда К.в. включает в себя куда-то в середину программы, напр. в область стека. Зараженные программы (либо их копии) могут передаваться через *дискеты* или по сети на другие компьютеры. Поскольку пользователи *персональных электронных вычислительных машин* (ПЭВМ) широко обмениваются или передают программы на дискетах, количество зараженных программ может быть значительным, приводя к своего рода эпидемиям. Этому также способствует распространенная в нашей стране практика использования одной ПЭВМ несколькими *пользователями*. Опасность существенно возрастает при наличии *винчестера*. В этом случае один неквалифицированный пользователь может нанести значительный ущерб другим. Особую опасность представляют любители *компьютерных игр*, обычно слабо знающие операционную систему (ОС) и не вполне понимающие смысл выполняемых ими действий.

Первые программы, напоминающие К.в., были созданы в 1972 (игра "Дарвин") и в 1976 (системная программа WORM ("Червь")). В 1984 была распространена игра ANIMAL ("Животное"), которая имела почти все черты современного К. в. Программа, реализованная как игра, предлагала играющему с ней задумать некоторое животное и за определенное время пыталась при помощи уточняющих вопросов (имеет ли волосы, живет ли в воде и т.д.) отгадать задуманное. Если программа не успевала отгадать задуманное животное, то "смирно" просила игрока сообщить, какой вопрос нужно задать, чтобы отгадать. Этот вопрос запоминался и таким образом программа "самообучалась". Но разработчик ее использовал, вероятно, тот факт, что на его компьютере всякому пользователю отводился собственный каталог для работы, но не был механизм защиты каталога от других пользователей (или программ). Программист изменил игру так, что после каждого "поумнения" программа копировала себя в каталог другого пользователя. Игра была популярна, и скоро все компьютеры фирмы содержали медленно разрастающуюся и множущуюся программу. Ситуация не была опасной, но многочисленные копии существенно засоряли дисковое пространство. Уничтожение копий не

дало нужного эффекта достаточно было оставить единственную копию и скоро все размножалось вновь. Вопрос был окончательно решен с помощью средства, которое по современной терминологии может быть названо *антивирусом*. В 1985 была опубликована игра "Core War" ("Война в памяти"). В ней два игрока пишут по одной программе каждый на языке низкого уровня (см. *Уровень алгоритмического языка*) Redcode. Программы помещаются в большой циклически замкнутый участок памяти. Каждая команда занимает одну ячейку *памяти ЭВМ*. Управляющая программа поочередно исполняет одну команду каждой программы, подобно простейшей системе реального времени (см. *Режим разделения времени*). Программы атакуют друг друга, пытаются избежать повреждений и восстанавливая поврежденные области. Простейшая атака состоит в использовании команды MOV (записать в память). Напр., MOV 1000,0 может "убить наповал" вражескую программу, если попадет в следующую исполняемую команду (т.е. если следующая выполняемая команда "врага" расположена по адресу 1000) или "ранить", если это данные либо исполняемые команды, или наконец, "попасть мимо", если ячейка 1000 противной стороной не используется.

Появление настоящих вирусов, т.е. программ, ориентированных специально на нанесение различного рода ущерба, относится к началу 80-х гг., связано с появлением ПЭВМ и носит характер противостояния пользователей группе безответственных или уголовных элементов. Первые случаи массового заражения компьютеров были отмечены в 1987. Так, Лехайский вирус, появившийся в одноименном университете США, в течение нескольких дней уничтожил содержимое нескольких сот дискет как из публичной библиотеки университета, так и личных дискет студентов. 30.XII 1987 был обнаружен К.в. в Иерусалимском Университете (Израиль). Хотя существенного вреда этот вирус не принес, он быстро распространился по всему миру и, по-видимому, является первым вирусом, распространение которого приобрело характер пандемии. В 1988 эта проблема в странах США и Западной Европы приобрела характер приоритетной. Особое внимание общественности привлек так наз. вирус Морриса: 2.XI 1988 Роберт Моррис-младший, аспирант факультета информатики Корнелльского университета, инфицировал с помощью написанного им вируса большое количество компьютеров (по ориентировочным оценкам, порядка 6000), подключенных к американской национальной сети Internet. Хотя никакой потери или изменения данных не произошло, многие тысячи часов рабочего времени были потеряны пользователями Internet. К концу 1989 г. в ряде стран (США, Великобритания, ФРГ) рассматривались законы, предусматривающие для разработчиков и распространителей К.в. значительные сроки тюремного заключения (в США до 15 лет). Это позволит, помимо программных и организационных мер, для борьбы с вирусами применять и правовые методы. Поэтому важным аспектом

борьбы с компьютерными вирусами становится выявления личности преступника или преступной группы, ответственной за создание или распространение соответствующего К.в.

Подобно биологическим вирусам, для большинства К.в. характерен определенный инкубационный период, в течение которого выполняемые вирусом несанкционированные действия ограничиваются заражением других программ или дискет. Инфицируя программы или дискеты вирусы могут распространяться от одной программы (дискеты) к другой, что делает их более опасными по сравнению с другими методами компьютерного вандализма. Операционная система MS-DOS, отличающаяся практически полным отсутствием защиты от несанкционированных действий, облегчает разработку К.в. Однако К.в. не являются программами, использующими ошибки или недостатки конкретной ОС. Для обеспечения своего функционирования достаточно вполне обычных операций, используемых большинством "нормальных" программ. Поэтому принципиально не может существовать универсальный метод, защищающий ОС от распространения любого вируса. Тем не менее можно существенно затруднить задачу создания и распространения К.в., как применяя специальные методы в ОС, так и используя дополнительные резидентные и нерезидентные программные средства защиты.

Последствия, вызываемые при проявлении К.в.: отказ системы в выполнении той или иной функции (напр., блокирование вирусом RC=1701 загрузки программы с защищенной от записи дискеты), выполнение действий, не предусмотренных программой (напр., изменение данных в к.-л. файле); разрушение отдельных файлов, управляющих блоков или всей файловой системы в целом (форматирование диска, удаление файла и т.д.); выдача ложных, раздражающих или отвлекающих сообщений (напр., "Скажи себе"); создание звуковых или визуальных эффектов (напр., падение букв в вирусе RC-1701, замедление выполнения программы в вирусе RCE-1913, проигрывание мелодии в RCE-1805, движущийся на экране ромбик в Vx1=1C и т.д.); инициирование ошибок или сбоев в программе либо ОС (напр., переполнение стека), перезагрузка или зависание операционной системы MS-DOS; блокирование доступа к системным ресурсам (разрастание зараженных файлов за счет их многократного повторного заражения, невозможность передать зараженной программе параметры, замедление работы компьютера путем выполнения холостого цикла из нескольких команд при каждом прерывании таймера); имитация сбоев аппаратуры (перевод части кластеров в псевдосбойные на дискете или винчестере, "зависание" компьютера через некоторое время после перезагрузки ОС и т.д.); ускорение износа оборудования или попытки его порчи. Наносимый К.в. ущерб может иметь катастрофический характер (напр., уничтожение винчестера) в сочетании с длительным "инкубационным периодом" либо, наоборот, К.в. может наносить мелкие повреждения

данных (выполняемые достаточно часто), которые гораздо труднее обнаружить, вследствие чего они намного опаснее массивованного разрушения данных.

Наиболее уязвимая часть файловой системы ДОС — таблица размещения файлов. Если она разрушена, то ДОС не может определить местонахождение того или иного файла, хотя сами файлы не повреждены. К.в. может иже выполнять формализацию некоторых участков диска, содержащих системные данные. Поэтому необходимо достаточно часто дублировать управляющие данные файловой системы на другой, а ранее известный участок диска или дискету. Для этой цели можно использовать, напр., утилиты Нортон. На компьютерах типа АТ данные о конфигурации системы (тип установленного винчестера и др.) хранятся в небольшой энергозависимой памяти (CMOS). Уничтожение содержимого CMOS-памяти приводит к невозможности загрузиться с винчестера. Восстановление CMOS-памяти приводит к невозможности загрузиться с винчестера. Восстановление CMOS-памяти требует знания подробных технических данных о винчестере. Поэтому этот тип памяти также является потенциальным объектом атаки вируса. К.в. “бессмертны” и могут неограниченное время храниться в различного рода архивах. Даже полностью “уничтоженные” вирусы могут сохраниться в каком-либо архивном файле и случайно или умышленно “реанимироваться” через много месяцев или даже лет после их первоначального обнаружения и уничтожения. Следовательно, после появления определенного вируса необходимы специальные меры по предотвращению повторных заражений. Здесь можно двигаться в двух направлениях: найти первоисточник заражения либо разработать или установить программы, затрудняющие (сторожа) или делающие невозможными (вакцины) размножение вируса.

Далеко не все повреждения файловой системы, отказы винчестера или оборудования вызываются вирусами. Напр., некоторые типы винчестеров имеют довольно низкую надежность и “сыпятся” без всякого вмешательства вирусов. Есть компьютеры, которые можно загрузить, только дав им прогреться в течении получаса. Существуют дисководы, которые не только фрезеруют дискеты, но и при записи иногда стирают таблицу размещения файлов, причем восстановить ее при помощи утилит Нортон не удастся. Многие отечественные компьютеры и без всяких вирусов регулярно “зависают”. Однако имеется тенденция атрибутировать любое повреждение данных в присутствии К.в. Ходят слухи о К.в., повреждающих оборудование. Действительно, такие повреждения возможны. Напр., можно повредить участок люминофора (“выжечь пятно”) на монохроматическом мониторе, используя особенности схемы управления. Однако для цветного монитора это сделать невозможно. Ходят также слухи о каких-то коварных вирусах, якобы вводящих в резонанс головки винчестера. Очевидно, такая мифотворческая тенденция возникает в

связи с любым малоизученным и потенциально опасным явлением. Проблема “вирус — средства защиты” аналогична проблеме “оружие нападения — оружие защиты”. Поэтому следует ожидать, что длительное время К.в. будут актуальной проблемой, причем совершенствование средств защиты будет сопровождаться совершенствованием самих вирусов.

Трудности характеристики различных средств защиты от К.в. и значительное дублирование этих средств делают актуальным вопрос о классификации К.в. Эта классификация должна позволять однозначно охарактеризовать не только известные К.в., но и новые их разновидности по ограниченному числу сравнительно простых и непротиворечивых признаков. Н.Н. Безруков предлагает классификацию, в которой имя К.в. состоит из буквенного префикса, цифрового корня и, возможно, буквенного суффикса. Буквенный префикс характеризует среду размножения К.в.: можно выделить четыре основных типа К.в.: располагающихся в бут-секторе и сбойных секторах на диске (тип В или *бутовые вирусы*), в файлах типа .COM (тип С, относящийся к *файловым вирусам*), в файлах типа .EXE (тип Е, относящийся к *файловым вирусам*), передающиеся по сети (тип N, или так наз. сетевые вирусы). Цифровой корень характеризует длину вируса. Для К.в. типа С, Е и SE (файловых вирусов) он принимается равным приращению длины файла при заражении. Это приращение для ряда файловых вирусов не стабильно и зависит как от типа файла (.COM или .EXE), так и от длины заражаемого файла (напр., при дописывании своего тела в конец заражаемого файла некоторые вирусы выравнивают начало своего тела на начало параграфа, т.е. на смещение, кратное 16). В этом случае в качестве цифрового корня принимается миним. длина приращения. Для вирусов типа В в качестве аппроксимации длины принимается количество используемых секторов, умноженное на 512 (длину сектора). К.в. также делятся на резидентные и нерезидентные. Нерезидентные вирусы получают управление после загрузки в память зараженной программы, а затем ищут файл-жертву, используя PATH или другую *информацию*, и заражают этот файл. Затем управление возвращается зараженной программе и после ее окончания память компьютера освобождается от К.в.. В отличие от них резидентные вирусы после загрузки в память и передачи управления зараженной программе перехватывают ряд прерываний и остаются в памяти резидентными. Получив управление после того или иного прерывания, они выполняют определенные действия (напр., заражают каждую запускаемую программу, заражают .EXE и .COM файлы при копировании и т.д.). Для классификации таких вирусов используется суффикс R.

Структурно К.в. можно представить состоящими из двух частей: головы и хвоста. Головой наз. часть К.в., первой получающая управление, хвостом — часть его, расположенная отдельно от головы.

В простейшем случае вирус состоит из одной головы (большинство файловых вирусов). Такие К.в. наз. несегментированными. В отличие от них сегментированные К.в. имеют хвост и в какой-то мере аналогичны оверлейным файлам. Примером сегментированных К.в. являются б/т-овые вирусы, хотя возможна реализация сегментированных файловых вирусов.

В операционной системе MS ДОС наиболее распространены файловые нерезидентные, файловые резидентные и б/т-овые вирусы. В настоящее время для этой ОС существует несколько сотен компьютерных вирусов.

Н.Н.Безруков.

КОНВЕРГЕНЦИЯ (лат. *convergentio*, от *con* + *ergo* — сближаюсь, схожусь) — физиологический акт сведения зрительных осей обоих глаз на фиксируемом объекте. Осуществляется рефлекторно при фузии (слиянии изображения) с помощью наружных мышц глаза. Необходима для обеспечения бинокулярного зрения. Различают тоническую, аккомодативную, фузионную и проксимальную К. Длительная работа с видеотерминалом, особенно при вводе данных, при частом переводе взора с экрана на печатный текст, проявляется утомлением (мышечной *астенопией*), характеризующимся жалобами на боли в надбровной области и раздвоенностью изображения.

А.С.Коваленко, В.Г.Мартиросова.

КОНТЕ́КСТ (от лат. *contextus* — тесная связь, соединение) — законченный в смысловом отношении отрезок письменной или устной речи (текста), в пределах которого можно точно установить значение отдельного входящего в него слова или фразы. На интерпретацию конкретного предложения связного *текста* оказывают влияние как предыдущие предложения (текстовой К.), так и ситуационное окружение, в котором это предложение встречается (ситуационный К.). Ситуационный К. в связи с концепцией умолчания, свойственной естественным языкам, обычно в явном виде не присутствует в тексте. Он воссоздается участниками в процессе понимания ими *диалога*. Разделение на текстовой и ситуационный К. подчеркивает способ формирования К. — явный (текстовой) и неявный (ситуационный). В зависимости от задач, решаемых при анализе связного текста, выделяют глобальный и локальный К. Глобальный К. включает общую направленность текста и его ситуационное окружение. На основании его решается проблема выбора смысла многозначных слов (напр., “ядро”: снаряд для стрельбы; сердцевина ореха; ядро атома и т.п.). Локальный К. содержит предложения, непосредственно предшествующие обрабатываемому предложению. Используется для раскрытия “умолчаний” в

предложениях, содержащих эллипсис (пропуск в предложениях слова или словосочетания, понятного из контекста).

Н.А.Власенко, Л.Н.Гецко.

КОНТРАСТ — субъективная оценка разницы зрительного восприятия двух частей поля зрения, наблюдаемых одновременно или последовательно. Для численной характеристики К. используют яркостный К. — относительную разницу яркости между двумя частями зрительного поля (обе части этого поля могут значительно отличаться по размерам). Коэффициент К. определяется по формуле:

$$C = (L_1 - L_2) / (L_1 + L_2),$$

где L_1 — яркость наименьшей части (символа); L_2 — яркость наибольшей части (фона). Контрастное отношение определяется по формуле:

$$C_r = (L_1 + L_2) / L_2,$$

где L_1 — яркость информационного элемента; L_2 — яркость фона. К. изображения на экране дисплея зависит от окружающего освещения, т.е.

$$\begin{aligned} L_1 &= L_{1i} + L_e, \\ L_2 &= L_{2i} + L_e, \end{aligned}$$

где L_{1i} — яркость символа в темноте; L_{2i} — яркость фона в темноте; L_e — отраженная яркость окружения, следовательно,

$$C = (L_{1i} - L_{2i}) / (L_{1i} + 2L_e + L_{2i}).$$

Яркостный К. изображения на дисплее характеризуется двумя взаимосвязанными свойствами: относительными яркостями L_1 и L_2 ; плавными (по сравнению с письменно-печатными копиями) градиентами яркости между символом и фоном, что дает относительно расплывчатые символы.

Увеличить контраст можно уменьшением яркости фона, увеличением яркости символа или уменьшением отражения от окружающего освещения. Кроме того, краевой К. можно в некоторой степени компенсировать увеличением некоторых характеристик, связанных с К., одной из таких характеристик является размер знака. Увеличить К. можно использованием фильтров, пропускающих больше света от индикатора, чем к индикатору, применением специального антибликового покрытия экранов, затемнением экрана с помощью бленд и рациональным размещением поверхности экрана относительно источников света.

Коэффициент К. влияет на выбор оптимального уровня яркости в зависимости от величины яркости, отраженной от поверхности экрана, которая в свою очередь зависит от освещенности этой поверхности, создаваемой окружающей средой. Для представления точечных данных можно использовать контрастное отношение 3:1, но для обеспечения высокой точности опознания символов лучше выбрать 10:1. Для негативного (светлые символы на темном фоне)

К. (при уровне внешнего освещения 300 лк) рекомендуется: оптимальная яркость символов 80 — 160 нит; яркость символов должна быть регулируемой, символы не должны расплываться и должны быть хорошо различимыми при установке на максимум яркости; яркость фона должна быть не более 20 нит. При соблюдении этих требований можно получить оптимальное контрастное отношение в пределах 8:1 — 10:1. для позитивного (темные символы на светлом фоне) К. (при уровне внешнего освещения 300 лк) рекомендуется: яркость фона должна быть в пределах 80 — 170 нит; яркость фона должна быть постоянной по всей поверхности экрана. При этом можно получить оптимальное контрастное отношение в пределах 1:8 — 1:12. Физиологически позитивный К. разрешает многие из проблем внешней среды, такие, как *блескость* отражения, которые обычно встречаются при негативном К. Наиболее простой способ уменьшить блескость и отражения на экране — соответствующая установка дисплея по отношению к источникам света и отражений.

С.А.Миронченко, В.С.Пашковский, И.М.Рощина.

КОНТРОЛЬ ЗНАНИЙ — оценка соответствия знаний *обучаемого* планируемой норме. *Задачи К.з.* относятся к классу диагностических (см. *Диагностика знаний*). При К.з. большое значение имеет его достоверность — вероятность получения оценки, соответствующей уровню знаний. Различают методическую, алгоритмическую и инструментальную достоверности.

Под методической достоверностью подразумевается вероятность получения правильной оценки знаний при учете неадекватности модели *представления знаний*, процесса и формы их усвоения. Алгоритмическая достоверность — это вероятность того, что выбранный план К.з. обеспечивает правильность оценки знаний. Инструментальная достоверность определяется вероятностью правильного ввода и обработки результатов К.з. в компьютере. Достоверность К.з. может характеризоваться также степенью недооценки или переоценки знаний.

Ю.И.Лобанов.

КОНФЛИКТНОЕ ВЗАИМОДЕЙСТВИЕ — вид групповой совместной деятельности, характеризующейся наличием антагонистических целей у ее участников. Для некоторых решающих систем (РС), находящихся в К.в., одной из наиболее приоритетных целей является деструктурирование деятельности противника. Этим конфликт отличается от соревновательной деятельности, являющейся квазисовместной, а наиболее приоритетные цели ее участников предполагают получение *продуктов деятельности*, которые по каким-то заранее

оговоренным признакам являются более эффективными, нежели у других участников соревнования (таким продуктом может быть и сам процесс деятельности, если, напр., ставится задача достижения наименьших отклонений от технологии производства процесса), а также от всех видов кооперативной деятельности, предполагающих наличие непустого множества пересечения актуализированных целей у субъектов деятельности (*решающих систем*).

К.з. имеет большое значение в играх. Напр., каждая игра двух и более участников с ненулевой суммой выигрышей является, по сути, К.в. Такая позиция характерна для матем. теории игр, находит свое отражение и в ряде широко используемых определений игры. В более широком социальном контексте под конфликтом понимаем разрыв между целями деятельности и средствами для ее достижения — так наз. внутренний конфликт РС. Такой разрыв может возникнуть при появлении помех (ограничение и искажение информации, средств и ресурсов деятельности, снижении мотивации к ней и пр.) или в случае деструктурирования деятельности конфликтующей РС (*противником*).

Анализ конфликтов и способов их предотвращения и устранения играет важную роль в исследованиях *взаимодействия человеко-машинного*. В условиях внедрения вычислительной техники в народное хозяйство промышленно развитых стран особую актуальность приобретает проблема разработки так наз. дружественных *диалоговых систем* различного назначения для обеспечения комфортных условий профессиональной деятельности, особенно для начинающих *пользователей* компьютерных технологий (к которым относятся и *обучаемые*). Примером может служить анализ К.в. двух РС при решении прикладных и учебных задач. При решении прикладных задач осн. типы целей взаимодействия РС таковы: 1) Ц_в — выполнить решение задачи самостоятельно, 2) Ц_п — передать задачу партнеру; при решении учебных задач, 3) Ц_{уч} — усовершенствовать свои знания и умения, 4) Ц_{об} — усовершенствовать знания и умения партнера. Цели 1,3 можно отнести к деятельности одной РС, цели 2,4 предполагают наличие взаимодействия со второй РС и соответствуют отказу РС от самостоятельного решения поставленных задач.

Для анализа случаев возникновения конфликта удобно использовать вектор целей РС, у которой могут быть все указанные типы целей. Элементами этого вектора являются единицы для обозначения целей, относимых к одной РС, и нули — для целей, соответствующих отказу РС от деятельности и относимых к взаимодействию двух решающих систем РС1 и РС2 (табл.).

PC1	PC2	$\Pi_{в2}$	$\Pi_{п2}$	$\Pi_{уч2}$	$\Pi_{об2}$
		1	\emptyset	1	\emptyset
$\Pi_{в1}$	1	\emptyset	1	\emptyset	1
$\Pi_{п1}$	\emptyset	1	\emptyset	1	\emptyset
$\Pi_{уч1}$	1	\emptyset	1	\emptyset	1
$\Pi_{об1}$	\emptyset	1	\emptyset	1	\emptyset

В таблице несогласованные пары целей обозначены нулями (в случае конфликта). Напр., пара $\langle \Pi_{в1}, \Pi_{в2} \rangle$ соответствует *соревновательной игре*, когда каждый из партнёров стремится решить некоторую задачу самостоятельно, причём лучше или быстрее другого с целью получить известное вознаграждение (игровая цель $\Pi_{и}$). В общем случае осуществлять К.в. достаточно продолжительное время трудно. Одним из способов продолжения К.в. может быть введение во взаимодействие некоторой игровой цели (как в случае соревновательной игры). Согласованные пары целей представлены в таблице единицами — случай *кооперативного взаимодействия*. Аппарат анализа целей взаимодействия основывается на изучении структуры деятельности обучаемого в рамках *модели предметной области*. Такой подход позволяет описывать и строить различные структуры эффективных сценариев *диалога* с последующей их генерацией.

Э.Л.Ивахненко, Е.Д.Маргулис.

КОНЦЕПТ — специализированная *авторская система*, под-держивающая этап концептуального проектирования при разработке информационно-справочных, обучающе-контролирующих, тренирующих и моделирующих программ учебного назначения (ПУН). Была реализована в Институте кибернетики имени В.М.Глушкова АН Украины на IBM PC и может быть использована на классе персональных электронных вычислительных машин, совместимых с IBM PC. В основе концептуального проекта, разработанного с помощью системы К. лежат формы, подобные *Zeitgeist*. Однако систему можно настроить на спецификацию других типов ПУН, если в качестве содержания проекта с помощью *текстового редактора* системы К. ввести формы другого типа К. позволяет создавать спецификации ПУН с помощью специализированного текстового редактора. Автору проекта для работы в системе не требуется предварительной подготовки, т.к. все функции системы отображаются в *меню* и подсказках по функциональным клавишам. Кроме того, *пользователь* может получить справки о работе в системе и содержании самого проекта.

Система позволяет работать на двух уровнях: содержание проекта и раздел проекта. На первом уровне автору доступны следующие режимы работы: загрузка и сохранение *файла* проекта, просмотр

содержания проекта и выбор раздела проекта, печать проекта, получение кратких справок по разделам работы системы и содержанию каждого раздела проекта. При выборе раздела проекта автор попадает на второй уровень — в текстовый редактор системы, с помощью которого автор вводит спецификации текущих разделов в том виде, в каком они будут выглядеть в результирующем документе, или редактирует уже введенный *текст*, если соответствующий файл проекта был загружен. Содержание каждого раздела проекта может занимать 16 страниц; ширина страницы экрана соответствует ширине печатной страницы. Редактор позволяет вводить любой *символ* по его коду, а также получать расширенную справку по выбранному разделу проекта с примерами разработки.

Т.А.Лисюк.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ (лат. *conceptus* — понятие, идея, общее представление) — символическое представление основных отношений системы. Напр., К.м. в физике являются: *модель* газа как совокупности твердых шаров, модель жидкости как непрерывной среды с заданной плотностью, модель атома и пр. В КТО *концептуальная модель учебных знаний* отражает совокупность знаний *обучаемого* об изучаемом предмете, в качестве *концептуальной модели процесса обучения* используется схема целенаправленной системы управления самообучающимся объектом. К.м. служит приближенным представлением реальной системы и в процессе изучения может уточняться и даже заменяться новой.

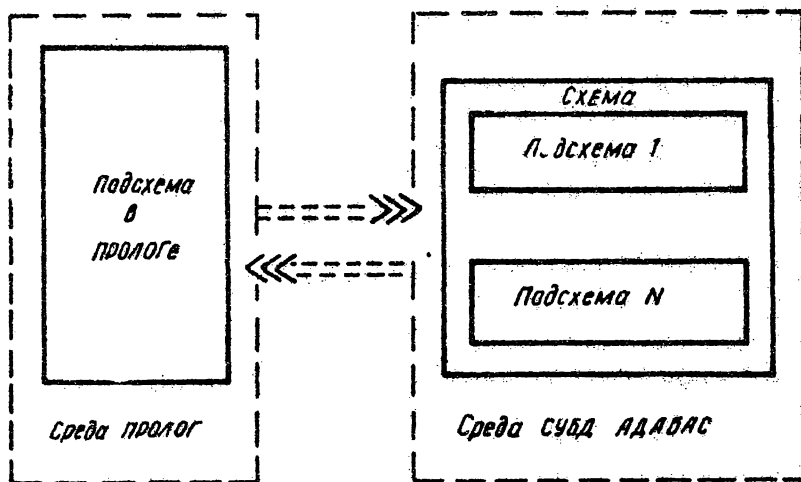
Ю.И.Лобанов.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ БАЗЫ ДАННЫХ — абстрактное отображение физической *базы данных* (БД). Для обеспечения независимости прикладных программ от данных необходимо ввести *модель данных*, которая будет отражать для пользователя полное информационное содержание БД, но физическую организацию хранения данных учитывать не будет. Модель должна иметь свою схему, отражающую структуру её данных. Схемой на... описание общей логической структуры БД; она представляет собой таблицу типов используемых данных, содержащую имена объектов и их атрибуты и определяющую существующую между ними связь. Схема представляет собой структуру, в которой могут быть помещены значения элементов данных. Для спецификации концептуальной схемы используют язык описания данных (ЯОД), позволяющий записывать концептуальную схему в терминах некоторой модели данных. Такая модель должна обладать структурой, достаточно богатой для описания существующих аспектов реального мира, и обладать возможностью автоматически определять эффективную реализацию концептуальной схемы физической схемой БД. Существ-

вуют три осн. модели, используемые в системах БД: иерархическая, сетевая и реляционная.

При проектировании концептуальной схемы администратор базы данных использует ЯОД, включающий описания схем (ЯОД-С) и язык описания данных подсхем (ЯОД-ПС). Подсхемой наз. некоторое описание данных, используемое прикладным программистом при работе с БД. Если концептуальную схему можно трактовать как "мир", то подсхема — это "взгляд на мир" для конкретного пользователя. На основе одной схемы можно составить много различных подсхем, т.е. получить "много взглядов на мир".

Использование интегрированной схемы ПРОЛОГ-АДАБАС подразумевает создание в среде Пролога множества подсхем, необходимых пользователю системы Пролог, без изменения схем или подсхем, реализованных в системе управления базами данных АДАБАС (рис.). Прикладной программист или конечный пользователь может



и не знать схему БД в целом из-за её сложности. Он должен иметь дело с теми конкретными приложениями и записями, которые ему нужны. Для наглядного представления схемы и подсхем БД используют графическое представление в виде диаграмм.

А.П.Ильяшенко, М.Е.Козлов.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ПРОЦЕССА ОБУЧЕНИЯ — символическое представление основных отношений системы обучения.

Служит приближённым представлением реальной системы обучения и в процессе изучения может уточняться и даже заменяться новой.

В качестве *концептуальной модели* процесса управления учебными действиями в компьютерных системах обучения используется схема целенаправленной системы управления самообучающимся объектом. Особенность такой системы состоит в том, что объект управления сам является целенаправленной адаптивной системой. Осн. задача управления процессом обучения формулируется как экстремальная задача преобразования знаний *обучаемого* из исходного состояния в целевое за миним. время при заданном наборе средств. Схема системы включает два осн. цикла: цикл идентификации (диагностики) и цикл управления. В первом цикле решаются задачи получения сведений о целях и состоянии *знаний* и умений *обучаемого*. Во втором — задача выбора подцели и планирования учебных действий.

Обучение рассматривается как многошаговый процесс совместного *принятия решений* и выполнения действий, направленных на достижение осн. цели — освоение требуемой системы *понятий* и действий. Текущая цель системы выбирается на основании анализа степени согласования текущих знаний *обучаемого* и осн. цели обучения. План действий конкретного *обучаемого* формируется как путь от текущего состояния знаний до желаемого. При его поиске прогнозируются возможные временные затраты.

Ю.И.Лобанов.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ УЧЕБНЫХ ЗНАНИЙ — обобщённое семиотическое описание взаимосвязи элементов *знаний* учебного назначения. Знания представляются в виде композиции трёх элементов: теории *предметной области*; модели *предметной области*; интерпретатора теории на *моделях*. Для формального *представления знаний* может использоваться язык предикатов первого порядка. При этом задание теории сводится к определению системы *предметных понятий* и их связей. Интерпретация и *модели* отражают семантический аспект теории, в отличие от синтаксического аспекта самой теории, в рамках которого делаются заключения об истинности формул теории по их внутренней структуре. Обобщённое представление понятий позволяет отчётливо выявить зависимость содержания понятий от совокупности утверждений принятой теории: содержание понятия определяется содержанием выводимых из него (или связанных с ним) утверждений и понятий.

Ю.И.Лобанов.

КОНЦЕПТУАЛЬНАЯ ПРОВЕРКА ОТВЕТОВ — сравнение *концептуальных моделей* эталонного ответа и ответа *обучаемого*. При проверке строятся концептуальные сети вопроса и сообщения

обучаемого. Затем сравниваются сети *вопроса* и *ответа*. Если сеть ответа совпадает с сетью вопроса или, являясь частью сети вопроса, включает необходимые дескрипторы (единица языка, соответствующая определённой *понятию*), ответ признаётся верным.

При сравнении сетей эталонного ответа и ответа обучаемого могут встретиться различные варианты. Если сети описываются абстрактной системой

$$S = \langle M, R \rangle,$$

где M, R — множества элементов и отношений, то сети эталонного ответа соответствует система $S_3 = \langle M_3, R_3 \rangle$, а сети ответа обучаемого $S_0 = \langle M_0, R_0 \rangle$. В этом случае возможны следующие результаты проверки: сети абсолютно различны, т.е. пересечения сетей нет: $S_3 \cap S_0 = \emptyset$; сети имеют общие фрагменты $S_3 \cap S_0 \neq \emptyset$, что возможно в двух случаях: множества дескрипторов сетей пересекаются, а связей нет: $M_3 \cap M_0 \neq \emptyset, R_3 \cap R_0 = \emptyset$; пересекаются множества и дескрипторов, и связей: $M_3 \cap M_0 \neq \emptyset, R_3 \cap R_0 \neq \emptyset$. Результаты первого варианта говорят о том, что ответ обучаемого совершенно неверен либо использует другую систему терминов. Результаты второго варианта позволяют более точно определить характер ошибки.

Ю.И.Лобанов.

КОНЦЕПТУАЛЬНОЕ ОБУЧЕНИЕ — обучение смыслу нового *понятия*. Смысл отдельного понятия представляется с помощью *концептуальной модели*, содержащей: термин, имя понятия; *модель* объекта понятия; перечень признаков объекта понятия, содержание понятия; множество объектов понятия, объём понятия.

Ю.И.Лобанов.

КОНЦЕПТУАЛЬНЫЕ ЗНАНИЯ — *знание* содержания (смысла) *понятия* в классе явлений (объектов или процессов); обобщённое представление о явлении как системе (напр., как множестве связей между элементами системы). См. *Представление концептуальных знаний*.

Ю.И.Лобанов.

КООПЕРАТИВНОЕ ВЗАИМОДЕЙСТВИЕ — вид групповой совместной деятельности, характеризующийся наличием согласованных целей у её участников. Под согласованными понимаются, напр., такие пары целей решающих систем (РС): одна РС имеет цель учиться, вторая — обучать (управлять *учением*) первую; одна РС хочет передавать решение некоторой задачи второй РС, у которой цель — осуществить решение этой задачи. К.в. возникает, как правило, при невозможности (или нежелании) РС осуществлять самостоятельную деятельность. Эта невозможность может быть

обусловлена недостаточностью ресурсов, *знаний* и *умений*, а также другими причинами. При К.в. нуждающиеся друг в друге партнёры характеризуются: взаимопониманием; направленностью на осуществление взаимодействия с целью совместного решения некоторых задач (несогласованность целей — см. *Конфликтное взаимодействие*); пониманием равноправности партнёрства. См. также *Взаимодействие человеко-машинное*.

А.М.Довгялло, Э.Л.Ивахненко.

КОРРЕЛЯЦИЯ (лат. *correlatio* — соотношение) — взаимное соответствие предметов, процессов, *понятий*. В матем. статистике — вероятностная зависимость, не имеющая строго функционального характера. Иными словами, корреляционная зависимость между двумя величинами имеет место тогда, когда каждая из них зависит не только от другой величины, но и от случайных факторов. К. двух величин положительна (или, напротив, отрицательна), если возрастание любой из них сопровождается преимущественно возрастанием (или соответственно уменьшением) другой величины. Положительная К. существует, напр., между уровнями *сложности задачи* и *трудности задачи*.

Г.А.Балл.

КРИТЕРИАЛЬНАЯ ЗАДАЧА (от греч. *κρίτηριον* — мерило) — *задача*, решение которой служит критерием достижения целей *обучения*. Напр., в результате изучения курса химии УІІІ класса школьники должны, в частности, научиться: “на основании знаний периодической системы химических элементов Д.И.Менделеева и строения атомов составлять формулы важнейших соединений, определять вид химической связи и прогнозировать характерные общие свойства веществ ...”; “на основе знания валентности элементов составлять формулы соединений, состоящих из двух элементов, формулы оснований и солей по известной валентности металлов и кислотных остатков ...”. Чтобы достаточно полно описать требования к результатам обучения, следует, во-первых, возможно более полно отразить принятые цели обучения в системе К.з., во-вторых, охарактеризовать не только сами К.з., но также и те средства их решения, которыми должны овладеть *обучаемые*, равно как и требуемый уровень владения этими средствами.

К.з. всегда являются массовыми: обучаемые должны овладеть средствами, обеспечивающими (или хотя бы облегчающими) решение не к.-л. конкретной задачи, а некоторого их класса. Используемые в обучении проверочные задачи должны быть частными видами или *моделями* критериальных. Иногда термин “К.з.” употребляется для обозначения конкретных проверочных задач. Если в рамках общего образования цели обучения получают конкретизацию в системе К.з.,

то профессиональное образование нуждается и в обратном механизме: выделенная путём анализа нормативной деятельности специалиста система К.з. служит основанием для разработки целей обучения.

Г.А.Балл.

КУРСРАЙТЕР — один из первых авторских языков программирования, ориентированный на создание автоматизированных учебных курсов (АУК). Создан в 60-х гг. в Иллинойском университете (США). Один из вариантов этого языка стал прообразом первого отечественного авторского языка программирования для создания АУК — языка описания курсов.

В.А.Третьяк.

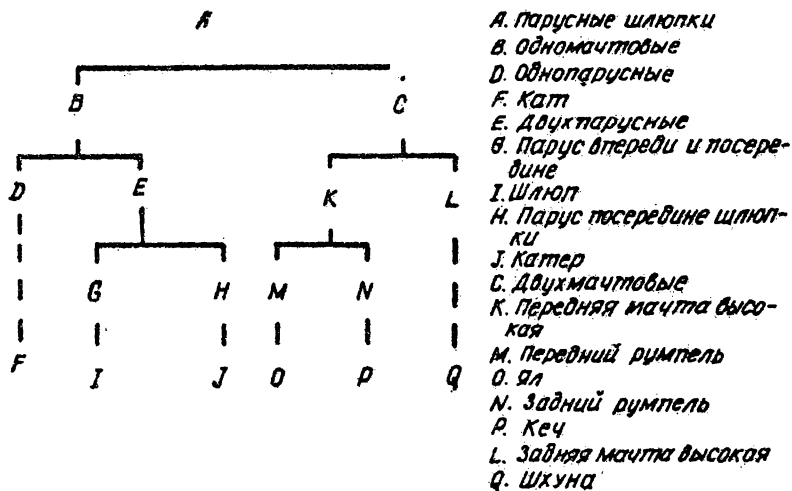
“ЛАБИРИНТ” — учебная среда, предназначенная для обучения младших школьников начальным понятиям программирования на примере составления программ для исполнителя — робота в лабиринте. Позволяет также решать в режиме диалога матем. задачи: поиск пути в невидимом лабиринте, поиск кратчайшего пути и др. В “Л.” предусмотрены следующие режимы работы обучаемого: управление исполнителем-роботом посредством клавиш-стрелок, управление роботом с записью последовательности ходов в виде программы, ввод с клавиатуры и редактирование программы, вывод программы на внешние носители информации, ввод программы с них, непрерывное или пошаговое выполнение программы. Сложность задач определяется объемом отображаемой информации о лабиринте. Преподаватель может создавать самостоятельно, редактировать, генерировать автоматически, выводить на внешние носители и вводить с них лабиринты, изменять режим отображения информации о лабиринте. “Л.” имеет демонстрационный режим, звуковое сопровождение, обеспечивает подсказки.

В.А.Бардадым.

LDS (англ. Lesson Design System — система проектирования уроков) — универсальная многоуровневая авторская система, предназначенная для автоматизации этапов проектирования и реализации программ учебного назначения (ПУН). Позволяет создавать учебные материалы на машинных носителях информации, линейные и разветвленные ПУН демонстрационного, вопросно-ответного, обучающе-контролирующего и других типов. Предоставляет автору помощь в разработке печатных учебных материалов и ПУН, наз. диалоговыми уроками. Компоненты урока наз. сегментами, которые, в свою очередь, состоят из компонентов, имеющих несколько форм представления. LDS реализована на языке Паскаль в Калифорнийском университете для ПЭВМ Apple.

Система состоит из четырех редакторов: структуры содержания; стратегии; экранов; диалоговых урков. Процесс разработки уроков в системе приблизительно соответствует таким этапам создания учебных материалов: проектирование содержания, разработка *стратегии* обучения и форм представления учебных материалов, *программирование* (для *компьютерного обучения*). Работа в редакторах основана на *меню* и подсказках. Система подсказок базируется на соответствующих теориях разработки учебных материалов. Каждый редактор может сгенерировать печатную копию результатов своей работы, а также *файл* на диске для ввода в редактор на следующей стадии или для дальнейшего анализа их автором.

Результатами работы редактора структуры содержания являются структурные схемы содержания, такие, как иерархии, таксономии (классификации *понятий*) или блок-схемы. Редактор предоставляет помощь в разработке структуры содержания изучаемого предмета. Он поддерживает три наиболее часто встречающихся типа обучающих материалов: таксономия, выполнение *процедур*, применение принципов. Каждому типу обучающих материалов соответствует свой редактор. Автор выбирает нужный ему редактор из меню. Редактор содержания работает в двух режимах: создания и редактирования. В режиме создания автор вводит записи по подсказкам. Записи автора выводятся на экран *дисплея* компьютера одновременно в двух форматах: текстовом с отступами, отражающими структуру *информации*, и графическом. В режиме редактирования *пользователь* имеет возможность модифицировать структуру содержания, добавляя, удаляя или изменяя записи. Система управляет процессом редактирования, основываясь на положениях соответствующей теории. Для *обучения* классификации понятий используется редактор таксономии, обеспечивающий построение на экране дерева решений, в котором каждая вершина соответствует некоторому атрибуту класса изучаемого понятия и имеет метку. В режиме создания автор вводит метки и атрибуты в виде структурированного *текста*, причем отступы обеспечиваются системой. Одновременно система создает на экране дерево в графическом виде (рис.). Для обучения применению принципов, лежащих в основе данных явлений, и пользуется редактор предшествования. В этом редакторе автор разрабатывает иерархическую структуру, определяющую порядок изучения принципов, понятий и фактов. Согласно встроенному в систему ограничению, обучению понятий не может предшествовать обучение принципам, а перед фактами не могут изучаться принципы или понятия. Эта структура создается и редактируется аналогично таксономии. В редакторе процедур процедура представляется множеством шагов, решений и ветвей, указывающих порядок выполнения некоторых действий. При обучении процедуре система помогает автору в выполнении структурного анализа этой процедуры. Пользователь по подсказке вводит главные шаги процедуры. Затем



каждый шаг можно разбить на более мелкие, пока размер и содержимое конечного множества шагов не будет соответствовать тому, что *обучаемый* уже знает. В основе работы редактора процедур лежат управляющие конструкции, подобные конструкциям языка Паскаль. По подсказкам системы пользователь вводит метки для каждого условия и действия и объекты действий в текстовом формате с отступами. После завершения ввода система строит блок-схему процедуры и выводит ее на экран рядом с текстом. Режим редактирования позволяет автору изменять ранее созданное описание процедуры.

Редактор стратегии основан на теории представления компонентов; согласно ей для каждого понятия, процедуры или принципа, которому обучают, существуют, по крайней мере, три первичные формы представления: обобщение, примеры и практические упражнения. Эта теория определяет также некоторые вторичные представления и область их применения. Основываясь на структуре содержания, введенной автором, редактор стратегии выделяет необходимые обучающие сегменты и предлагает порядок их представления. Тип и порядок этих сегментов отличаются для каждого из трех типов структур содержания. Затем по подсказке редактора стратегии для каждого сегмента автор определяет компоненты, необходимые для его усвоения. Редактор стратегии подсказывает автору, какие формы представления необходимо ввести для каждого сегмента. После каждой подсказки система работает, как *текстовый редактор*, принимая обучающую информацию, соответствующую

текущей форме представления. В любой момент времени автор может изменить форму представления, запросить систему проанализировать структуру, которую он определил, и указать пропущенные или неадекватно специфицированные компоненты.

Редактор экранов помогает автору подготовить статические (на печать) и динамические (на экран) выводы для каждого компонента, определенного в редакторе стратегии. Редактор работает в трех режимах: стандартные текстовые символы; стандартные плюс специальные (созданные пользователем в редакторе шрифтов) символы; стандартные символы плюс графика. Готовую страницу можно запомнить на диске и вывести на *печатаю! ее устройство*. Редактор экранов содержит образцы выводов форм представления для различных структур содержания и стратегий обучения.

Порядок динамических выводов задается с помощью редактора диалоговых уроков. *Язык программирования* специального назначения предоставляет возможность комбинирования динамических выводов, созданных с помощью редактора экранов. В редакторе диалоговых уроков автор определяет тип обработки ответов обучаемого и управление последовательностью выводов на экран. Этот редактор содержит образцы стратегий для различных видов обучения. Автору также доступны те типы обработки *ответов*, которые считаются наиболее подходящими для данных сегментов. Результатом его работы является ПУН, которую можно выполнить *операционной системой Apple*. Выходом из редактора диалоговых уроков является *программа*, управляющая представлением экранов, включающая динамические представление, ветвление и обработку ответов.

Т.А.Лисюк.

LEFO (нем. Lehrprogrammformulierungs sprache — язык формулирования обучающих программ) — *авторский язык программирования* для разработки обучающих программ для микрокомпьютеров. Разработан в 1980-83 в Дрезденском педагогическом институте и впоследствии там же опробован. С 1988 применяется расширенная и модифицированная версия языка — LEFO1. К используемому микрокомпьютеру предъявляются следующие требования: оперативная *память ЭВМ* не менее 64 К; внешняя память 780 К (дискковод типа МФС 1.6); *операционная система SCP* (совместимая с CP/M). Алфавит LEFO включает три группы *символов*: буквы (заглавные буквы латинского алфавита), цифры и специальные символы (разделители). Осн. элементы языка — числа, идентификаторы, идентификаторы задан:й и *ключевые слова*. Идентификатор должен иметь длину не более восьми символов. Для ключевого слова значимы три первых символа. *Программа* на языке LEFO состоит из заголовка и последовательности *операторов*, заключенных между ключевыми словами BEGINN (начало) и ENDE

(конец). Заголовок включает имя обучающей программы, а также может содержать *информацию* о необходимых моделирующих программах, используемых *процедурах* и трассировке хода *обучения*. Операторы состоят из ключевых слов, определяющих действие, которое должно быть выполнено над данными, задаваемыми с помощью соответствующих параметров. В языке LEO различают два класса данных: программные учебные. Программные данные являются составной частью операторов и не зависят от действий *обучаемых*. Эти данные формируются *автором* и заносятся в соответствующие *файлы* с помощью *текстового редактора*. В LEO существуют такие файлы обучающих программ: 1) файл шагов, содержащий информацию, подсказки, объяснения, реплики и другие *тексты*, которые должны выдаваться на экран *обучаемого*; каждый шаг имеет свой идентификатор; текст может держать не более восьми строк по 62 символа; 2) файл, содержащий тексты справочного характера по некоторой *предметной области*; каждый текст может содержать не более 12 строк по 62 символа; 3) файл *вопросов*, содержащий все вопросы обучающей программы, в т.ч. множество однотипных вопросов, из которого случайным образом можно выбрать конкретный экземпляр; каждый вопрос включает идентификатор вопроса, предусмотренное число баллов за *ответ*, текст вопроса (не более восьми строк по 62 символа); 4) файл ответов, содержащий предусмотренные ответы для вопросов из соответствующего файла (образцы для сравнения, данные для оценки и диагноза, возможно, идентификатор следующего шага); каждый ответ может занимать не более четырех строк по 62 знака; 5) файл путей обучения, содержащий для каждого задания допустимую последовательность прохождения учебного материала вплоть до этого задания, которая служит образцом для сравнения в условных операторах.

Учебные данные характеризуют усвоение материала *обучаемым* и собираются в процессе выполнения программы. Манипуляция этими данными производится посредством соответствующих операторов языка. LEO работает с тремя файлами учебных данных, параллельно с которыми поддерживаются три файла так наз. индивидуальных данных, содержащих данные об *обучаемом*, некоторую информацию, извлеченную из файлов учебных данных, а также дату работы; эти данные доступны после завершения работы *обучаемого* с программой. Матрица оценок содержит все необходимые данные для выполнения заданий и оценки ответов, в частности: идентификатор выполненного задания; номер случайно выбранного задания; ответ *обучаемого*; признак правильного/неправильного ответа и номер варианта ответа; возможное и полученное число баллов; идентификатор возможного симптома ошибки. Вектор симптомов содержит данные о том, какие задания обрабатывались и какие симптомы с какой частотой встречались. Вектор истории обучения содержит информацию о пройденном к текущему моменту пути в виде списка идентифика-

торов выполненных заданий в последовательности их обработки. Операторы языка реализуют в процессе выполнения программы элементарные дидактические функции, используя программные и учебные данные. Приведем примеры нескольких операторов языка LEFO. Оператор передачи информации: VER [MITTLE] <список идентификаторов>, напр., VERMLOB2, HINW3. Для всех идентификаторов, перечисленных в списке, соответствующий текст из файла шагов будет последовательно выдан на экран. Оператор решения задачи (вопроса): LOE [SE] [*] <список идентификаторов вопросов>, напр., LOE QUARD1.20, QUADR2. Для всех указанных идентификаторов вопросов последовательно выполняются следующие действия. На экран выдается текст вопроса (задание), и актуализируется матрица оценок. Система ожидает ответа, при этом, как правило, допускаются использование компьютера в режиме калькулятора, обращение за справочной информацией, а также использование имеющихся моделирующих (имитационных) программ. Полученный ответ анализируется, актуализируются учебные данные. Символ * за ключевым словом LOESE обеспечивает передачу накопленных учебных данных из матрицы оценок в соответствующий файл индивидуальных результатов. Условный оператор WENN <условие> DANN <оператор1> SONST <оператор2>. Этот оператор используется для ветвления обучающей программы в зависимости от определенных условий, напр., количества правильно и неправильно выполненных заданий, наличия некоторого симптома или варианта ответа. Другие операторы обеспечивают диагностику результатов усвоения в пределах последовательности заданий, вызов процедур, определенных автором, использование моделирующих программ, построение последовательности операторов, циклов, а также реализуют процедуру случайного выбора. Этот относительно ограниченный набор средств, предоставляемый автору, позволяет реализовать достаточно обширный класс структур обучающих программ. Поясним это на примере: символы (* и *) ограничивают комментарии к программе

(* внешний цикл *)

WIEDERHOLE

BEGINN

WIEDERHOLE 3MAL LOESE KAU.10

(*цикл повторить трижды: выполнить задание из группы KAU.10*)

FALLS ANTW VON KAU, KAU, KAU=

(* Если ответ на вопрос из 'KAU' равен: *)

1: (* это правильное решение *) VERM LOB2;

(* выдать сообщение LOB2 *)

2: (* типичная ошибка 1 *) VERM HINW2;

(* выдать пояснение HINW2 *)

3: (* типичная ошибка типа 2 *) VERM HINW3;

(* выдать пояснение HINW3 *)

4: (* типичная ошибка типа 3 *) VERM HINW5;

- (* выдать пояснение HINW5 *)
- 0: (* непредполагаемый ответ *) VERM HINW1;
- (* выдать сообщение HINW1 *)
- (* конец "внутреннего" цикла *)
- (* конец "внешнего" цикла *)

BIS 3 VON KAU, KAU, KAU RICHTIG ODER 3 ZYKLEN

(*, до тех пор, пока три задания из группы KAU не будут выполнены правильно, или "внешний" цикл не выполнится трижды *).

Идентификатор KAU.10 определяет группу из десяти вопросов в файле вопросов. Во внутреннем цикле обучаемому предлагается решить три случайным образом выбранных из них задания (оператор LOESE). Оператор FALLS обеспечивает выдачу соответствующего сообщения по результатам выполнения задания. Этот процесс продолжается до тех пор, пока все три задания не будут решены правильно, однако не более чем дважды (внешний цикл).

В.-Х.Хартвиг.

ЛИНГВИСТИЧЕСКАЯ ИГРА — игра, осуществляемая на языковом материале. Игровые элементы в Л.и. конструируются также благодаря "игре слов". При этом широко используются синонимия, омонимия, языковые парадоксы, юмор и т.д. Примером Л.и. является "Словесный покер".

Э.Л.Ивахненко, Е.Д.Маргулис.

ЛИНГВИСТИЧЕСКИЙ ПРОЦЕССОР — комплекс программных или программно-аппаратных средств, осуществляющих анализ/синтез фраз естественного языка. Задачей Л.п., функционирующего в рамках прикладной интеллектуальной системы, является перевод естественных языковых *текстов* на некоторые формализованные языки (внутреннее представление), конструкции которых должны выражать смысл этих текстов, и наоборот, по внутреннему представлению синтезировать естественные языковые тексты. Все остальные блоки прикладной интеллектуальной системы работают только с внутренним представлением. В каждом Л.п. можно выделить декларативную и процедурную части. На уровне декларативной части описываются, напр., *словари* Л.п., а на уровне процедурной — *алгоритмы* анализа/синтеза естественных языковых текстов. В соответствии с уровнем описания языка в Л.п. выделяются блоки *морфологического анализа, синтаксического анализа и семантического анализа*, каждый из которых имеет декларативную и процедурную части. Реализация любого Л.п. должна предполагать наличие формальных средств задания каждого его блока. Существует несколько подходов к реализации Л.п. При одном из них реализация Л.п. осуществляется средствами некоторого одного языка программирования или с

помощью спектра соответствующих *алгоритмических языков*. Ныне это самый распространенный подход к реализации Л.п. Второй подход предполагает использование существующих компонентов системы представления лингвистических *знаний* в качестве специального ЯП. Часто при построении Л.п. используется подход, при котором отдельные модули (напр., блоки морфологического, синтаксического или семантического анализа) программируются отдельно, а затем с помощью задания на генерацию собирается конкретная конфигурация лингвистического процессора.

Н.А.Власенко.

ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ — формальная модель *естественного языка ограниченного*. Различают внешнее и внутреннее Л. о. *информационных систем*. Внешнее Л.о. ориентировано на конечных *пользователей*, внутреннее — на программные средства систем. Внешнее Л.о. — это диалоговые средства, предоставляемые пользователю системой (*диалог* типа “меню”, диалог с использованием штампов или клише, диалог на ограниченном естественном языке). Л.о. задач обработки текстов состоит из языка представления содержания текстов и программных средств его обнаружения — *лингвистических процессоров*. В соответствии с современными концепциями лингвистики содержание текста понимается как фрагмент системы *знаний о предметной области*, так что наиболее важными компонентами Л.о. являются *язык представления знаний* и лингвистический процессор.

Н.А.Власенко.

ЛИНЕЙНАЯ ОБУЧАЮЩАЯ ПРОГРАММА — программа, состоящая из жесткой, единой для всех *обучаемых* последовательности шагов учебной *информации* (порций, доз), причем в ходе или в конце изучения информации шага выполняются контрольные задания, а порядок изучения шагов не зависит от правильности выполнения заданий. При реализации Л.о.п. единственным показателем индивидуализации является критерий скорости усвоения, способ же обучения по Л.о.п. — единый для всех обучаемых. См. также *Программированное обучение*.

Н.М.Розенберг.

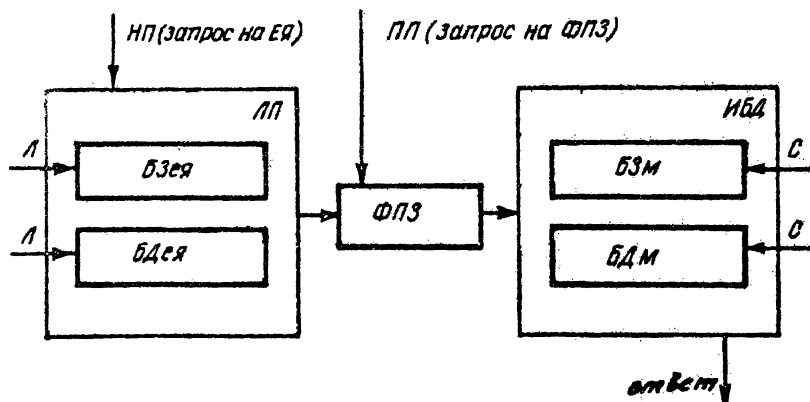
ЛИНИЯ ВЗГЛЯДА — условная линия, определяющая направление взгляда. Определяется взаимным расположением в пространстве экрана дисплея и тела оператора (в положении сидя). Наиболее удобно такое расположение экрана *дисплея*, когда взгляд оператора направлен вниз под углом 20° к горизонтальной плоскости, проходящей на уровне глаз (нормальная Л.в.). Плоскость экрана дисплея должна располагаться перпендикулярно Л.в. оператора. При

этом нужно выбирать такое положение дисплея, чтобы на его экране не возникали отражения от светильников и других источников света (см. *Световой поток*).

Л.в. тесно связана с расстоянием наблюдения, которое зависит от вида работы за дисплеем. Расстояние наблюдения должно находиться в пределах от 400 до 700 мм. Каждое рабочее место имеет свои конструктивные особенности в зависимости от назначения. В связи с этим необходима значительная гибкость в расположении элементов рабочего места относительно друг друга. Сам дисплей или стол, на котором он установлен, должны иметь конструкцию, позволяющую изменять горизонтальный и вертикальный углы наклона плоскости экрана в зависимости от вида работ.

С.А. Миронченко В.С. Пашковский.

ЛИПС (Лингвистический Процессор в информационных Системах) — инструментальный пакет прикладных программ для естественоязыкового общения с базами данных (БД), реализованный в среде Тролог. Состоит из двух частей (рис.): лингвистического процессора (ЛП), переводящего естественоязыковой запрос в



некоторое формальное представление запроса (ФПЗ); интеллектуального банка данных (ИБД), в котором по ФПЗ организуется поиск в БД, запрашиваемой информации. Как ЛП, так и ИБД состоят из двух баз: базы данных и базы знаний (БЗ). В случае ИБД, БДм — база данных реляционного типа, БЗм — правила манипулирования (поиска, суммирования, выдачи) данными и аксиомы, позволяющие получать информацию, не заданную в БД в явном виде. В случае

ЛП, БД_{ЕЯ} — словарь, а БЗ_{ЕЯ} — грамматика используемого языка и алгоритмы анализа.

С информационными системами, построенными на базе ЛИПС, могут работать различные категории специалистов: конечные (неподготовленные) пользователи (НП), обращающиеся к системе с запросом на естественном языке (ЕЯ) с целью получения информации, хранящейся в БДм; подготовленные пользователи (ПП), обращающиеся к системе на формальном языке (ФПЗ); лингвисты (Л), вносящие изменения в словарь и пополняющие грамматику новыми правилами; специалисты в области БД, вносящие изменения в БДм и пополняющие БЗм новыми аксиомами, с помощью которых осуществляется логический вывод информации, не заданной в БДм в явном виде. Специалисты в области БД работают в тесном контакте с лингвистами — при изменении БДм и появлении новых возможностей у системы соответствующие изменения должны быть внесены в БД_{ЕЯ} и БЗ_{ЕЯ}.

Н.А.Власенко.

ЛИСП (англ. LISt Processing language — язык обработки списков) — язык программирования функционального типа, ориентированный на обработку структурных данных. Широко используется для программирования задач искусственного интеллекта. В области обучения применяется для создания интеллектуальных и экспертных обучающих систем.

Типы данных в Л. представлены списками, состоящими из атомов, которые могут быть числовыми или цепочками символов. Во втором случае атом может быть идентификатором, которому сопоставлено некоторое значение (числовое или строковое). Имена базовых операций Л. также представлены в виде атомов. Программа на Л. построена как суперпозиция функций, имеющих форму списковых структур. Поэтому определение функции можно обрабатывать так же, как любую другую списковую структуру, либо выполнять как программу с помощью функций APPLY или EVAL.

Осн. видом передачи управления в Л. является рекурсивный вызов функции. Ветвление внутри функций производится с помощью функции COND, осуществляющей условный выбор следующей исполняемой функции. В связи с тем, что рекурсивный вызов функции для моделирования итераций влечет большие накладные расходы, реализованы функции, позволяющие написать оператор цикла, подобно тому, как это делается в процедурных языках. Интерфейс с пользователем осуществляется на уровне операций ввода—вывода.

В.А.Третьяк.

ЛОГИЧЕСКИЙ ЯЗЫК ОПИСАНИЯ КУРСОВ (ЛЯОК) — язык программирования декларативного типа для описания сценариев и программирования автоматизированных учебных курсов (АУК). Имеет средства для описания учебного материала и средства программирования обучающе-контролирующего диалога. Учебный материал АУК группируется по разделам и разбивается на страницы. Разделам должны быть присвоены заголовки. На этапе трансляции учебный материал собирается в теоретическую часть курса, содержание которой отражается в оглавлении. В процессе эксплуатации курса к разделам теоретической части возможно как прямое обращение пользователя по имени заголовка, так и программируемое автором курса обращение по ходу обучающе-контролирующего диалога. По усмотрению автора АУК теоретическая часть может отсутствовать. Обучающе-контролирующая часть АУК представляется заданиями, состоящими из групп или последовательностей вопросов. В заданиях с группами вопросов каждому обучаемому случайно выдается по одному вопросу из каждой группы, а в заданиях с последовательностями — одна последовательность. Правильные и предполагаемые неправильные ответы обучаемых описываются эталонами с использованием механизма сопоставления ответа с группой эталонов, предусмотренных в управляющей структуре вопроса. В описании эталонов используются логические связки, специальные знаки для игнорирования некоторых частей ответа; имеется возможность многократного анализа ответа по частям. Совпадение ответа с одним из эталонов определяет дальнейший ход диалога — повтор текущего вопроса или переход к следующему.

Важной особенностью языка является механизм задаваемых автором функциональных клавиш, с помощью которых обучаемый имеет возможность конструировать и редактировать сложные ответы. Реализация языка для компьютеров графическими средствами позволяет описывать рисунки. Предложения ЛЯОК имеют унифицированную структуру: ключевое слово, определяющее функцию предложения; имя предложения; ключевые параметры и их значения. Предложения могут содержать текст, который выдается на экран при выполнении данного предложения. Пример — фрагмент курса:

```
#ВОПРОС:В1,ЧИСЛО ПОВТОРОВ=3,МАКСОЦЕНКА=5,
НЕ ОТВЕТИЛ='РОЙНА НАЧАЛАСЬ 1 АВГУСТА 1914 ГОДА';
В КАКОМ ГОДУ НАЧАЛАСЬ ПЕРВАЯ МИРОВАЯ ВОЙНА?
#РЕАКЦИЯ:14!1914#,ОЦЕНКА=5,ПОВТОРИТЬ=N;
ДА. ПРИЧИНАМИ ВОЙНЫ ЯВИЛИСЬ...
#РЕАКЦИЯ:39!1939#,ОЦЕНКА=2,ПОВТ=Д;
НЕТ. ЭТО ГОД НАЧАЛА ВТОРОЙ МИРОВОЙ ВОЙНЫ.
#РЕАКЦИЯ:1:45!1901:1945#,ОЦЕНКА=1,ПОВТ=Д;
НЕПРАВИЛЬНО.
#РЕАКЦИЯ:~%#.#,ПОВТ=И;
ОТВЕЬТЕ ЧИСЛОМ.
```

#КОНЕЦ ВОПРОС:В1,ОЦЕНКА=1;
НЕ ПОНИМАЮ ВАС.

На ответ 14 или 1914 будет реакция: оценка 5, на экран выведется текст: "ДА. ПРИЧИНАМИ ВОЙНЫ ЯВИЛИСЬ ...". Ответам 39 или 1939 соответствует вторая реакция, а любому числу в интервалах 1—45 или 1901—1945 — третья: обучаемый должен повторить ответ. Эталон четвертой реакции (^%) означает "не число". В этом случае ответ игнорируется и выдается реплика: "ОТВЕТИТЕ ЧИСЛОМ". Для всех непредусмотренных ответов выполняется предложение :#КОНЕЦ ВОПРОС: оценка 1, реплика "НЕ ПОНИМАЮ ВАС", вопрос повторяется. Если обучаемый не ответит за три попытки, то выдается текст, указанный в параметре НЕ ОТВЕТИЛ. Программы на ЛЯОК обрабатываются *транслятором*, выдающим листинг с указанием синтаксических ошибок и генерирующим выходной код в операторах языка описания курсов или в макрокомандах ассемблера языка. Л. реализован на машинах типа ЕС, СМ, ДВК и КУВТ-86.

Р.Ковалюнас.

ЛОГИЧЕСКОЕ ОПИСАНИЕ предметной области, логическая спецификация — формализация *предметной области* (ПО) в рамках некоторого *исчисления логического*. Строится по следующей схеме. 1) Фиксируется некоторый логико-математический язык, содержащий средства для описания свойств ПО в виде формул. 2) На этом языке задается некоторое логическое исчисление. Формулы, выводимые в этом исчислении, определяют свойства ПО. Таким образом, аксиомы исчисления можно рассматривать как первичные знания о ПО, а доказательство теорем в исчислении — как получение новых знаний. Л.о. ПО является адекватным (полным), если используемое исчисление адекватно (полно) относительно рассматриваемой ПО. Иногда рассматривается другой способ Л.о. ПО: определяется интерпретация формул используемого языка в *модели предметной области* и теоремами ПО объявляются все формулы, истинные в этой модели. Если истинность формулы в модели ПО можно проверить конструктивно (с помощью некоторого *алгоритма*), то такой способ Л.о. ПО совершенно аналогичен предыдущему. Однако для большинства ПО конструктивная проверка истинности невозможна, поэтому для полноценного Л.о.ПО в этих случаях необходимо строить исчисления. Л.о.ПО по сравнению с другими методами описаний более абстрактно, т.к. не требует фиксировать представление объектов ПО и позволяет рассматривать различные реализации (модели) этих объектов, удовлетворяющие аксиомам. Одним из воплощений идеи Л.о.ПО в *программировании* является *Пролог*.

В.М.Антимиров.

ЛОГО (LOGO, от греч. *λόγος* — слово, смысл, идея) — 1) компьютерная система, предназначенная для обучения детей разного возраста; 2) используемый в этой системе язык программирования (ЯП); 3) “философия” обучения, т.е. система взглядов на сущность и процесс обучения, которая и послужила стимулом к разработке соответствующего языка программирования и обучающей системы. Созданию языка, системы и “философии” Л. положил начало учебный эксперимент. В лаборатории искусственного интеллекта Массачусетского технологического института С.Пейперт, специалист в области компьютерного обучения (в прошлом — сотрудник известного швейцарского психолога Ж.Пиаже) перенес свою концепцию обучения на компьютерные средства. Акцент был сделан на развитие интеллектуальных умений в процессе программирования, т.к., по мнению авторов проекта, оно выступает как наиболее благоприятная среда для усвоения фундаментальных понятий и идей (таких, как “именование понятий”, “планирование”, “разбиение задачи на подзадачи”, “обратная связь”, “итерация и рекурсия”). Поскольку традиционные ЯП слишком сложны для детей из-за перегруженности многочисленными кодами, синтаксическими правилами и др., овладение понятиями желательнее осуществлять в игровой форме.

Язык Л. — язык очень высокого уровня (см. *Уровень алгоритмического языка*), работающий с графическими образами и списками, во многом связанный с языком Лисп и весьма похожий на него (изучение Лисп нередко начинается с Л., поскольку Л. легок и доступен в обучении и был задуман как учебный язык). Л. был создан как язык четко определенный, с фиксированным словарем, синтаксическими командами высокого уровня и возможностью конструировать логические процедуры из элементов самого языка. Популярности Л. в качестве учебного языка способствовала его упрощенная, но удобная графическая интерпретация, так наз. черепашья графика; вследствие этого Л. быстро вошел в штатные языки многих компьютеров. В 1980, примерно через 15 лет после создания “полного” языка Л., был разработан его сокращенный вариант для учебных целей и бытовых компьютеров. Ныне имеются диалекты Л. для наиболее распространенных персональных электронных вычислительных машин, и в этом одна из причин популярности программ, написанных на Л.

Анализ процесса обучения и использования “среды Л.” (или, по выражению С.Пейперта, “микрокультуры Л.”) нельзя проводить в отрыве от общей философско-педагогической концепции, лежащей в основе Л. : отличие от традиционных обучающе-контролирующих систем, Л. создает условия для “дивергентного обучения”, которое не ограничивается одним или, в лучшем случае, несколькими заранее созданными сценариями, а может активно конструироваться самим обучаемым на основе его индивидуальных особенностей и пожеланий.

Философия Л., отражающая предлагаемый С.Пейпертом подход к компьютерному обучению, основана на идее использования компьютера как модели, которая может повлиять на наш способ мышления о самих себе. Два осн. идейных источника этой философии — результаты, полученные психологической школой Ж.Пиаже и исследования в области искусственного интеллекта. Особое внимание уделено реализации феномена “обучения по Пиаже” (“научение без обучения”), предполагающего усвоение ребенком значительного объема знаний без специально организованного обучения, путем погружения его в естественную среду (напр., усвоение родного языка, происходящее естественным путем, посредством общения). С.Пейперт считает, что его философия обучения должна стать основой прежде всего для преподавания матем. дисциплин: “в докомпьютерной” культуре возможности приобретения матем. знаний ограничены, а компьютерная среда Л. позволяет, напр., создать “Матлэнд” — страну математики, в которой ученик будет разговаривать с компьютером именно на матем. языке и, естественным образом “погружаясь” в микрокультуру Л., усваивать матем. понятия. Наряду с понятиями среды, “микрокультуры”, “культуры Л.”, “когнитивной микрокультуры среды Л.”, которые у С.Пейперта нередко выступают как синонимы, используется и понятие микромира, означающее ограниченную и хорошо определенную учебную среду.

Осн. объекты микрокультуры Л. — кибернетическое животное “черепаха” и “математически говорящее существо” (компьютер), управляющее им. Обучаемый в среде Л. не столько обучается сам, сколько обучает черепашку (“не компьютер программирует ребенка, а ребенок программирует компьютер”): в основе черепашьей геометрии лежат понятия компьютерной науки и, напр., усвоение геометрии осуществляется как усвоение понятий программирования. Соответственно и отношение к ошибкам в философии Л. особое: они рассматриваются как необходимые и полезный компонент учебного процесса; в процессе исправления ошибок обучаемый постепенно совершенствует свою программу, овладевая новыми знаниями; “поле самостоятельности” в Л. достаточно велико: ошибки не ;страняются сразу, обучаемый имеет возможность самостоятельно проверить свои результаты и внести коррективы.

“Черепаха” имеет определенный начальный уровень “образованности”: может перемещаться в разных направлениях на указанное число шагов и поворачиваться в заданный угол. Из этих базовых команд Л. составляются простейшие программы, которые становятся для “черепахи” новыми процедурами.

Напр., программы

а) КВАДРАТ — ЭТО

FORWARD 5

RIGHT 90

или

REPEAT 4 [FD RT 90]

END

б) КВАДРАТ — ЭТО

```
FORWARD 5  
RIGHT 90  
FORWARD 5  
RIGHT 90  
FORWARD 5  
RIGHT 90  
END
```

будут выполнять рисование квадрата со стороной 5. Для этого достаточно указать имя процедуры КВАДРАТ. Для рисования квадратов разного размера применяются процедуры с параметрами: программа

```
КВАДРАТ : X — ЭТО  
REPEAT 4 [FD: X RT 90]  
END
```

будет рисовать квадрат со стороной 15, если значение $X=15$. Для этого достаточно вызвать процедуру КВАДРАТ 15.

Осн. принципы философии Л.: большую часть жизненного опыта человек усваивает непосредственно из окружающей его среды; “обучение по Пиаже” значительно лучше традиционного школьного обучения; для “обучения по Пиаже” определенной дисциплине нужно создавать соответствующую среду Л., в которой ребенок, играя и “погружаясь” в эту среду, осваивает трудную при обычном обучении дисциплину. Идеи С.Пейперта, достигшие пика популярности в начале 80-х гг., в последнее время подверглись критическому переосмыслению, поскольку появились данные о том, что многолетние эксперименты с Л. для детей разного возраста не всегда позволяют достичь рекламируемых С.Пейпертом результатов. Однако и сегодня система, язык и философия обучения Л. считаются примером наиболее удачного приложения психолого-педагогических конструктов в области компьютерного обучения, чему в большой степени способствуют работы самого С.Пейперта и многолетние эксперименты с Л. на разных контингентах обучаемых. Как отметила психолог Е.Н.Брусенцова анализ “философии Л.” показывает, что она не может быть рассмотрена как целостная теория обучения, а является обширным, подчас эклектичным, комплексом представлений о развитии и обучении детей. Остается нерешенным ряд проблем практического применения Л., а из-за нетехнологичности теоретической позиции С.Пейперта неудачи практической реализации Л. могут объясняться как следствие неверного понимания идей. Нельзя оправдать и полную замену традиционного обучения “обучением по Пиаже”, более того, и сама система Л., и опыт ее применения указывают, что последовательно осуществлять декларируемый способ обучения не всегда возможно. Нет и определенных закономерностей построения сред обучения. Вместе с тем в целом достоинства языка и “микрокультура” Л. — один из наиболее удачных примеров применения компьютеров в учебных целях.

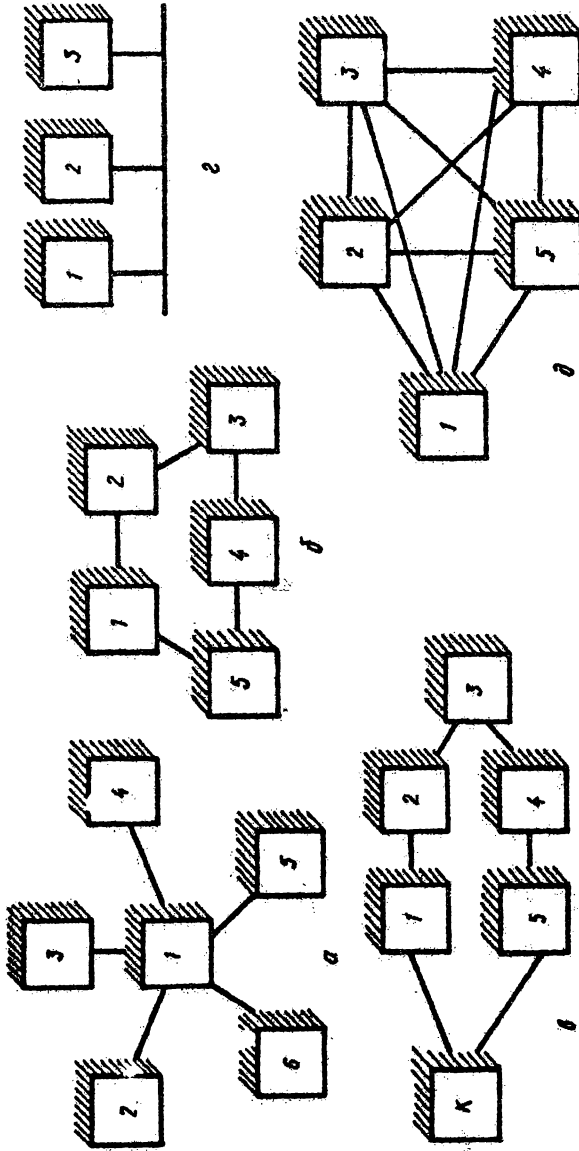
Е.Д.Маргулис.

ЛОКАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ — комплекс расположенных на ограниченной территории (здание, учреждение) аппаратно-программных средств, использующий ориентированные на эту территорию средства и методы передачи данных. Дает возможность однотипным и разнородным средствам вычислительной техники общаться друг с другом с помощью единой передающей среды. Связь может осуществляться между большими мини-ЭВМ и микро-ЭВМ, специализированными процессорами, персональными ЭВМ, терминалами, различным периферийным оборудованием, накопителями на магнитных дисках и лентах, а также специализированными средствами (регистрирующие и копирующие устройства, графопостроители, устройства связи с объектом). При этом Л.в.с. обеспечивает простое и удобное объединение всех средств. Главное отличие Л.в.с. от глобальных — наличие единого для всех абонентов высокоскоростного канала передачи данных.

К наиболее важным характеристикам Л.в.с. относятся структура (топология), тип передающей физической среды, скорость передачи сигналов, метод доступа станций Л.в.с. к среде, архитектура сети. В зависимости от размещения станций сети и способа соединения между ними структура Л.в.с. может быть звездообразной, кольцевой, петлевой, шинной и полносвязной (рис.). В Л.в.с. со звездообразной структурой центральным узлом служит ЭВМ или узел связи, осуществляющий коммутацию данных между станциями сети. Кольцевая структура Л.в.с. предполагает передачу информации от одной станции к другой, объединенным в кольцо, по однонаправленным каналам передачи данных. Петлевая структура Л.в.с. — разновидность кольцевой, где одна из станций является управляющей. Шинная структура Л.в.с. характеризуется использованием разомкнутого сегмента кабеля, к которому с некоторыми интервалами подключаются станции. Полносвязные структуры обеспечивают выбор наиболее дешевого маршрута между станциями и выгодны там, где усложнение логики управления окупается удешевлением связей.

По способу адресации и передачи данных все рассмотренные структуры Л.в.с. можно сгруппировать в два основных класса: ширковещательные и последовательные. В ширковещательных структурах каждая станция передает информацию, которая может быть воспринята всеми остальными станциями. В последовательных структурах каждая станция передает информацию только одной из станций.

Наиболее распространены Л.в.с. с шинной (магистральной) и кольцевой структурой, поскольку достаточно иметь единственный канал (моноканал), обслуживающий все системы сети в режиме мультиплексирования. Это обусловлено, во-первых, применением последовательного интерфейса как средства связи между используемыми системами, подключаемыми в локальную сеть, во-вторых, использованием в качестве физической среды передачи данных витых



пар проводников, коаксиального кабеля или волоконно-оптических линий, которые на небольших расстояниях обеспечивают высокую пропускную способность канала передачи данных. При использовании канала передачи данных в режиме мультиплексирования отпадает необходимость в маршрутизации сообщений, поскольку маршруты однозначно определены, т.е. отсутствует необходимость в сетевом уровне управления. Нет также надобности в *процедурах*, обеспечивающих эффективное использование ресурсов канала, и различных форматах сообщений.

Канальный уровень Л.в.с. делится на два подуровня: управления логическим звеном (УЛЗ) и управления доступом к среде (УДС). В функции подуровня УЛЗ входят адресация станций, передача кадров данных между станциями и управление протоколом данных. Станции в Л.в.с. связываются друг с другом через общий информационный канал. Поэтому возникают проблемы коллективного использования и разрешения конфликтов и столкновений в канале. Эта функция возлагается на подуровень УДС, который реализует *алгоритм* доступа к среде и адресацию станций.

Различают детерминированный (приоритетный), свободный и комбинированный доступ абонентов к каналу. В Л.в.с. с детерминированным доступом канал связи предоставляется поочередно всем абонентам в выделенные моменты времени. При этом распределение во времени может быть синхронным или асинхронным. В Л.в.с. со свободным доступом каждая станция захватывает канал для передачи данных в произвольный момент времени. В Л.в.с. с комбинированным доступом используется случайный и детерминированный доступ к каналу на разных фазах работы системы.

Физический уровень организации Л.в.с. обеспечивает сопряжение станции с физической средой, кодирование и декодирование сигналов, их буферизацию, поддерживает и восстанавливает битовую синхронизацию. Применительно к специфике средств передачи данных, используемых в Л.в.с. с моноканалом, сложность процедур на физическом, канальном, сетевом и транспортном уровнях невелика. Поэтому эти уровни удастся эффективно реализовать в основном техническими средствами — контроллером, сетевым адаптером.

Ряд особенностей позволяет выделить Л.в.с. в отдельный класс вычислительных сетей — размещение на сравнительно небольшой территории, простые методы передачи данных, низкий уровень ошибок, гибкая и легконастраиваемая структура, низкая стоимость сети передачи данных по сравнению со стоимостью подключаемых устройств.

Л.в.с. широко применяют в народном хозяйстве, они являются осн. техническим средством при оборудовании учебных классов и кабинетов информатики и вычислительной техники.

И.А.Емченко, К.Чакрыов.

ЛОКАЛЬНАЯ СЕТЬ УЧЕБНОГО НАЗНАЧЕНИЯ — локальная вычислительная сеть для связи компьютера преподавателя с компьютерами обучаемых с целью организации эффективной учебной работы. Используется для создания комплекса учебной вычислительной техники. Л.с.у.н. могут быть с расширенными и ограниченными возможностями. К Л.с.у.н. относятся *ВАРИЕЛ, МИКРОНЕТ-16, ФЕЛНЕТ*.

В.Н.Никулин, П.Горват.

МАГНИТНЫЙ ДИСК — магнитный носитель информации в форме диска. Различают М.д. жесткие и гибкие. Жесткий М.д. (ЖМД) представляет собой плоский диск из алюминиевого сплава толщиной около 1,5 мм, на обе стороны которого нанесен магнитный рабочий слой толщиной 1 — 3 мкм. В центре диска находится посадочное отверстие. Информация записывается на дорожках — замкнутых концентрических окружностях. Продольная плотность записи — 100 бит/мм, поперечная (радиальная) — около 10 дорожек/мм. Информационная плотность пакета из 12 ЖМД — до 100 М. Запись и воспроизведение в накопителях на ЖМД (см. *Винчестерский накопитель*) производятся бесконтактно. Расстояние между магнитной головкой и поверхностью диска (0,5 — 1 мкм) поддерживается воздушной подушкой, возникающей при высокой рабочей скорости диска (1500 — 6000 об/мин). Гибкий М.д. изготавливают из полиэтилентерефталата, с обеих сторон на него наносят магнитный рабочий слой толщиной 1—4 мкм. Информационная плотность — от 360 К до 1,2 М. Используют такой диск в *дискетах*.

В.Л.Леонтьев, В.И.Сигалов.

МАКРОАССЕМБЛЕР — транслятор с ассемблера языка в язык машинных команд, содержащий макропроцессор. Перед трансляцией макропроцессор выполняет подстановку последовательности операторов в текст программы макроопределения, задаваемую макрокомандами и их фактическими параметрами. Программа макроопределения может находиться в тексте исходной программы или библиотеке макроопределений.

В.А.Бардадым.

МАКРОВЫЗОВ — указание макропроцессору на необходимость замены одной последовательности элементов исходного текста на другую. Процесс замены наз. макроподстановкой, а результирующий текст — макрорасширением. М. состоит из имени или образца макроопределения и, возможно, списка фактических параметров.

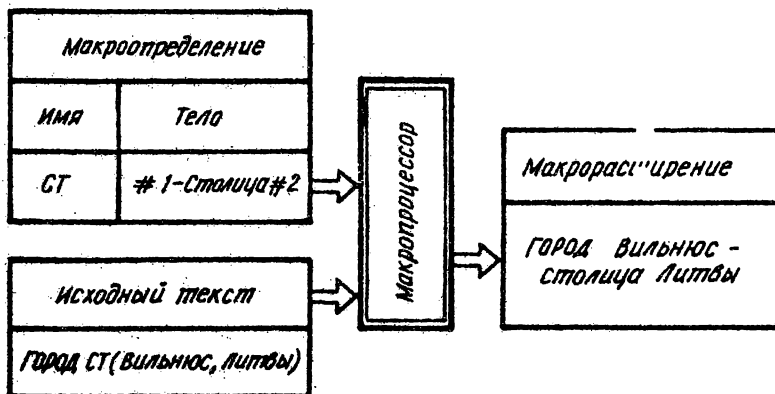
В.А.Петрушин.

МАКРОГЕНЕРАТОР — то же, что и *макропроцессор*.

МАКРОКОМАНДА (от греч. *μακρός* — большой, длинный и франц. *commande* - приказ) — 1) В языках ассемблера — оператор *макрывзова*, указывающий, что перед трансляцией необходимо вставить в текст программы макрорасширение. Ассемблер, содержащий *макропроцессор* для обработки М. и *макроопределений*, наз. *макроассемблером*. М. применяются для расширения набора операторов *ассемблера языка*, обеспечения краткой записи программ, их структуризации, настройки программ по параметрам, использования стандартных библиотек, рассчитанных на различные области применения. Стандартные наборы М. используются как специализированные языки программирования, 2) В диалоговых системах — средство, позволяющее задавать последовательности элементарных актов *диалога* и использовать эти последовательности как новые элементарные акты диалога.

В.А. Бурдадым.

МАКРООПРЕДЕЛЕНИЕ — программа на языке *макропроцессора*, задающая способ замены текста *макрывзова* на соответствующий ему текст макрорасширения. Самый простой вид М. содержит имя и тело — текст, на который заменяется имя при макрывзове. Более сложные М. содержат формальные параметры, которые заменяются на фактические параметры при макрывзове. Пример использования параметров приведен ниже (рис.). Здесь тело М. СТ содержит два формальных параметра — #1 и #2, которые при макрывзове СТ(ВИЛЬНЮС.ЛИТВЫ) заменяются на строки “ВИЛЬНЮС” и



“ЛИТВЫ” соответственно. Допуская *M.* и *макровызовы* внутри *M.* и в параметрах *макровызовов*, а также рекурсивные *M.*, можно придать *макросредствам* дополнительную гибкость (см. *GPM*).

Понятиям “*макроопределение*” и “*макровызовы*” близки понятия “*описание*” и “*вызовы подпрограммы*”. Однако в *языковых процессорах* компилирующего типа результатом *макровызова* является подстановка текста на некотором языке *программирования*, а результатом *вызова подпрограммы* — исполнение ее команд. Эта разница несущественна для *языковых процессоров* интерпретирующего типа, когда порожденный текст сразу же исполняется (напр., при интерпретации *каталогизированных процедур* языка управления заданиями ОС ЕС).

В.А.Петрушин.

МАКРОПРОЦЕССОР, *макрогенератор* — компонент *программного обеспечения ЭВМ (языковый процессор)*, позволяющий производить замену одной последовательности элементов входного текста на другую. Элементы текста могут быть *символами* или более сложными синтаксическими конструкциями (словами, командами и т.п.). На необходимость замены указывают посредством *макровызова*. *M.* различаются по: 1) выходному или базовому языку: если *M.* пригоден для одного базового языка, то он наз. специализированным (напр., *макроассемблеры* и языки управления заданиями в *операционной системе*), в противном случае — универсальным (напр. *GPM*); 2) синтаксису *макровызова*: различают *вызовы* с предупреждающим маркером и *вызовы* по образцу; 3) методу передачи параметров: по имени или по значению; 4) средствам периода генерации (языку *M.*): типам переменных, *операторов*, допустимости вложенных и рекурсивных *макровызовов*; 5) методам реализации. Первоначально *M.* появились как средство расширения *ассемблера языка*, но затем получили широкое распространение в качестве *препроцессоров* для машинно-независимых языков (*ПЛ/1, Си, Смолток-80* и др.). Осн. области применения *M.*: расширение языков, трансляция с одного языка на другой, генерация текстов, систематическое редактирование и поиск.

В.А.Петрушин.

МАКРОСРЕДСТВА в автоматизированных обучающих системах — программные средства, позволяющие автору писать часто повторяющиеся наборы *операторов* в форме *макроопределений*. Автор *автоматизированного учебного курса (АУК)* в процессе создания его может вызвать *макроопределение*, тем самым устраняя необходимость повторного описания этого набора *операторов*. В *автоматизированных обучающих системах с языками программирования* типа *Курсрайтер, Тьютор* и язык описания курсов при регистрации

АУК после метки END автоматически создаются метки MACROS и MACEND, определяющие зону макроопределений. В эту зону вставляются операторы макроопределений, и они могут генерироваться только в том курсе, в котором находятся. В М имеются следующие типы операторов и команд: операторы макроопределения, задающие основу для последовательности генерируемых операторов АУК; операторы макрокоманды, в которых задаются параметры, используемые при генерации вызываемого макроопределения; команды генерации, использующиеся для копирования в АУК материала из макроопределений после замены в них аргументов на параметры макрокоманды. Операторы макроопределения записываются в такой последовательности: 1) имя макроопределения — метка; 2) МА — оператор начала макроопределения; 3) операторы макроопределения следуют после оператора МА и являются операторами ЯОК; 4) ЕА — конец макроопределения. Операторы макрокоманды используются в курсе для вызова макроопределения. К ним относятся операторы: СМ — вызвать макроопределение и СС — продолжить вызов. Генерация макрорасширений может выполняться одним из двух способов: 1) автор вводит в АУК оператор СМ и СС и после последнего оператора СС вводит оператор GEN; выполняется генерация макрорасширения, на экран терминала автора выводится сообщение GENERATION COMPLETED (генерация закончена) и следующий номер оператора. Автор может продолжать вводить операторы после сгенерированного макрорасширения; 2) операторы СМ и СС вставляются в АУК, как было описано выше. Затем в любое время после сообщения TYPE COMMAND автор может ввести команду GENERATE.

Т.А.Каменева.

MALT (Machine Language Tutor — учитель машинного языка) — генерирующая автоматизированная обучающая система, предназначенная для обучения программированию на машинно-ориентированных языках. Способна автоматически генерировать задачи программирования, конструировать собственную программу решения, осуществлять контроль правильности решения обучаемого, указывать на логические ошибки, обеспечивать соответствующие уточняющие корректирующие воздействия. В значительной степени способна индивидуализировать обучение на основе детализации контроля и степени инструктажа каждого обучаемого по результатам его работы. Основываясь на модели обучаемого, система определяет уровень сложности обучения. Модель предметной области представлена в виде концептуального дерева, в котором каждый узел представляет осн. понятие (концепт) и соответствует информации, необходимой для работы проблемного генератора и решателя: понятие числа, понятие имени, уровень этого понятия, определяющий относительную

его сложность, имя проблемного генератора для этого понятия, имя определенного решения, список параметров, необходимых для решения задачи. Модель ПО включает 21 понятие и начинается с осн. логических операций и числовых превращений. Дуги концептуального дерева указывают на возможность получения из примитивов более сложных задач. Генерация программы-задания осуществляется проблемным генератором на основе запрограммированных в системе элементарных заданий, которые можно использовать многократно при составлении полной программы. Параметры задания генерируются случайным образом. Сложность генерируемого задания определяется уровнем подготовки пользователя. Задание предъявляется обучаемому на естественном языке (предварительно проверяется, не решал ли его обучаемый ранее) и параллельно в виде элементарных подзадач его составляющих поступает на решатель, автоматически решающий сгенерированную задачу. По ходу выполнения задания осуществляется постоянный контроль правильности выполнения заданий с целью указания возможных ошибок, оказания помощи.

В системе MALT используются два метода контроля правильности выполнения программ. Наиболее общий метод состоит в пошаговом анализе каждого сегмента программы (по мере его написания), что позволяет иметь немедленную обратную связь с обучаемым. Оперативная проверка предполагает, что система имеет подробную информацию о состоянии программы в любой момент времени. При получении решения обучаемого система сравнивает его с собственным сконструированным решением, которое она считает правильным. Если они не совпадают, определяется, возможны ли разные ответы. Если возможны, система конструирует их и вновь сравнивает с решением обучаемого. Если же система приходит к выводу, что полученный ответ все же является ошибочным, она оказывает обучаемому необходимую помощь — от небольших подсказок до предъявления ему лучшего из возможных вариантов программы. Если ответ обучаемого совпадает с к.-л. генерируемым системой ответом, то система воспринимает решение обучаемого как действительно альтернативу её собственному решению. Второй метод контроля основан на моделировании сегмента программы, в процессе которого система устанавливает, выполняется ли нормальное окончание программы. Ненормальное окончание программы вызывается такими условиями, как бесконечные циклы, неопределённые программы или ветви программы и др. Любое такого рода условие немедленно исправляется обучаемым, вновь устанавливается набор необходимых условий, и моделирование продолжается. Если любое конкретное конечное условие показывает, что программа пользователя не выполнила свою функцию правильно, MALT пытается исправить программу. Поскольку в системе известны точные результаты, которые должны быть получены, обучаемому предоставляется краткое

описание ошибки, однако изолировать место ошибки в программе пользователя система не может. Это должен сделать он сам. Если обучаемый не способен исправить свой сегмент программы, то М. генерирует для него свой правильный вариант сегмента программы. После проверки примитивного задания MALT повышает или понижает индикатор уровня знаний обучаемого, связанный с используемой программой понятий. Этот индикатор определяет степень контроля и инструктажа обучаемого, которые он должен получить при следующем обращении к данной программе понятий. Эти данные затем используются в процессе генерации задания и влияют на вероятность генерации конкретной подзадачи, являющейся, таким образом, обратной функцией уровня предыдущих успехов обучаемого по всем примитивным заданиям данной подзадачи.

Система MALT обеспечивает одновременную работу с 30 терминалами в системе разделения времени на IBM 360/65. При этом каждому терминалу доступны не более 12 страниц (48 байт), время реакции ответа обучаемого не более 10 с.

Л.Н.Гецко.

МАНИПУЛЯТОР РУЧНОЙ — устройство ввода информации, выполняющее двигательные функции аналогично движениям руки человека при перемещении объектов на экране дисплея. Позволяет осуществлять игровые операции (выстрелы, прыжки, удары и пр.). Иногда применяется как альтернатива клавиатуры. Наиболее распространены М.р. двух типов: джойстик и "мышь". Джойстик состоит из подставки, закрепляемой на поверхности стола, подвижного рычажка и кнопок. Наклон рычажка вызывает замыкание одного или двух из четырех контактов в подставке, что в свою очередь указывает одно из восьми возможных направлений движения. В аналоговых джойстиках наклон рычажка изменяет сопротивление потенциометров и, следовательно, величины напряжений сигналов, вводимых в ЭВМ. В этом случае угол наклона рычажка влияет на скорость перемещения объекта в заданном направлении. "Мышь" свободно перемещается по плоской поверхности, напр., стола. Перемещения вызывают вращение двух координатных датчиков, с каждого из которых в ЭВМ поступают импульсы. Количество импульсов пропорционально углу поворота датчика и, следовательно, величине линейного перемещения по соответствующей координате. М. р. используют в игровых и учебных персональных компьютерах.

А.Рафф, В.И.Сигалов.

МАТРИЦА ТОЧЕК — прямоугольная таблица определенного размера, элементами которой являются точки, каждая из которых находится на пересечении столбца и строки. Генерация символов с помощью М.т. — наиболее распространенный способ представления

текстовой информации на экранах видеодисплейных устройств, в которых используется принцип растрового сканирования. Каждый символ формируется с помощью высвечивания определенного числа точек, взятых из М.т. размером 5 x 7, 7 x 9, 9 x 11 или 11 x 14. Согласно большинству международных стандартов М.т. размером 7 x 9 считается мини.альной. Чем больше точек в матрице, тем вероятнее возможность получить наиболее разборчивый и четкий символ. Точки в М.т. должны быть круглыми или квадратными (что предпочтительнее), но не овальными. Расстояние между точками желательно минимизировать, чтобы контур символа воспринимался как неразрывный.

С.А.Миронченко, В.С.Пашковский.

МАШИННАЯ ГРАФИКА — область *программирования*, ориентированная на разработку *программного обеспечения*, предназначенного для создания, хранения и обработки *моделей* объектов и их изображений с помощью компьютера. Занимается синтезом изображений реально существующих или воображаемых объектов. Связанная с ней область обработки образов ориентирована на обратный процесс (анализ сцен), определяемый как восстановление дву- и/или трехмерных объектов по их изображениям. Применение М.г. дает возможность существенно обогатить содержание учебного процесса с применением компьютера, предоставляя мощные средства работы с графической *информацией* учебного назначения. Целью применения М.г. является реализация взаимодействия человека с компьютером посредством изображений. Изображение как средство общения, обеспечивая необходимую точность работы с компьютером, представляется более естественным и вследствие этого более эффективным для большинства людей, в частности тех, кто не обладает необходимым уровнем профессиональной подготовки в области программирования.

Различают “пассивную” и интерактивную М.г. “Пассивная” М.г. не допускает участия человека в процессе синтеза изображения. Интерактивная М.г. предоставляет в распоряжение *пользователя* средства динамического управления содержанием изображения, его формой, размерами и цветом на экране *дисплея*. Это управление осуществляется с помощью *клавиатуры*, джойстика (см. *Манипулятор ручной*), *светового пера*, мыши и других устройств.

М.г. в *автоматизированных обучающих системах* (АОС) используют для обеспечения синтеза изображений в ходе компьютеризованного учебного процесса. По сравнению с традиционными средствами изготовления и воспроизведения учебной графической информации (фотографией, кино, телевидением) М. г. АОС имеет дополнительные преимущества, среди которых — возможность визуализации абстрактных объектов и оперативность подготовки графиче-

ской информации. Применение М.г. в АОС открывает возможности для оптимального сочетания традиционной для АОС алфавитно-цифровой информации с графической. В результате существенно повышается дидактическая эффективность АОС в учебных заведениях разных уровней системы народного образования (школах, профессионально-технических училищах, техникумах, вузах). Для компьютерной технологии обучения особый интерес представляет интерактивная машинная графика.

В.И.Отенко, М.Тагарински.

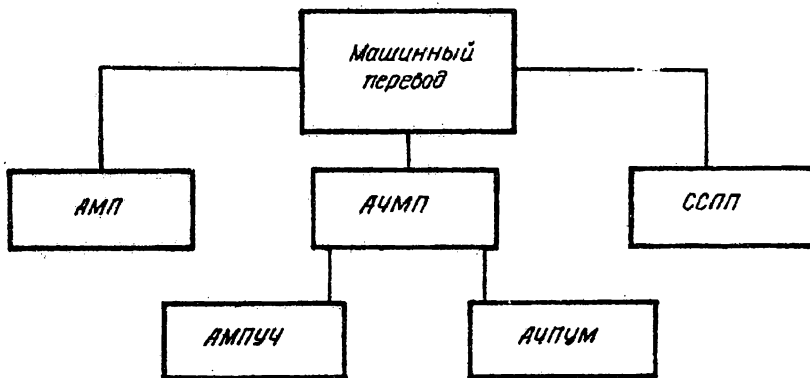
МАШИННО-ОРИЕНТИРОВАННЫЙ ЯЗЫК — алгоритмический язык, основу которого составляет машинный язык конкретной электронной вычислительной машины. Предназначен для решения задач системного программирования. По способу представления и обработки данных М.-о.я. делятся на императивные языки, функциональные языки и декларативные языки, по технологической направленности — на языки спецификаций, языки проектирования и языки программирования.

В.А.Третьяк.

МАШИННЫЙ ПЕРЕВОД — перевод текстов с одного естественного языка на другой с помощью средств вычислительной техники; область научных исследований, связанных с созданием систем М.п. в указанном выше смысле. Современные действующие систем М.п. (табл.) можно условно разбить на три группы (рис. на с. 313): системы автоматического (автономного) машинного перевода (АМП); системы автоматизированного человеко-машинного перевода (АЧМП); системы словарной поддержки переводчика (ССПП) или автономный терминологический банк данных (БД). Системы АМП осуществляют перевод без участия человека. Это, однако, не означает, что вводимый в компьютер текст не подвергается предварительной подготовке (предредактированию) со стороны человека, а выдаваемый компьютером результат не подвергается постредктированию. Эти системы (напр., *MULTIS*, *СИСТРАН*, *ПЛАСТ*) располагают мощными автоматическими словарями и более или менее развитыми системами морфологического анализа, семантического анализа и синтаксического анализа. Системы АЧМП опираются на достаточно большие словари, имеющие слабое грамматическое и семантическое обеспечение. Они подразделяются на системы М.п. с участием человека (АМПУЧ) и системы перевода, осуществляемого человеком с участием машины (АЧПУМ). Системы обоих подтипов дают возможность человеку осуществлять интерредактирование текста, т.е. вмешиваться в работу переводящего компьютера. При использовании АМПУЧ перевод осуществляется компьютером, но по ходу его

Основные действующие системы машинного перевода

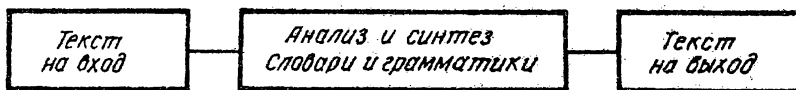
Название системы	Разработчик	Тип системы	Языковые пары	Тематика	Вид текста	Объем словарей для одной пары языков / ... /	Лингвистическая стратегия	Тип компьютера
MULTIS (СИ-ЛОД ПК)	ЛГТИ им. А. И. Герцена (г. Ленинград)	АМП	Многоязыковая	Вычислительная техника, публикацистика	Вычислительная документация, информационные сообщения	30	Трансфер	ЕС ЭВМ, IBM PC
ПЛАСТ	КПИ им. С. Лазо (г. Кишинев)	АМП	Английский — русский	Технология печатных плат	Патенты, статьи	25	Трансфер	ЕС ЭВМ
СИМ-ПАР	НИИ КП Нефтехим (г. Киев)	АМП		Нефтехимия, Термомодифика	Заголовки документов	5	Ориентация на трансфер	ЕС ЭВМ
СПАС	ВЦП (г. Москва), совместно с БНР	АЧМП	Многоязыковая	Вычислительная техника	Техническая документация	20		IBM PC ЕС-1832
АЛПС	США, Канада, ALPS	АЧМП	То же	То же	То же	10—300		
СИСТРАН	США, SYSTRAN	АМП	" "	" "	Научно-техническая документация	150—400	Прямой перевод	IBM-360, Siemens-7740
ATLAS II	Франция, Флизицу	АМП	Английский — японский		Техническая документация	300	Интерлингва	FACON M



осуществления компьютер может обращаться за помощью к человеку-редактору (см. АЛПС). Напротив, АЧПУМ используется чаще всего тогда, когда перевод осуществляется человеком, но имеют место ситуации возможного обращения за помощью к терминологическому БД. Эти системы (напр., СПАС) успешно применяются в рамках *автоматизированного рабочего места* переводчика. ССПП имеют наиболее простую организацию — представляют собой автоматический словарь, постоянно пополняющийся новыми терминами.

Наряду с приведенной структурно-функциональной классификацией можно выделить еще несколько подходов к оценкам и классификациям систем М.п. Напр., по количеству естественных языков, с которыми работает система, различают дву- и многоязыковые системы. Первые осуществляют перевод только для одной языковой пары и при этом лишь в одну сторону, но не наоборот. Если же система охватывает более одной языковой пары или осуществляет двусторонний перевод, она является многоязыковой. По целям, на которые может быть ориентирован перевод, различают М.п. научно-исследовательский; учебный, ориентированный на сбор информации; предусматривающий распространение информации. Последние два вида М.п. особо актуальны при решении внешнеторговых операций.

В действующих системах М.п. применяются следующие инженерно-лингвистические подходы: прямой способ М.п., использование межязыковых операций (трансфер) и перевод через язык-посредник (интерлигва). Стратегия прямого перевода (рис.) используется в системах, ориентированных на перевод с одного конкретного языка на другой, и заключается в том, что каждое предложение текста



переводится посредством последовательности стадий (обычно около 10), в которых выход из каждой стадии является входом в следующую. Эффективность таких систем зависит от качества словарей, глубины морфологического анализа текста на входном языке (ВхЯ) в серию возможных эквивалентных слов и словосочетаний на выходном языке (ВыхЯ). Примером системы со стратегией прямого перевода является американская система СИСТРАН. Стратегия межъязыковых операций (трансфера) заключается в том, что текст на ВхЯ сначала переводится в абстрактное внутреннее представление, а затем с помощью правил трансфера преобразуется на лексическом и структурном уровнях. На третьем уровне осуществляется генерация переводного эквивалента (см. рис. к ст. MULTIS). Для трансфера необходимо наличие трех словарей: словника ВхЯ, двуязычного словаря трансфера и словника ВыхЯ. Уровень трансфера изменяется от системы к системе, а представление варьируется от синтаксических маркеров глубинных структур до синтаксико-семантических аннотированных деревьев. Кроме того, стадия трансфера включает двуязычный компонент, что при создании многоязыковых систем заметно усложняет задачу, поскольку трансфер необходимо описывать для каждой пары языков. При трансферном подходе устраняются лишь те неоднозначности, которые затрудняют выбор переводных соответствий ВыхЯ. Перевод через язык-посредник (стратегия интерлигва) существует только в теории. Здесь представление смысла входного текста предполагается независимым от к.-л. естественного языка и непосредственно используется для синтеза выходного текста. Попытки применить некоторые идеи языка-посредника обнаруживаются в системе *ATLAS II*.

Системы М.п. различаются также в зависимости от вида текстов, на которые они ориентированы. Ныне существуют системы, работающие только с заголовками научно-технической документации (СИМПАР), с техническими спецификациями (отдельные версии СИЛОД), с описаниями изображений (ПЛАСТ). Существуют также универсальные системы, обеспечивающие перевод технических текстов разных типов (СИСТРАН, MULTIS). Для М.п. художественных текстов системы еще не разработаны.

К.Р.Пиотровская.

MENO-II — интеллектуальная обучающая система, разработанная в Массачусетском и Йельском университетах (США) на основе системы *MENO-TUTOR* и предназначенная для помощи программистам-новичкам в изучении языка Паскаль. Основные цели системы: а) обнаружение семантических и прагматических ошибок в простых программах; б) выявление неправильных представлений у обучаемого, приведших к ошибкам в программах; в) обучение правильным представлениям о предметной области на основе обнаруженных ошибок.

Система обнаруживает 18 ошибок в операторах присваивания и цикла, а также предлагает их объяснение на основе выявленных неправильных представлений. При этом различаются семантические ошибки, связанные с непониманием смысла конструкций языка, и прагматические ошибки, связанные с неверной трактовкой решаемой задачи. С целью обнаружения прагматических ошибок для каждой учебной задачи программист-эксперт составляет спецификацию в виде последовательности планов решения подзадач. Спецификация представляет собой базу знаний (БЗ) по решению данной задачи. Процесс диагностики состоит из двух этапов. На первом этапе программа обучаемого проходит синтаксический анализ, затем обобщается и в терминах планов сопоставляется со спецификацией задачи; если сопоставление неудачно, то фиксируются ошибки в операторах программы. На втором этапе на основании ошибок происходит поиск неправильных представлений; при этом используются следующие четыре БЗ: знания эксперта по программированию; сеть (каталог) ошибок в операторах программы; сеть ошибочных представлений о языке программирования и задаче; модель обучаемого, включающая знания о знаниях обучаемого и историю его обучения.

Эксперименты с системой показали, что текущая БЗ по ошибкам в Паскаль-программах охватывает менее 30% возможных типов ошибок, а достоверность диагностики обнаруженных ошибок составляет около 55%. По мнению авторов системы, неудачи MENO-II заключаются в том, что представление ошибок в фиксированном контекстно-независимом виде не позволило преодолеть изменчивость и многообразие реальных программ даже для небольших задач. Приемник системы MENO-II система PROUST имеет более мощный язык спецификаций и решает задачу диагностики на основе метода управления предсказанием, пытаясь подобрать наилучшую интерпретацию диагностируемой программы.

В.А.Петрушин.

MENO-TUTOR — экспериментальная интеллектуальная обучающая система, разработанная в Массачусетском университете (США) в начале 80-х гг. и послужившая основой для систем MENO-II и

PROUST. Специально разрабатывалась как полигон для опробования методик обучения в различных предметных областях (ПО).

Подсистема управления *учением* представляет собой трёхуровневую структуру, наз. сетью управления взаимодействием (СУВ). Верхний уровень составляют педагогические состояния: введение, обучение, диагностика и завершение. Педагогическим состоянием подчинены стратегические состояния (их всего 9), которым подчинены тактические состояния (27). В целом все состояния охватывают наиболее общие понятия процесса обучения. Каждая конкретная методика обучения задаётся в СУВ путём установления связей между состояниями (связи по умолчанию) и разработкой метаправил, каждое из которых срабатывает при выполнении некоторого условия, зависящего от ПО, модели обучаемого и состояния взаимодействия, и переводит систему в новое состояние. В системе различаются стратегические и тактические метаправила. Пример метаправила:

S1 — стратегическое метаправило

Из учить—данным.

В проверить—понимание.

О п и с а н и е: Переводит систему в режим выдачи ряда общих вопросов на различные темы.

А к т и в и з а ц и я: Текущая тема завершена, и система не уверена в высоком качестве знаний обучаемого.

П о в е д е н и е: Генерирует объяснение перехода от детальной проверки одной темы к поверхностной проверке различных тем в рамках знаний обучаемого.

Каждое состояние СУВ реализовано как список со слотами для функций (*фрейм*), которые выполняются при обработке данного состояния. Слоты определяют такие детали, как спецификация выводимого текста, следующее состояние перехода, обновление модели обучаемого и модели взаимодействия. СУВ обходит итеративная программа, передающая управление согласно связям по умолчанию, если перед обработкой очередного состояния не сработало к.-л. метаправило. Эксперименты с системой проводились в двух ПО: для обучения сложным физическим явлениям типа образования осадков и обучения программированию на языке Паскаль.

В.А.Петрушин.

МЕНЮ (от лат. *minutus* — подробное описание данных по пунктам) — список (набор, перечень, каталог) элементов данных, вариантов команд, ответов, изображаемых на экране дисплея. Функционально предъявление М. на экране предполагает выбор пользователем необходимого или желаемого варианта. Такой выбор из предлагаемого М. осуществляется путем ввода номера, буквы или указаний на соответствующий пункт М. курсором. При организации и выборе его содержания, а также расположения различных М. на

экране дисплея важно учитывать психологические и эргономические требования к организации выводимой на экран осн. и вспомогательной информации, чтобы уменьшить количество ошибок пользователя. Определение состава терминов в М., выбор числа вариантов в каждом М. и последовательность предъявления различных М. должны тщательно планироваться. Желательно предусматривать простой выход из последовательности М., возможность возврата к предыдущим М., а также зоны, содержащие пояснения.

Н.И.Повакель.

МЕТАЗНАНИЯ (от греч. *μετά* — между, после, через) — знания о знаниях. Появляются на этапе формирования *базы знаний* (БЗ), когда эксперт сообщает знания не о задаче, как иковой, а знания о знаниях, уже заключенных в БЗ в виде, напр., таких правил: 1. В режиме КОНТРОЛЬ правила, касающиеся режима КОНТРОЛЬ, имеют преимущества над правилами режима ПОМОЩЬ. 2. Правила, описывающие законы всемирного тяготения, если и будут видоизменяться, то в весьма редких случаях. 3. Правило КОНТРОЛЬ-ТЕМА-5 работает только потому, что существует правило ОБУЧЕНИЕ-ТЕМА-5. Так, первое метаправило дает некоторое стратегическое руководство для выбора правил и приводит к улучшению работы систем, основанных на представлениях знаний, благодаря сужению пространства поиска решений. Третье метаправило содержит обоснование правила КОНТРОЛЬ-ТЕМА-5, объясняя его происхождение, случаи его реального применения. Знания метауровня так же, как и знания, связанные с предметной областью (ПО), необходимо извлечь у эксперта и внести в БЗ компьютерной системы.

Большая часть М. происходит от недостаточного понимания самого явления. Напр., М. могут “появиться” в таком процессе. Эксперт сообщает совокупность фактов, теорем, уравнений, операций, которые составляют теорию “нулевого” порядка для рассматриваемой ПО, и, по мнению эксперта, носят исчерпывающий характер. Убедившись в неработоспособности системы, воплощающей эти знания, эксперт сообщает дополнительные сведения. Эти знания несут в значительной степени эвристический характер: правила, известные из практики, противоречивые советы, неточные критерии оценки и т.п. Эти дополнительные знания можно характеризовать как “поправку первого порядка” к фактической теории нулевого порядка. Когда даже эти знания не приводят к результатам, на которые способен эксперт, появляется третий тип знаний — М. типа: правило X непригодно в случае выполнения условия A; правила относительно A не являются абсолютно надежными (вводит коэффициент уверенности) и т.д. М. показанных типов явно или неявно используются в большинстве систем, основанных на представлениях знаний. М. в таких системах находят ряд применений: для поиска

и отбора правил; для обеспечения информации о правилах и знаниях, связанных с ПО; для оправдывания правил и тем самым для усиления способностей системы давать объяснения своих действий; для помощи в обнаружении ошибок в форме только что введенных правил и тем самым для помощи в расширении и модификации БЗ; для обеспечения ввода в систему новых знаний посредством демонстрации структуры имеющихся знаний всех типов и т.д.

Примером практического использования метазнаний для пополнения БЗ системы новыми знаниями может служить система EURISCO. Разработчики наделяют ее некоторым подобием “самосознания”, выраженного метаправилами. Эта система затрачивает много времени на управление своим поведением, запоминает обнаруженные ею правила и применяет их к самой себе. Среди открытых ею результатов был и такой: EURISCO обнаружила, что правила, поставляемые человеком, обычно оказываются лучше ее собственных, поэтому она создала эвристику:

ЕСЛИ правило создано системой

ТО вычеркни его

К счастью, первым правилом, которое оказалось стертым системой, было именно это правило.

А.А.Сахно.

МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ — формальные системы, с помощью которых описывают словесные знания. Делятся на декларативные (см. *Декларативное представление знаний*) и процедурные (см. *Процедурное представление знаний*). Для разработки баз знаний часто используют всевозможные сочетания различных М. п.з., напр., суперпозицию представлений, сочетание в гибридном представлении *семантических сетей, фреймов* и *продукций* (см. *Продукционные методы представления знаний*). Чтобы решить задачу на уровне “человек—решатель”, необходимо обладать двумя типами знаний. Первый тип включает *информацию о взаимосвязях моделей, категориях, аналогиях, иерархиях, объектах и отношениях между ними, т.е. представлениях структур и функций*. Второй тип знаний — это эмпирические ассоциации, возникающие при анализе структур и функций. Первый тип знаний воплощается в фреймы и семантические сети, второй, как правило, — в *продукции*.

В.В.Колос.

МИКРОКЛИМАТ кабинета автоматизированного обучения — комплекс физических факторов окружающей среды в ограниченном пространстве *кабинета автоматизированного обучения (КАО)*, оказывающий влияние на функции организма *обучаемого*. Определяется осн. физическими параметрами: температурой, скоростью движения и влажностью воздуха, температурой окружающих повер-

хностей, лучистой энергией осветительных приборов, физическими полями видеотерминалов на электроно-лучевых трубках. Предполагается, что температура воздуха в КАО должна составлять 20—25°С, скорость движения воздуха — 0,2 м/с, относительная влажность воздуха — 40—60%. Большое влияние на формирование температурно-влажностного режима в КАО оказывает их ориентация. Благоприятна для таких кабинетов ориентация в северном направлении.

Видеотерминалы являются источником шума в осн. частотном спектре 16—40 Гц. Уровни шума в этой частоте составляли, по данным ряда авторов, в среднем 53 дБ. Такой шум может вызывать сильную утомляемость обучаемых. Меры борьбы с ними: отделка стен шумозащитным материалом, а также строгий контроль за исправностью используемой вычислительной техники. Низкая относительная влажность может привести к повышению напряженности электростатических полей, продуцируемых работающими компьютерами.

Для поддержания необходимого М. применяется система вентиляции, сочетающая средства естественной и искусственной вентиляции. Все кабинеты должны быть оборудованы каналами с приточно-вытяжной вентиляцией. К искусственным вентиляционным средствам можно отнести бытовые кондиционеры. При площади кабинета 66 м² и высоте потолков около 3 м необходимо ставить до 3 кондиционеров. В воздухе КАО содержатся тяжелые ионы и пылевые аэрозоли. Вследствие наличия электростатических полей пыль интенсивно оседает на поверхность экрана и корпус монитора. Поэтому после каждого урока с применением видеотерминалов необходима влажная уборка помещения.

А.С.Коваленко.

МИКРОКОМПЬЮТЕР — то же, что и микро-ЭВМ.

МИКРОЛАБ — комплекс технических средств и программного обеспечения, ориентированный на проведение автоматизированных лабораторных занятий и экспериментов. Дает возможность осуществлять сбор первичных данных от объекта, их обработку и сохранение на индивидуальной *дискете*, распечатку данных на *печатающем устройстве*, а также использовать индивидуальные расписания проведения лабораторных упражнений и индивидуализировать лабораторные занятия на основе гибкой обратной связи с обучаемым. Комплекс разработан в Высшем машинно-электротехническом институте имени В.И.Ленина (Болгария). Основу технических средств М. составляют ПЭВМ "Правец 32", контроллер и лабораторный макет. Корпус лабораторного макета плоский по форме, конструктивно разделен на постоянную и сменную части. Постоянная часть представляет собой универсальную аналого-цифро-

вую периферию, сменная часть — исследуемый объект. Связь между двумя частями лабораторного макета осуществляется коммутационными проводами. На лицевую поверхность макета нанесена цветная схема, отражающая его содержание. Конструктивное исполнение лабораторного макета позволяет легко исследовать аналоговые, цифровые и гибридные объекты.

М. Тагарински.

МИКРОНЕТ-16 — локальная сеть учебного назначения на базе персональных электронных вычислительных машин Правец-16 и IBM PC/XT/AT. Дает возможность реализовать: централизованную загрузку и запуск программ учебного назначения; оперативную связь между преподавателем и обучаемым; контроль процесса обучения; оперативное вмешательство в ход работы обучаемых; проведение контроля усвоения знаний; сбор и обработку статистической информации; коллективное использование периферийных устройств локальной сети. М.-16 разработан в Высшем машинно-электротехническом институте имени В.И.Ленина (Болгария). Общее число абонентов М.-16 может достигать 255, структура — шинная, множественный доступ абонентов к каналу связи с обнаружением несущей и детерминированным разрешением конфликтов. Скорость обмена 1Мбит/с, соединительная среда — коаксиальный кабель или витая пара, длина общей шины может достигать 1 км. Сетевое программное обеспечение совместимо с операционной системой MS DOS 3.10, обеспечивает считывание и запись содержимого оперативного запоминающего элемента любого абонента, а также содержимого видеопамяти любой ПЭВМ в составе сети, осуществление дисковых операций с накопителем на жестких и/или гибких магнитных дисках каждого абонента.

М. Тагарински.

МИКРОПРОЦЕССОР — алгоритмически универсальное программно управляемое устройство, выполненное на микросхемах (одной или нескольких) большой степени интеграции и осуществляющее прием, обработку и выдачу цифровой информации. В зависимости от количества микросхем различают М. одно- и многокристалльные. Однокристалльными являются М. с фиксированной разрядностью информационного слова и системой команд. Наиболее распространены 8- и 16-разрядные однокристалльные М., развивается выпуск 32-разрядных однокристалльных М. для использования в персональных ЭВМ. Многокристалльные М. строятся на основе микропроцессорных секций ограниченной разрядности (2, 4 или 8 бит), допускающих сопряжение друг с другом. Для нормального функционирования М. должен быть сопряжен с памятью ЭВМ и периферийными устройствами.

В. И. Сигалов.

МИКРО-ЭВМ, микрокомпьютер — *электронная вычислительная машина*, выполненная на основе микропроцессора. Отличается малыми габаритами и невысокой стоимостью, ограниченными возможностями для подключения дополнительного оборудования.

Кроме микропроцессора, в состав М.-ЭВМ входит *запоминающее устройство* для хранения программ и данных и аппаратурная часть интерфейса периферийного оборудования. Все составляющие могут быть размещены на одной плате с печатными электрическими соединениями — М.-ЭВМ одноплатная. Есть М.-ЭВМ, выполненные на нескольких печатных платах, а также однокристалльные М.-ЭВМ в виде большой интегральной схемы. М.-ЭВМ, выполненные на нескольких платах, характеризуются наибольшей гибкостью: можно варьировать и расширять набор устройств ввода — вывода информации, наращивать память ЭВМ, подключить вспомогательные процессоры. Однокристалльная М.-ЭВМ наименее гибка, но компактна, надежна, низкой стоимости. Особенность одноплатных М.-ЭВМ — реализация интерфейса периферийных устройств на основе программируемых интерфейсных микросхем. Важное направление развития М.-ЭВМ — *персональные электронные вычислительные машины*.

В.П.Семихов, В.И.Сигалов.

МИНИКОМПЬЮТЕР — то же, что и *мини-ЭВМ*.

МИНИ-ЭВМ, микрокомпьютер — *электронная вычислительная машина* массового использования, для которой характерны высокая надежность при среднем диапазоне производительности, умеренная стоимость, укороченная длина слов, развитая система прерываний, развитое программное обеспечение и сравнительно малые габариты. Рост степени интеграции, повышение быстродействия элементарной базы, развитие архитектуры ЭВМ привели к смещению границ между отдельными классами ЭВМ. Современные М.-ЭВМ по своей производительности сравнимы со средними и большими ЭВМ.

Отличительная черта М.-ЭВМ — простота создания проблемно-ориентированных комплексов. Проблемная ориентация М.-ЭВМ осуществляется схемными и микропрограммными способами. Схемный способ предполагает макс. соответствие архитектуры и структуры М.-ЭВМ заданному классу задач. Такой подход, сильно ограничивая универсальность машины, позволяет получить высокую производительность. Микропрограммный способ дает меньшую производительность М.-ЭВМ, чем схемный, но позволяет существенно расширить класс эффективно решаемых задач. Широкие функциональные возможности М.-ЭВМ, снижение требований, предъявляемых к квалификации пользователей и уровню обслуживания, набор

периферийных устройств значительно расширяют спектр областей применения.

М.-ЭВМ используют для решения задач комплексной автоматизации управления сложными технологическими объектами, научными исследованиями, процессами обучения, обработки информации. Ими оснащают кабинеты информатики и вычислительной техники и учебные классы.

И.А.Емченко.

МИОПИЯ (от греч. *μῦωψ* (*μύωψ*) — близорукий), близорукость — аномалия рефракции, при которой попадающие в глаз параллельные лучи сливаются перед сетчаткой, образуя неотчетливый образ. Различают аксиальную М. (вследствие удлинения оптической оси глаза) и М. вследствие изменения кривизны преломляющих поверхностей глаза. В развитии М. большую роль играет наследственность. М. бывает наследственной и приобретенной. Различают три степени М.: слабую (до 3,0 диоптрий), среднюю (от 3,0 до 6,0 диоптрий) и высокую (выше 6,0 диоптрий).

Наибольшую опасность представляют врожденная М. и М., приобретенная в детском возрасте, которые, как правило, носят прогрессирующий характер вследствие роста глаза и приводят к целому ряду осложнений. В настоящее время нет экспериментально подтвержденных сведений, что работа с видеотерминалом является причиной развития и прогрессирования М. Однако известно, что М. может послужить фактором развития *астенопии* при неправильном режиме работы за экраном.

А.С.Коваленко, В.Г.Мартирсова.

МНОГООКОННЫЙ ИНТЕРФЕЙС — диалоговое взаимодействие, при котором отдельным функциональным программным блокам отводятся прямоугольные области (окна) на экране дисплея. Многооконный способ представления информации на экране дисплея предполагает наличие нескольких зон предъявления информации, отделенных друг от друга: в пространстве, цветом, формой и/или другими способами представления информации. При организации окон на экране дисплея не рекомендуется использовать краевые зоны экрана, а также применять в большом количестве эффекты, привлекающие внимание пользователя (мелькания, повышенную яркость, обратный контраст и др.). К таким эффектам следует прибегать лишь в особых, психологически оправданных ситуациях. При М.и. р. омеандуют четко идентифицированные зоны размещения вспомогательной информации на экране дисплея: зона подсказок, зона комментариев, зона управляющих сообщений, зона для сообщений об ошибках. Вопросно-ответные сообщения и подсказки желательно помещать в верхней части экрана, явно выделяя

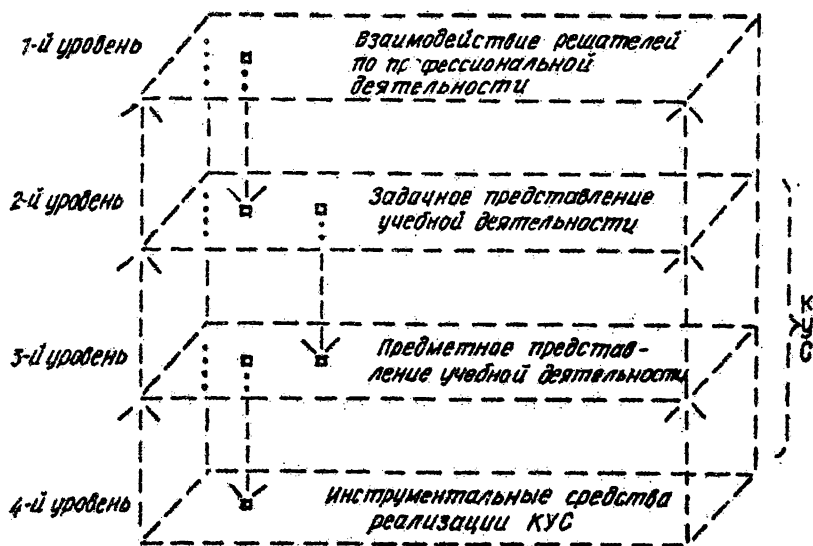
отведенную для этого зону, напр., отделяя ее горизонтальной линией от осн. информации на экране.

При использовании *цветового кодирования изображения* в организации М.и. следует руководствоваться рядом рекомендаций: необходимо минимизировать количество цветов, используемых на экране одновременно; цвет переднего плана и цвет фона должны находиться в сочетании между собой и с цветом *символов*, которые должны быть выделены на экране. Исходя из этого не рекомендуется использовать яркие цвета для границ окон и для заглавий; окна следует разделять между собой цветом фона; при *кодировании окон* нужно учитывать три функции цвета: выделение, задний план и нормальное чтение; выделяемые объекты, напр. сообщения об ошибках, рекомендуется окрашивать в цвета, используемые для предостережения. При обработке *текста* для кодирования информации, предъявляемой *пользователю* непрерывно, но необходимой ему только изредка, должны применяться нейтральные цвета.

В.М.Бондаровская, Н.И.Поякель.

МНОГОУРОВНЕВАЯ МОДЕЛЬ ОБУЧАЮЩЕГО ДИАЛОГА — модель, которая строится на основе выделения основных классов целей, преследуемых *обучаемым*, рассматриваемым как *решатель*. Многоуровневость задается иерархически упорядоченными классами целей, которые определяются типами *задач*, решаемых в *компьютерной учебной среде* (КУС). Рассмотрим следующие четыре класса целей (рис.): Ц1: обеспечить взаимодействие решателей по профессиональной деятельности *пользователя*; Ц2: определить структуру *учебной деятельности* пользователя на основе описания *учебных задач*; Ц3: реализовать объектное описание предметного представления учебной деятельности пользователя и реализовать детализацию учебной деятельности; Ц4: выделить и/или разработать *инструментальные средства* построения и реализации КУС.

Первый уровень определяется профессиональными целями пользователя. Для их реализации в него включаются диалоговые программы, обеспечивающие решение профессиональных задач пользователя. Второй уровень задается целями, определяющими вид учебной деятельности пользователя. На этот уровень переходят при слабой подготовленности пользователя к использованию в своей профессиональной деятельности ресурсов КУС. Цели реализуются за счет включения в уровень описаний учебных задач с соответствующими диалоговыми программами. Третий уровень задается целями, ориентированными на изучение структуры *предметной области* (ПО) задач пользователя. Его описание включает набор моделей объектов ПО с соответствующими программами доступа. Четвертый уровень определяется целями поддержки *диалога* с пользователем. Его реализуют в среде специальных пакетов прикладных программ,



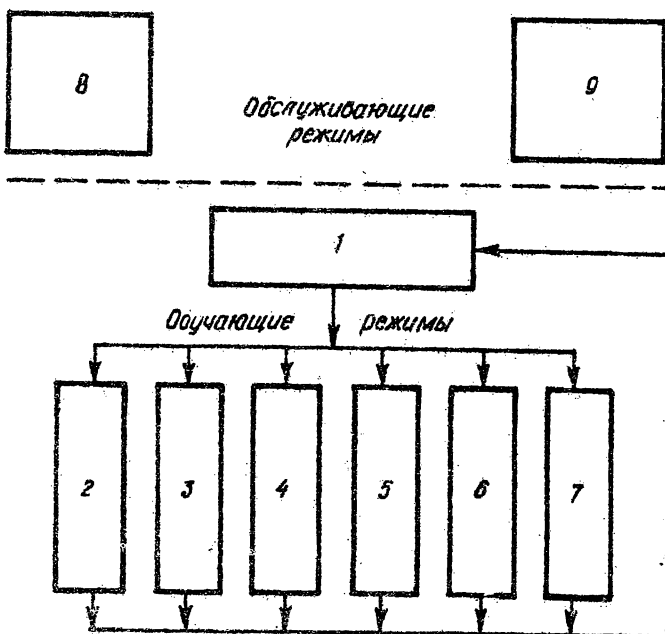
обеспечивающих адекватное представление моделей объектов, включение их в описание учебных задач и средства реализации диалога на первом уровне. Следовательно, I—III уровни определяют структуру и направленность учебной деятельности обучаемых — от решения прикладной задачи к изучению предмета задачи. Уровни II—IV определяют архитектуру КУС и направленность реализации КУС — от средств реализации моделей объектов предмета задачи к учебным задачам и сопоставления их с профессиональными задачами обучаемого.

А.Е.Стрижак.

МНОГОФУНКЦИОНАЛЬНЫЙ АВТОМАТИЗИРОВАННЫЙ УЧЕБНЫЙ КУРС (МФ АУК) — автоматизированный учебный курс (АУК) для обучения программированию, обладающий свойствами полноты предметного содержания, полноты методов обучения программированию и расширяемостью по языкам программирования (Я.п.) изучаемых в курсе. Свойство полноты предметного содержания МФ АУК обеспечивает сопровождение пользователя на всех этапах обучения и позволяет осуществить постепенное повышение уровня программистской квалификации в процессе работы с курсом. Предметное содержание МФ АУК определяется необходимыми этапами обучения программистов и связанные с этими этапами дидактическими целями: усвоением осн. принципов обработки

информации, реализуемых архитектурными особенностями компьютера; усвоением способов построения алгоритмов для решения различных классов задач; изучением Я.П, формированием умений и навыков по описанию алгоритмов на этом языке; формированием навыков работы с командным языком операционной системы, в среде которой выполняются программы; освоением процесса отладки программы. Свойство полноты методов обучения предоставляет возможность использования различных подходов к методике обучения программистов — языкового и алгоритмического. Суть языкового подхода состоит в том, что круг целей обучения концентрируется на изучении конструкций и свойств Я.П, однако этап обучения по выработке практических навыков программирования на изученном языке не сопровождается АУК. Языковой подход реализуется общепринятыми методиками программированного обучения, усиленными широкими возможностями быстродействующих компьютеров. Алгоритмический подход представляет собой концепцию, на основе которой на первый план выступает обучение способом конструирования алгоритмов решения задач, а язык изучается в той мере, насколько это необходимо для программирования составленных алгоритмов. В системах, использующих алгоритмический подход, предусматривается обучение конструированию и описанию алгоритмов и управление со стороны компьютера при записи соответствующих программ на изучаемом Я.П. Оценка эффективности, путем сравнения достигаемого уровня обученности при использовании языкового и алгоритмического подходов, показывает тенденцию к лучшему знанию тонкостей Я.П при работе АУК языкового подхода, но более твердые навыки конструирования алгоритмов — при использовании алгоритмического подхода. Свойство полноты методов обучения в МФ АУК представляет возможность комбинирования языкового и алгоритмического подходов на различных этапах обучения, обеспечивая тем самым высокие дидактические характеристики курса. Свойство расширяемости по языкам связано с наличием в структуре МФ АУК авторских средств, позволяющих вводить в систему курсы обучения новым языкам Я.П либо изменять обучающую программу для уже имеющихся языков. Свойство расширяемости по языкам обеспечивается открытостью структуры МФ АУК, которая позволяет оперативно и с незначительными трудозатратами производить настройку МФ АУК на новые Я.П.

МВ АУК организуется и эксплуатируется на базе пакета прикладных программ автоматизированной обучающей системы и представляется двумя самостоятельными частями — обучающими и обслуживающими режимами (рис.). В структуре курса главной управляющей программе (1) предписывается осуществление функций монитора, обеспечивающего вход обучаемого в систему и координацию взаимодействия режимов в процессе его работы. Режим изучения архитектуры и принципов функционирования компьютера



(2) отражает особенности той модели компьютера, на которой работает базовый пакет *автоматизированной обучающей системы* (АОС), в среде которого реализован МФ АУК. Режим обучения конструированию алгоритмов (3) реализует один из компонентов алгоритмического подхода к обучению и отражает особенности машинно-зависимого конструирования алгоритмов и конструирования алгоритмов с ориентацией на ЯП высокого уровня (см. *Уровень алгоритмического языка*). Для выбранной модели компьютера этот режим является частью МФ АУК, которая не расширяется при введении в систему новых ЯП. Обучающий режим *диалогового программирования* (4) представляет собой разновидность *тренажера по программированию*; реализуется алгоритмический подход к обучению и предназначен для решения *учебных задач* в условиях недостаточного знания ЯП. По заданной обучаемым блок-схеме алгоритма совместно с компьютером и под его управлением конструируется программа решения задачи. В процессе работы обучаемому предоставляется (если это необходимо) учебная информация, причем только в том объеме, который требуется для программирования его задачи. Расширение режима по ЯП производится с помощью авторских средств (8) и (9). Режим обучения (5) реализует языковой подход и предназначен для детального

изучения конструкций ЯП и формирования навыков в использовании эффективных приемов программирования. Режим обучения ЯП имеет также собственную управляющую программу, обеспечивающую выход на обучение требуемому языку. Аналогично производится расширение по языкам авторскими средствами. Режим выполнения и отладки программ (6) обеспечивает пользователю выполнение программ на любом ЯП операционной системы и предоставляет в его распоряжение набор отладочных средств (см. *Обучение отладке программ*). Обучающие функции режима обеспечиваются возможностью запроса помощи на этапе отладки программы и управление действиями пользователя со стороны компьютера. Режим выполнения и отладки программ не расширяется при введении в систему новых ЯП. Информационно-справочный режим (7) в составе МФ АУК позволяет обучаемому обратиться к библиотеке учебного обеспечения курса за получением различной информации (напр., по системным кодам завершения заданий), к сообщениям трансляторов, "листать" по собственному усмотрению учебные пособия, имеющиеся в библиотеке курса. Авторский режим генерации программ синтаксического анализа (9) строится на основе использования теории формальных грамматик и позволяет решить одну из важных проблем, определяющих дидактическую эффективность МФ АУК, — разработку средств анализа свободно конструируемых ответов при обучении программированию. Авторский режим позволяет достаточно просто получить набор подпрограмм-функций, осуществляющих синтаксический анализ ответов обучаемого, представляемых в виде свободно конструированных элементов изучаемого ЯП.

Помимо программного обеспечения, неотъемлемой частью МФ АУК является организационное, учебное и методическое обеспечение. Организационное обеспечение включает инструкции по генерации и сопровождению курса и предназначено для обеспечения эффективной эксплуатации и полного использования возможностей МФ АУК. Учебное обеспечение предназначено для непосредственных пользователей МФ АУК — обучаемых. Учебное обеспечение определяется содержанием обучения и включает, наряду с разделами учебного материала, рекомендации по организации взаимодействия с МФ АУК в процессе обучения. Методическое обеспечение — руководство для преподавателя, проводящего занятия с использованием МФ АУК, — содержит программу и технологическую карту курса, варианты заданий и типовые задачи, а также рекомендации по использованию в процессе работы определенных режимов МФ АУК в зависимости от контингента обучаемых.

Наиболее широкую известность и распространение в нашей стране получили МФ АУК для обучения программированию на *Фортране* — *АФРОДИТА* и МФ АУК для обучения программированию на *Ассемблере ЕС ЭВМ* — *АИДА*.

В. Д. Рынгац.

МНОЖЕСТВО ЭЛЕМЕНТОВ ОПОРНЫХ ПОНЯТИЙ — отобранный набор объектов предметной области (ПО), на основе которого реализуется целенаправленная начальная структура учебной деятельности обучаемого в компьютерной учебной среде. Позволяет обеспечить объективную ориентацию учебной деятельности. Композиционная структура учебной деятельности обучаемого строится на основе связей между объектами ПО. Описание М.э. о.п. реализуется в среде автоматизированного словарного комплекса учебного назначения.

А.Е.Стрижак.

МОБИЛЬНОСТИ ХАРАКТЕРИСТИКА — величина, оценивающая метод реализации мобильного программного обеспечения (см. *Мобильность пакета прикладных программ*) для сравнения различных методов и выбора наиболее подходящего. Одна из систем М.х. состоит из следующих показателей. Показатель начальной стоимости данного метода реализации — процентное отношение стоимости первой реализации к стоимости реализации программированием на ассемблере языке, Показатель стоимости переноса данного метода реализации — процентное отношение стоимости последующих реализаций к стоимости реализации программированием на языке ассемблера. Показатель ограничения переноса определяется как процент вычислительных установок на которых данный метод не применим. Показатель неэффективности — выраженная в процентах (усредненная) неэффективность реализации по скорости и объему по сравнению с реализацией на языке ассемблера.

Эти характеристики служат основой для сравнения и выбора метода мобильной реализации программного обеспечения методом экспертных оценок. Напр., если в качестве языка реализации выбрать язык ассемблера, то четверка характеристик имеет вид (Стоимость — 100%, Перенос — 100%, Неудачно — 0%, Неэффективность — 0%). Если переносимая программа имеет вычислительный характер, то оценки для языка Фортран могут иметь вид (30%, 5%, 0%, 10%). Это означает, что программу на Фортране можно написать в три раза быстрее, перенос ее не представляет трудностей, т.к. все объектные машины имеют компиляторы, и программа будет незначительно уступать программе на языке ассемблера по объему памяти ЭВМ и быстродействию.

В.А.Петрушин.

МОБИЛЬНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ — 1) Свойство программного обеспечения, состоящее в возможности его переноса с ЭВМ одного типа на ЭВМ другого с низкими трудозатратами; 2) раздел программирования, занимающийся теоре-

тическими и практическими вопросами переноса программ. М.п.о. является актуальной, т.к., во-первых, реализация большого программного обеспечения на новом классе ЭВМ весьма дорогостояща; во-вторых, аппаратные средства обновляются чаще, чем "интеллект" ЭВМ в виде программ; в-третьих, появление сетей ЭВМ привело к необходимости переноса программ и данных. М.п.о. обеспечивается следующими методами: написанием программы на широко распространенном стандартизованном языке программирования (напр., Фортране, Паскале, Си); написанием программы на инструментальном ЯП с дальнейшей его реализацией с помощью макропроцессора или компилятора компиляторов; методом "раскрутки" состоящим в построении серии совместимых компиляторов, причем каждый следующий написан на входном языке предыдущего, а первый компилятор создается мобильным одним из предыдущих методов или настолько прост, что его можно каждый раз написать заново. Для оценки метода реализации мобильного программного обеспечения существуют мобильности характеристики.

Свойство М.п.о. важно при создании программного обеспечения компьютерной технологии обучения. Различают мобильность инструментальных средств и прикладных программ учебного назначения. Создание мобильных инструментальных средств (напр., автоматизированных обучающих систем) обуславливает мобильность созданных с их помощью прикладных программ.

В.А.Петрушин.

МОБИЛЬНОСТЬ ПАКЕТА ПРИКЛАДНЫХ ПРОГРАММ — свойство пакета прикладных программ (ППП), определяющее возможность переноса его из одной операционной среды в другую. Характеризуется трудозатратами, необходимыми для осуществления переноса. Является важнейшим свойством ППП, включаемых в компьютерную технологию обучения (КТО), поскольку способствует упрощению их тиражирования, сопровождения, а также облегчает обучение работе с ними (не возникает необходимости переучивания при смене технической базы КТО).

Существуют два подхода к решению задачи разработки мобильных ППП. Суть первого состоит в том, что М.п.п. рассматривается относительно любого компьютера, т.е. предполагается, что перенос ППП возможен на любой компьютер. Альтернативный этому подход рассматривает перенос ППП на некотором классе компьютеров, относительно которого выдвигается требование, чтобы любой компьютер этого класса допускал принципиальную возможность реализации функций данного ППП с учетом требований конечного пользователя. К наиболее распространенным методам обеспечения М.п.п. относятся методы, в основу которых положено использование: универсальных языков программирования; универсальных

машинно-ориентированных языков; виртуальных машин (воображаемых машин, моделируемых на реальных с помощью программных средств); эмуляции (имитации) функционирования одной вычислительной системы средствами другой).

Осн. направлением развития исследований в области создания мобильных ППП является разработка специализированных технологий, обеспечивающих производство мобильных ППП в зависимости от области применения, в частности технологий разработки мобильного программного обеспечения (см. *Мобильность программного обеспечения*) учебного назначения.

Д.Н.Батанов, В.И.Отенко.

МОДЕЛЬ (франц. *modele*, от лат. *modulus* — мера, образец, норма) — система, которая благодаря структурному сходству с другой (моделируемой) системой может служить источником информации о ней либо ее заместителем. Всякая М. используется некоторой активной системой (отдельным человеком, организацией, научным сообществом, автоматом и т.п.). М. может быть вторична по отношению к моделируемой системе (для обозначения которой в этом случае используются также термины “прототип” и “оригинал”) или первична по отношению к ней. Так, чертеж можно считать М. изображенного на нем изделия и тогда, когда он выполнен по готовому изделию, и тогда, когда изделие изготавливается по чертежу.

Различают М.: материальные (напр., действующие М. механизмов или животные, используемые в медицинских экспериментах); идеальные (существующие в индивидуальном или общественном сознании); материализованные (изображения, *тексты* и их системы). Всякой идеальной М. соответствует несущая ее материальная или материализованная М. (напр., психической М. — некоторая система нервных процессов; понятийной М., существующей в науке, — некоторая система текстов). Материальные М., применяемые для передачи информации, описываются обычно под названием сигналов (они являются одновременно и вторичной, и первичной М., разумеется, по отношению к разным моделируемым системам). М., компонентами которой служат *знаки*, наз. знаковой. Такие М. чаще всего являются материализованными, но существуют также идеальные знаковые М. (напр., текст, заученный человеком *наизусть*).

Разнообразные М., несущие информацию об интересующих исследователей объектах, выступают как средства научного исследования. Вместе с тем в дисциплинах, изучающих функционирование активных систем (кибернетике, информатике, нейрофизиологии, психологии, лингвистике и др.), М. являются и предметом исследования. Специфические М., в которых достигнута высокая степень формализации и абстрагирования от несущественных особенностей

моделируемых систем (матем. М.), служат предметом исследования в математике. При целенаправленном создании М., в т.ч. математических, обычно стремятся к тому, чтобы М. была изоморфным (см. *Изоморфизм*) или хотя бы гомоморфным (см. *Гомоморфизм*) образом моделируемой системы.

Компьютерная технология обучения содержит ряд специфических, комфортных для *пользователей* компьютерных средств, позволяющих осуществлять моделирование с помощью компьютерной графики. Это облегчает формирование разнообразных материализованных М., их постепенное усложнение, способствует выполнению требований наглядности, активности *обучаемых* в построении таких М. и оперировании ими, реализации других *принципов обучения* в условиях КТО. Примером может служить М. *робототехнического участка*, сконструированного средствами *ЭЛКОНа*.

Г.А.Балл.

МОДЕЛЬ ДАННЫХ — фиксированная система *понятий* и правил для представления структуры *данных, состояния* и динамики *проблемной области* (ПО) в *базах данных*. Построение М.д. необходимо для обеспечения естественного отражения объектов реального мира в терминах структур, ограничений и *операций*. Задача сводится к тому, чтобы разработать М.д., которая наиболее естественным для конечного *пользователя* способом описывала бы ПО и имела возможность интерпретироваться компьютерными средствами. Такими М.д. являются инфологические *модели*, в основу которых положен аппарат формальной логики — исчисление предикатов. Помимо инфологической, существуют и другие, более общие модели, ориентированные на представление структур и данных реального мира.

Каждый компьютер обладает относительно простой, но достаточно определенной М.д. это допустимые в компьютере форматы *данных* и совокупность операций, выполняемых над ними. С помощью исходной М.д. можно построить более сложные. Для этого необходимо перейти к некоторой абстрактной машине, обладающей более удобной М.д. для представления исходных данных и решения прикладных задач. Примером абстрактных машин являются *трансляторы* или *интерпретаторы* с *алгоритмических языков*. Каждый язык *программирования* обладает своей М.д., независимой от машинной реализации и ориентированной на улучшение моделирования определенных видов реальных ситуаций.

М.д. *алгоритмического языка*, определяемая совокупностью его *операторов* декларативного и процедурного типа, обычно идентифицирует типы логических структур данных, которые можно представить соответствующей М.д., а также специфицирует признаки

(характеристики) структур данного типа, правила составления этих структур, правила обработки их процедурными операторами

Все вышесказанное можно отнести и к М.д. любой программной системы обработки данных и, в частности, к М.д., поддерживаемой конкретной *системой управления базами данных*.

Наиболее известны иерархическая, сетевая и реляционная М. д. Иерархическая модель — это набор элементов, каждый из которых представляет собой иерархическую структуру связи одного элемента (исходного) с другими (порожденными), т.е. отношение одного ко многим. Использование такой структуры обычно означает, что существует простой переход от порожденного элемента к исходному (т.е. для порожденного элемента имеется только один исходный), и что обратный переход (т.е. от исходного к порожденному) является сложным из-за присутствия связей “один ко многим”. Иногда существует простая связь в обоих направлениях (“один к одному”). Сетевая модель — развитие предыдущей модели, где объекты и их связи объединяются в виде сети, т.е. представляют собой отношение “многие ко многим”. В сетевой модели порожденный элемент может иметь более одного порождающего его элемента. Реляционная модель — совокупность отношений определенных порядков, изменяющихся во времени. Отношения в реляционной модели — это двумерные таблицы, состоящие из совокупности n -мерных кортежей (записей). Набор значений элементов одного типа, т.е. столбец отношения, наз. доменом (элемент данных). Осн. средствами манипулирования данными являются реляционная алгебра и реляционное исчисление, основанное на логике предикатов. В иерархическом представлении М.д. связи между элементами существуют явно, в сетевом — связи представляют с помощью указателей. Однако сам факт существования их означает, что связи рассматриваются раздельно. В реляционной модели связи представлены наглядно, причем большим преимуществом такой модели является возможность использования формальных методов манипулирования данными.

А.П. Ильяшенко, М.Е. Козлов.

МОДЕЛЬ ДИАЛОГА “ПРЕПОДАВАТЕЛЬ—ОБУЧАЕМЫЙ” — формализация обучающего диалога, служащая основой для создания подсистемы управления обучением в автоматизированных обучающих системах (АОС). Разработана в 80-х гг. в Чешском политехническом институте (г.Прага). Основана на представлении обучающего диалога в виде последовательности элементарных обучающих взаимодействий, каждое из которых состоит из трех фаз: стимул от преподавателя к обучаемому; реакция обучаемого на стимул (ответ обучаемого); анализ и отклик преподавателя на реакцию обучаемого. При планировании диалога в программах

учебного назначения (ПУН) каждому стимулу сопоставляют предполагаемые реакции обучаемого. В результате получается М.д. "п.-о." в виде ориентированного графа, узлы которого изображают обучающие взаимодействия, а дуги — переходы с одного стимула другому. Следовательно, при проектировании обучающего диалога выделяют множество стимулов, предполагаемых ответов и откликов, которые объединяют в триады элементарных обучающих взаимодействий предписыванием стимулу предполагаемых реакций и откликов на них, а затем предписывают каждому отклику др. триады.

М.д."п.-о." используются при создании ПУН в АОС (АОС-ВУЗ, Мобильных автоматизированных обучающих системах) и авторских системах (ВУЙСОР, РАКУРС).

Э.Мазик.

МОДЕЛЬ НАУЧЕНИЯ МАТЕМАТИЧЕСКАЯ — математическое описание процесса научения, т.е. усвоения *обучаемым* определенной системы знаний, умений и навыков, способствующих приобретению индивидуального опыта. Используется для матем. описания результатов психолого-дидактических экспериментов и для формального представления гипотетических *концептуальных моделей* процесса научения. В первом случае эмпирические данные, полученные при исследовании процессов научения, выражаются в виде матем. выражений в целях поиска закономерностей в совершенствовании поведения. Во втором случае матем. выражения используют для формально-теоретического представления концептуальных идей той или иной теории и технологии обучения.

В уравнениях М.н.м. чаще всего используются переменные, обозначающие силу навыка и ее предельно возможное значение, количество запомненного учебного материала, вероятности некоторых реакций, число повторений определенных учебных ситуаций. Разнообразие М.н.м. объясняется различием условий эксперимента (характер изучаемого материала, методики проведения экспериментов и др.).

М.н.м. широко применяются при исследованиях закономерностей процессов усвоения учебного материала, обосновании методик обучения и поиска способов оптимизации *технологии обучения*.

Ю.И.Лобанов, Т.Б.Протасова.

МОДЕЛЬ ОБУЧАЕМОГО — компонент программы учебного назначения (ПУН), содержащий *информацию об обучаемом*. Различают, как правило, следующие составляющие М.о.: обобщенные характеристики обучаемого — психофизические (темперамент, экстравертированность, пол, возраст и т.п.), социальные (уровень полученного образования, профессия и т.п.) и др. величины, влияющие на процесс обучения и характеризующие обучаемого безотносительно к

изучаемой предметной области (ПО); информация об отношении обучаемого к изучаемой ПО; история обучения, т.е. осн. характеристики процесса взаимодействия обучаемого с ПУН. Иногда М.о. наз. только информацию об отношении обучаемого к изучаемой ПО (М.о. в узком смысле). Различают *накрывающие* и *процедурные* М.о. в узком смысле.

Накрывающая (оверлейная) М.о. состоит в самом простом случае из набора, а в наиболее общем случае — из сети характеристик по одной на каждое изучаемое понятие, отношение или задачу. Каждая характеристика (которая может быть составной) ставит в соответствие ПО показатель типа “знает/не знает” (“имеет/не имеет”) или более сложную оценку степени владения обучаемым данным материалом. Если под моделью эксперта понимать модель предметной области с максимально возможными оценками качества владения ее элементами, то накрывающая М.о. является подмножеством модели эксперта. *Процедурная М.о.* представляет собой модель деятельности обучаемого и представляется алгоритмом, набором правил или в самом общем случае процедурной сетью, вершины которой соответствуют возможным состояниям, а дуги — операторам, преобразующим одно состояние в другое. *Процедурная М.о.* отражает его текущее состояние и содержит как правильные, так и ошибочные схемы деятельности, поэтому М.о. этого типа наз. моделирующей ошибки. *Накрывающая М.о.* более проста в реализации и поэтому широко используется в ПУН (*FLEX, GUIDON*), тогда как *процедурная М.о.* применяется в интеллектуальных тренирующих программах (*BUGGY*).

В.А.Петрушин.

МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ — совокупность структурированных знаний об объектах и закономерностях предметной области (ПО), о способах решения и средствах решения задач в моделируемой среде. В автоматизированных обучающих системах (АОС) М.п.о. является предмет обучения. М.п.о. служит основой для работы различных блоков АОС: генерации учебных задач, анализа ответов обучаемого, формирования диагностических сообщений, выбора пути продолжения обучения для конкретного обучаемого (управление процессом адаптивного обучения). Включает информацию: об анализируемых объектах; отношениях, отражающих иерархическую структуру объектов; анализируемых свойствах для каждого объекта и возможных значениях для каждого свойства; о правилах, определяющих последовательность изменения свойств; функциях и/или процедурах, позволяющих вычислить значения требуемых свойств объектов (сюда включаются и операции, переводящие объект из состояния в состояние, т.е. функции, изменяющие значения свойства, отражающего состояние объекта); предикатах

(отношениях), соответствующих предусловиям применения некоторых функций, напр., условиям применимости операций, переводящих анализируемый объект из состояния в состояние; правилах применения операций, соответствующих упомянутым выше функциям и/или процедурам; об отношениях, отражающих иерархическую структуру функций и/или процедур; свойствах, значения которых задаются априори. Эта информация определяет общую структуру М.п.о.

В программных системах для описания М.п.о. чаще всего используются логические, фреймовые, сетевые *методы представления знаний* и различные их комбинации. Особое значение при выборе представления имеет специфика данной ПО. Напр., если ПО содержит описательную информацию, то, как правило, используется информационно-сетевая модель. Если изучаемый предмет имеет сильно выраженную иерархию, то в качестве М.п.о. используется дерево в той или иной его модификации.

М.п.о. для изучения любого языка программирования должна содержать знания о его синтаксисе, семантике, а также обладать информацией по диагностике ошибок обучаемого и может быть представлена в виде сети. Каждому узлу сети можно поставить в соответствие определенный класс объектов языка. Дуги сети отражают информационные связи между объектами языка. Множество узлов сети, соответствующих объектам языка, можно задать на ряд уровней в зависимости от сложности и специфики различных подмножеств объектов. Напр., для языка *Пролог* этими уровнями будут: примитивы — некоторые первичные элементы языка, известные любому обучаемому и не требующие дополнительного объяснения (напр., буквы, цифры); элементарные конструкции языка (идентификаторы, переменная, число, строка); составные объекты (операции, выражения, функции, суперпозиции операций и функций); атомарные формулы (предикаты, метапредикаты, встроенные процедуры); предложения языка (аксиомы, теоремы); программы. Знания о предмете в каждом узле сети, а также об их использовании при организации процесса обучения (прежде всего в задачах генерации заданий обучаемому, конструировании ответов) можно представить набором *фреймов*, связанных между собой в информационную сеть указанного вида. Для каждого фрейма должны быть выделены априорные понятия. Вызов фрейма возможен лишь тогда, когда априорные понятия помечены как известные для данного обучаемого. Напр., для реализации процесса генерации задания обучаемому либо формирования *ответа* на его *запрос* (просьба привести пример) иницируемый фрейм должен содержать правила конструирования объекта языка и включать также процедуру генерации этого объекта. Следовательно, фрейм, соответствующий узлу сети, представляющей модель изучаемой ПО, имеет следующую структуру: объект языка для данной цели обучения, т.е. предмет

изучения; множество информационных связей с др. объектами языка (входящие и выходящие дуги для рассматриваемого узла информационной сети); процедура генерации объектов данного класса; процедура диалогового конструирования объектов данного класса (приобретает особое значение для изучения объектов более высоких уровней); синтаксические правила построения объекта языка (в языке исчисления предикатов первого порядка); семантика объектов — текстовая информация или формализованное представление семантики объектов языка данного класса. Слотами, т. е. заполняемыми единицами сконструированного фрейма, будут: условия, которые должны удовлетворяться при вызове фрейма (напр., множество априорных понятий для этого фрейма); множество сгенерированных объектов языка данного класса; количественные и качественные характеристики усвоения данного материала.

Л.Н.Гецко, Р.Ковалюнас, В.В.Колос.

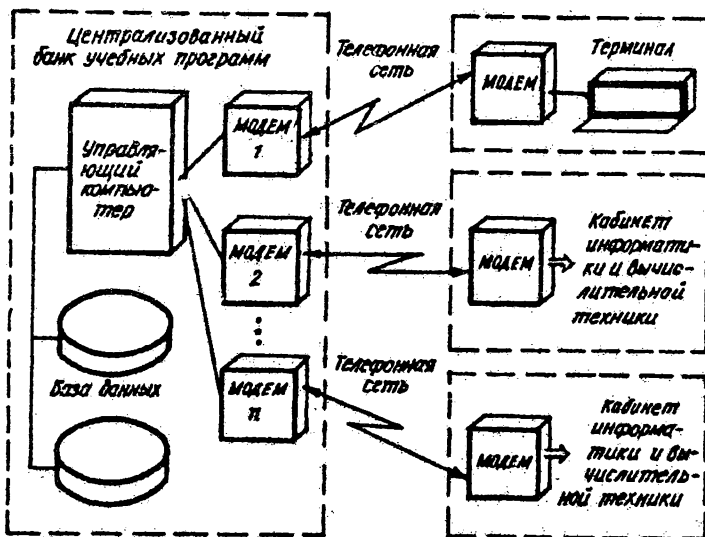
МОДЕЛЬ ЯЗЫКА — *текст* или какое-либо обобщение его в некотором алфавите. Согласно классификации И.И.Ревзина М.я. подразделяют на два класса: синтезирующие, в которых текст является выходным объектом, т.е. результатом действия модели, и аналитические М.я., в которых текст является входным объектом, т.е. исходным материалом для модели. Синтезирующие М.я. делятся на собственно синтезирующие, или модели синтеза (по заданному смыслу производят конечное множество соответствующих ему фраз) и порождающие, которые перечисляют все бесконечное множество правильных последовательностей элементов или к.ассов элементов языка, приписывая каждой ее синтаксическую (и по возможности семантическую) структуру (модель Хомского). Аналитические М.я. отличаются от синтезирующих тем, что их конечным результатом являются не конкретные фразы, а описания лингвистических фактов. На входе имеется некоторая совокупность фактов, характеризующих одновременно вход и выход собственно синтезирующей модели, заданы слова и их значения, фразы и их смысловые связи. Выходом такой модели должны быть определенные категории морфологии и синтаксиса. В машинном моделировании языка, наряду с указанными типами моделей, широко используются также М.я. статистические, теоретико-множественные, основанные на методах матем. логики и пр. Работы в области построения алгоритмов синтаксического анализа показали, что сложности синтаксических структур намного выше, чем это предполагалось ранее. Выяснилось, что без использования дополнительной семантической информации осуществление полного синтаксического анализа невозможно. Одновременно практика информационных работ показала, что имеющиеся алгоритмы неполного, частичного синтаксического анализа могут успешно применяться для многих практических задач обработки

сообщений, содержащих фрагменты и тексты (информационный поиск, *машинный перевод*, стандартные типы человеко-машинного диалога).

Операции над смыслами принципиально отличаются от операций над *символами*, условными *знаками*, что не учитывается разработчиками матем. М.я. При моделировании коммуникативного а на компьютере построение М.я. и предварительная интерпретация смыслов в терминах модели выполняется человеком. Это творческий процесс, т.к. сам процесс формализации не формализуем. Следовательно, нельзя дать и формальное описание процесса понимания сообщения; можно построить лишь грубый аналог для предварительной составленной *базы данных* о некоторых четких ситуациях. Исследования в области построения *программ*, понимающих естественный язык, выявили трудности и всю сложность решения как проблемы содержательной обработки сообщений, так и проблемы отбора и представления *знаний*, необходимых для *понимания* текста. Представление экстралингвистической информации превратилось в самостоятельную задачу, для решения которой используются различные подходы, основанные на применении сетевых структур, *фреймов*, планов, сценариев и пр. Ныне эти исследования характеризуются все более широким использованием идей функциональной *грамматики*, концептуального анализа, привлечением обширной экстралингвистической информации о единицах языка и *предметной области*. Поиски новых *стратегий* анализа языка плодотворны в области функциональных грамматик, в которых анализ осуществляется без разделения на синтаксический и семантический этапы. Для становления новых теорий языка и речевой деятельности, ориентированных на создание средств обработки сообщений на естественном языке, существенное значение имеет также использование результатов психологии, психолингвистики, когнитологии.

Т.Б. Андрусенко.

МОДЕМ, модулятор-демодулятор — устройство преобразования *информации*, представленной в цифровом виде, в специальные (электрические, оптические и др.) сигналы при передаче (модуляции) и обратного преобразования при приеме (демодуляции). Применяется при автоматической связи через последовательный канал, и форма представления сигнала определяется требованиями конкретного канала связи. Позволяет передавать данные (учебные программы) на большие расстояния по телефонной сети. М. имеет два *интерфейса* — акустический или электрический с аналоговой цепью (телефонной линией) и многопроводный цифровой с компьютером (рис.). При скоростях передачи до 20 Кбит/с для подключения М. с компьютером используется несимметричный интерфейс RS 232C (стандарт EIA, США), применяемый в Европе как интерфейс



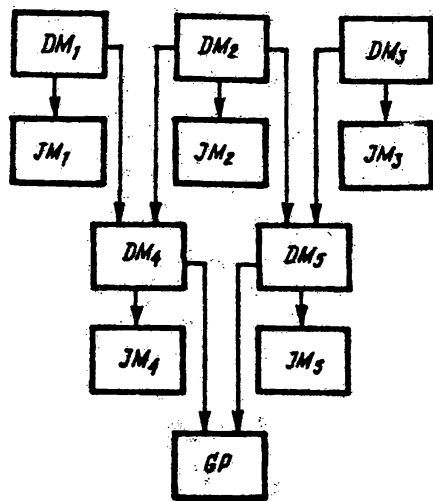
V.24/V.28 МККТТ (Международный консультативный комитет по телеграфии и телефонии), В качестве каналов связи используются телефонные линии. При скоростях передачи от 48 до 168 Кбит/с используются широкополосные М., которые связываются с компьютером по интерфейсу V.35 МККТТ. В этом случае применяют широкополосные линии связи.

В зависимости от режима эксплуатации различают М. для асинхронной стартстопной передачи (скорости передачи могут изменяться *пользователем*), для асинхронной и синхронной передачи — синхронные М., которые при использовании специальных форматов знаков могут работать в режиме стартстопной передачи данных (они генерируют сигналы синхронизации и работают с фиксированными скоростями передачи), для синхронной передачи (такие М. работают в режиме синхронной блоковой передачи с фиксированными скоростями передачи и генерируют сигналы синхронизации). По типу связи М. могут работать в дуплексном и полудуплексном режимах. Изготавливают М. в виде автономного конструктивного блока, используя законы частотной модуляции, когда код знака определяется частотой передаваемого сигнала или законов амплитудной модуляции — код знака в информации определяется амплитудой передаваемого сигнала.

М. применяют в вычислительных системах, в комплексах и сетях для оборудования кабинетов информатики и вычислительной техники.

И.А.Емченко, В.Л.Леонтьев, И.Стрембицки.

МОДУЛЬ-2 — язык программирования (ЯП) для малых и персональных электронных вычислительных машин. Относится к классу императивных (процедурных) языков высокого уровня (см. *Уровень алгоритмического языка*). Разработан швейцарским ученым Н.Виртом в 1980 на базе ЯП Паскаль и Модуля. От первого из них М.-2 унаследовал типы данных, операторы и процедуры, от второго — концепцию модуля. Модуль в языке М.-2 — это именованный набор описаний вместе с оператором (телом модуля), заключенным между ключевыми словами *begin* и *end*. Набор описаний может включать компоненты: список импорта, список экспорта, определение констант и типов, описание (внутренних) переменных, описание процедур. В списке импорта указываются идентификаторы объектов (констант, переменных, типов, процедур), описанных вне модуля, но использующихся внутри него. В списке экспорта перечисляются идентификаторы тех объектов, которые определены в данном модуле и используются другими модулями. Определения констант, типов, описания переменных и процедур в М.-2 имеют ту же форму и смысл, что и в языке Паскаль. Тело модуля служит для инициализации внутренних переменных, при отсутствии их телом модуля будет пустой оператор. Следовательно, программист может управлять видимостью (т.е. доступностью) всех определяемых объектов. Это позволяет с помощью модуля реализовать абстрактный тип данных или абстрактный объект. В языке различают три типа модулей — главные, определяющие и реализующие. Главный модуль не содержит списка экспорта; он образует



гл. программу. Определяющий модуль содержит только описания: список экспорта, определения констант, типов, описание переменных и заголовки процедур (но не их тела). Каждому определяющему модулю соответствует реализующий модуль, содержащий полное описание процедур и, возможно, описания др. неэкспортируемых объектов. Все объекты, описанные в определяющем модуле, доступны в соответствующем реализующем модуле (без явного указания в списке импорта). Следовательно, программу на М.-2 можно представить как иерархический набор модулей (рис.). В изобра-

своном на рисунке примере главная программа (модуль GP) использует объекты, определенные в модулях DM₄, DM₅, (и реализованные в модулях IM₄, IM₅), а модуль DM₄ (а значит, и IM₄) импортирует объекты из DM₁ и DM₂ и т.д. Все эти модули компилируются раздельно. Поэтому часть из них можно разработать заранее, что дает возможность расширять реализацию языка (предопределенные модули). Именно таким способом в М.-2 реализованы средства ввода-вывода, средства работы с памятью ЭВМ и др. машинно-зависимые средства. Язык М.-2 позволяет разрабатывать большие программные системы с хорошей модульной структурой и высокой степенью мобильности пакета прикладных программ (за счет локализации использования машинно-зависимых средств).

В.М.Антимиров.

МОНОТОНИЯ (от греч. *μόνος* — один, единый и *τόνος* — напряжение) — комплекс физиологических и психологических изменений в организме человека, возникающих в специфических условиях деятельности и внешней среды (однообразная рутинная работа, напр., на конвейере, шум и пр.), вызывающих однообразную реакцию. Различают “монотонность”, характеризующую внешние объективные факторы трудовой деятельности и “состояние монотонии”, которое в большей степени относится к реакциям человека на монотонный труд. Субъективные проявления М.: скука, апатия, сонливость, усталость, раздражительность, снижение внимания и др.; они выступают и как ее внешние признаки, и как защитные реакции человеческого организма. М. обычно характеризуется снижением положительной мотивации к деятельности. Меры профилактики возникновения и развития состояний М. при работе с компьютером: рациональная организация трудовой деятельности, внедрение психологически обоснованных режимов труда и отдыха, использование различных активизирующих психологических средств.

Н.И.Повякель.

МОРФОЛОГИЧЕСКИЙ АНАЛИЗ (МА) — обработка словоформ вне связи с контекстом. Функцией МА является идентификация словоформы и приписывание ей комплекса морфологической информации (КМИ), необходимой для последующего синтаксического анализа и семантического анализа. В общем случае КМИ состоит из совокупности морфологических информаций (МИ) — строк. К МИ относят.: одушевленность, род, число, падеж, вид, лицо, залог и т.п. Алгоритм МА зависит от принятого в системе способа хранения информации. Существуют два осн. метода реализации МА: декларативный и процедурный. В декларативном методе реализации МА (декларативный МА) в словаре хранятся все возможные

словоформы каждого слова с приписанной им МИ. По сути дела, в декларативном МА нет собственно МА, а хранится его результат. Задача декларативного МА состоит в поиске словоформы в словаре и переписывании из словаря КМИ, соответствующего данной словоформе. Это приводит к тому, что декларативный МА работает быстрее, чем процедурный МА. К недостаткам декларативного МА относится необходимость хранения всех словоформ каждой основы, что увеличивает трудоемкость подготовки словарной информации по сравнению с процедурным МА. При процедурном МА в словаре системы хранятся основы слов. Функции процедурного МА: выделение в словоформе основы, идентификация ее и приписывание данной словоформе соответствующий ей КМИ.

Н.А.Власенко.

МОТИВАЦИЯ (англ. *motivation*, от лат. *motivus* — побуждение) — совокупность причин, побуждающих человека к активной деятельности и придающих ей осмысленность. Представляет собой механизм развития, конкретизации и опредмечивания потребностей, которые не только определяют собой мотивы, но изменяются и обогащаются в деятельности человека. Деятельность побуждается одновременно несколькими мотивами, в комплексе обуславливающими положительную или отрицательную М. Обычно один из мотивов является ведущим, а другие — подчиненными, иногда выполняющими лишь функцию дополнительной стимуляции. Особенность ведущих мотивов состоит в том, что наряду с функцией побуждения и направления деятельности они придают деятельности, ее объектам и условиям тот или иной субъективный, личностный смысл. М. играет важную роль в любых видах деятельности. Формирование положительной М. — обязательный этап организации успешной целенаправленной деятельности человека. Важное место занимает М. и в компьютеризованной деятельности человека. Наличие резко отрицательной М. к работе с компьютером понижает продуктивность труда, повышает вероятность возникновения ошибок. Формирование положительной М. к работе с компьютером является одной из ведущих предпосылок *психологической готовности* к деятельности, опосредствованной компьютерами. При организации *компьютеризованного обучения* важнейшую роль играет формирование положительной М. *обучаемых*.

Н.И.Повякель.

МПРОЛОГ — язык, позволяющий создавать программы, состоящие из нескольких модулей, на языке Пролог (модульный Пролог). Является попыткой создать реализацию Пролога для практических применений в больших масштабах. МП. — наследник первой венгерской реализации Пролога, созданной в 1975 “Старый”

Пролог опирался в основном на марсельскую модель и был использован в многочисленных экспериментальных, а также нескольких промышленных приложениях. Работа по проекту МП. началась в 1978 с целью создать совершенно новую реализацию, соединяющую опыт эксплуатации с усовершенствованными методами реализации.

Система МП. предназначена для того, чтобы удовлетворять двум противоположным стремлениям: поддерживать процесс разработки программы и помогать в создании *Пролог-программ*, пригодных для промышленной эксплуатации. Чтобы обеспечить дружескую для пользователя среду при создании программ, была построена специальная подсистема разработки программ (написана на МПро-логе).

Б. Домёлки.

MPS (лит. Mokyto Programavimo Sistema — ...) — экспериментальная система обучения решению задач программирования на языке ПЛ/1, функционирующая в составе типовой обучающей системы АОС-ВУЗ. Знания об учебной среде программирования представлены в MPS моделью вида *семантических сетей*. В учебной среде программирования выделено 9 обобщенных подзадач, имеющих характерные планы решения. *Учебные задачи* состояются из различных комбинаций подзадач с полностью или частично конкретизированными значениями. Анализ ответа (программы *обучаемого*) основывается на знании структуры решения задачи и проходит в 3 этапа: 1) построчный синтаксический контроль и формирование семантического представления программы; 2) настройка описания задачи на конкретное решение; выяснение *способа решения* задачи; выяснение соответствий между переменными в программе и объектами в описании задачи; 3) анализ подзадач согласно сценарию анализа обобщенных подзадач, представленных в модели предметной области. На последних этапах анализа неправильных решений (для выбора наиболее прагматически правильной интерпретации ошибок) используется информация, получаемая от обучаемого в ходе диалога о решаемой задаче. Для выяснения способа решения система выдает обучаемому *вопросы* по описанию задачи и по *ответам* и определяет подзадачи, имеющиеся в программе обучаемого. Затем путем анализа программы выясняется назначение каждой переменной с дальнейшим сопоставлением их объектам описания задачи. При неполном или неоднозначном соответствии выдаются дополнительные вопросы обучаемому. После такой настройки обобщенного описания задачи на конкретное решение выполняется анализ правильности реализации каждой подзадачи.

Пусть, напр., необходимо решить следующую задачу: "Имеется 20 различных товаров, характеризующихся ценой и сортом (1,2,3).

Уменьшить цену товаров 3-го сорта на 10%." В процессе анализа программы возможен такой диалог:

MPS: ДАННЫЕ О ТОВАРАХ в вашей программе представлены в отдельных массивах или матрицей?

Об.: В отдельных.

MPS: Я полагаю, что переменная I в вашей программе используется ДЛЯ ИНДЕКСАЦИИ.

Об.: Да.

MPS: Каким именем Вы обозначили массив ЦЕНЫ ТОВАРОВ?

Об.: C.

MPS: Ввод ЦЕНЫ ТОВАРОВ: правильно,
Ввод СОРТЫ ТОВАРОВ: правильно,
Цикл ДЛЯ ИНДЕКСАЦИИ: правильно,
Вычисление ЦЕНЫ ТОВАРОВ:

Должно вычисляться по формуле: $C(I)=C(I)-0.1*C(I)$,
а у Вас: $C(I)=10*C(I)\backslash 100$.

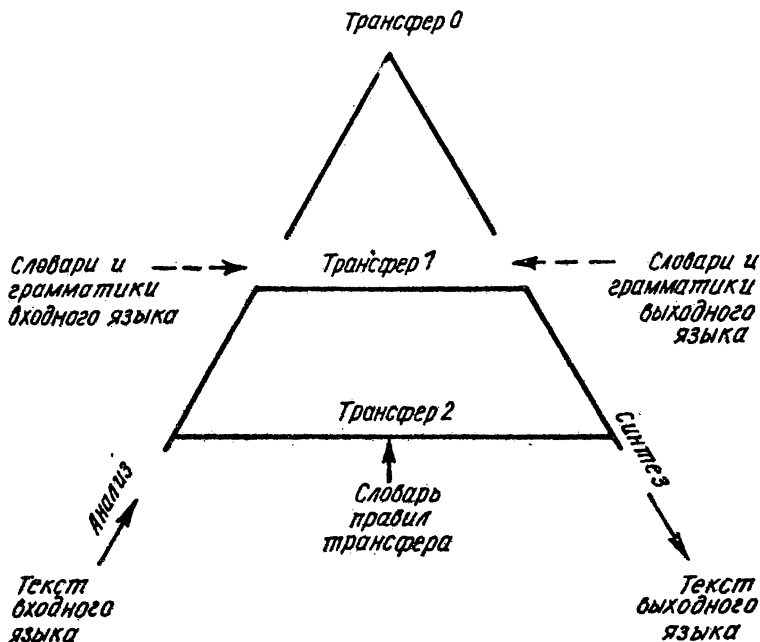
Заглавными буквами в сообщениях MPS выделены названия объектов из формального описания задачи.

MPS способна анализировать учебные программы длиной до 40 строк. В программах разрешается использовать простые переменные, одно- и двумерные массивы, содержащие числовые данные. Каждая решаемая задача должна быть представлена в системе формальным описанием. Язык описания задач позволяет представлять задачи содержательного характера. MPS обеспечена простым редактором текста программы и имеет собственный диалоговый отладчик программы. Имеются автоматизированные учебные курсы, рассчитанные на проведение десяти одночасовых лабораторных работ в дисплейном классе. Задачник содержит 80 описаний задач. Точность диагностики ошибок достигает 90%.

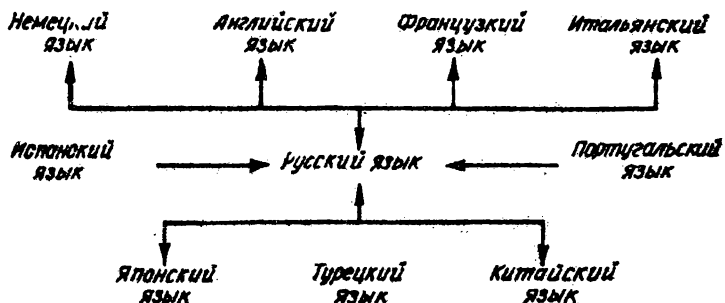
Р.Кавалюнас, В.Реклайтис.

MULTIS (от MULTIlanguage machine translation System — многоязыковая машинная трансляционная система), СИЛОД-ПК — многоязыковая система автоматизированного машинного перевода, использующая в качестве лингвистической стратегии "межъязыковые операции" — трансфер (рис.4). Была создана в 1980 в Ленинградском педагогическом институте им.А.И.Герцена в результате адаптации системы машинного перевода СИЛОД (Система Информационно Логической Обработки Документов) для персонального компьютера (отсюда второе название).

M. функционирует на 16-разрядных персональных электронных вычислительных машинах с жестким диском. Система заметно убыстряет перевод научно-технических и публицистических текстов и может использоваться в качестве вспомогательного средства для получения экспресс-информации специалистами, не владеющими



иностранном языке. В качестве трансфера выступают машинные грамматики, что позволяет не только быстро получать перевод, но и адекватно передавать смысл исходного текста. Гибкая структура программного обеспечения и лингвистического обеспечения дает возможность включать новые входные языки и формировать лингвистическую базу для новых тематических областей (рис.).



К.Р.Пиотровский

МЫШЬ — см. Манипулятор ручной.

НАКОПИТЕЛЬ — внешнее запоминающее устройство для ввода — вывода и хранения информации. Содержит носитель информации и блок записи / чтения. Различают Н. электронные и электромеханические. В электронных Н. носителями информации служат полупроводниковые и на магнитных доменах микросхемы, магнитные пленки и кольца (последние применяются все реже).

Электромеханические Н. бывают с носителями на магнитной ленте (НМЛ), на гибких и жестких (*винчестерский накопитель*) магнитных дисках (НМД) и на оптических дисках (НОД). Пробразом НМЛ послужил магнитофон. Механическая часть блока записи / чтения НМЛ содержит подающий и приемный узлы, между ними находится неподвижная магнитная головка, к которой прижимается лента в рабочем режиме (считывания или записи). В этом режиме лента движется с постоянной скоростью, сматываясь с подающего узла и наматываясь на приемный. Электронная часть НМЛ связывает магнитную головку с ЭВМ, формируя сигналы с необходимыми амплитудно-временными характеристиками, а также управляет движением ленты в режиме поиска информации. В режиме поиска лента движется значительно быстрее, чем в рабочем режиме.

Пробразом НМД послужил проигрыватель грампластинок. Механическая часть НМД содержит узел вращения диска и узел перемещения магнитной головки. Диск вращается всегда с постоянной скоростью. Узел перемещения в режиме поиска с помощью электронной части устанавливает магнитную головку на выбранную дорожку диска. В рабочем режиме головка неподвижна, и электронная часть НМД работает так же, как и в НМЛ.

НОД работает аналогично НМД, но в нем вместо магнитной головки используется лазер.

В.Л.Леонтьев.

НАСТОЛЬНАЯ ИЗДАТЕЛЬСКАЯ СИСТЕМА (НИС) — *автоматизированное рабочее место*, обеспечивающее создание текстовых и графических документов различного назначения в пригодной для типографской обработки форме. Применяется в управленческих, издательских, торговых системах, для рекламы, проектирования и производства продуктов (в системах САПР и АСУП), в обучении, здравоохранении, спорте и др. видах деятельности. НИС обеспечивает ввод, редактирование и вывод текстовой и графической информации. Ввод текстовой информации осуществляется с клавиатуры терминала с использованием различных шрифтов (латинского, кириллицы, греческого), а также посредством оптического считывателя для дальнейшей обработки в виде текстового файла; графическая информация вводится с помощью аналоговых устройств ввода (мышь (см. *Манипулятор ручной*), световое перо, аналоговое табло и др.)

и оптического считывателя; текстовая и графическая информация — с различных носителей информации и из сети переноса данных. Редактирование осуществляется: стандартными функциями текстовых редакторов; специальными функциями редактирования (полуавтоматический перенос, просмотр орфографии и терминологии, нумерация примечаний и сносок, составление именных и предметных указателей, составление оглавления, административно-учетная обработка текста, поиск ключевых слов и контекстов и пр.); типографскими функциями, позволяющими применять: различные формы и типы шрифтов с изменением размеров в одном и том же тексте; линейную, пятнистую и растровую графику внутри текста; верстку, верстку по колонкам, изменение верстки внутри страниц или фрагмента текста; различные виды выделения, цвет. НИС осуществляет вывод текста, линейных, пятнистых и растровых график на бумагу, экран и световое табло; обеспечивает многоступенчатую плотность, цветной вывод, изменение размеров текста и графики, одно- и двухстраничную печать, печатание нескольких экземпляров, вывод на специальные носители (слайды шрейбпроектора, типографские пленки, печатные плиты).

В состав НИС входят: персональные электронные вычислительные машины или мини-ЭВМ; базовое и инструментальное программное обеспечение; печатающее устройство (лазерное и/или мозаичное); оптическое считывающее устройство (сканер); прочие устройства (мышь, копировальное и др.). Наибольшее распространение получили НИС, использующие ПЭВМ. Критерии при выборе компьютера для НИС: программная база должна обеспечивать необходимые функциональные требования к НИС; центральный процессор должен обеспечить работу необходимых программных средств; емкость оперативной памяти ЭВМ, определяемая базовым программным обеспечением, должна быть достаточно велика, чтобы увеличить скорость обработки информации большого размера. В состав внешней памяти входят: стационарный диск (служит средством быстрой обработки информации); винчестерский накопитель; накопитель на гибких магнитных дисках (для выполнения редакторских работ не удобен; используется для хранения архива и обеспечения ввода—вывода в открытых системах). Устройства ввода—вывода: клавиатура (для обеспечения ввода национальных алфавитов и управляющих функций); оптический считыватель для ввода целых страниц; печатающие устройства с высоким качеством печати (лазерные или АЦПУ с чернильной струей); мозаичная печать для промежуточных работ; цветной графический дисплей (полностраничный экран формата ДИН А4); средства связи (в системах местной и дистанционной сети). Большинство доступных в настоящее время НИС реализовано на компьютерах семейства IBM PC.

А.Киш.

НЕПРЕРЫВНАЯ ПОДГОТОВКА СПЕЦИАЛИСТОВ. При Н.п.с. используют комплекс учебно-методических документов и рекомендаций, в которых наряду с требованиями к специальной подготовке установлены объем и содержание подготовки на всех этапах обучения, даны рекомендации по оборудованию специализированных учебных лабораторий, по обеспечению направлений подготовки, включая автоматизацию управления учебной деятельностью. Осн. принципами Н.п.с. являются согласование содержания подготовки и обеспечение его преемственности на всех уровнях образования. Для определения степени освоения *компьютерной грамотности* введены разные уровни базовой и специальной подготовки. На разных уровнях образования требования к базовой подготовке различны. Базовый курс информатики в школе включает овладение осн. понятиями *информатики*, понимание принципов функционирования, управления и общения с компьютером, овладение простейшими навыками *пользователя*. Базовая подготовка в системе профессионально-технического образования развивается в направлении более глубокого и детального освоения тех разделов информатики, которые связаны с формированием профессиональных умений. В системе высшего и среднего специального образования предусмотрены три уровня базовой подготовки: первый уровень предназначен для специальностей, использующих компьютер как инструмент при выполнении расчетов и обработке текстовой *информации* (в основном гуманитарные специальности — квалификация “пользователь”); второй уровень — для специальностей, в которых компьютер используется как инструмент профессиональной деятельности при автоматизированном *проектировании*, автоматизации научных исследований, технологических процессов и т.д. (инженерные и естественные специальности — квалификация “прикладной программист”); третий уровень — для специальностей, связанных с разработкой технических и программных средств информатики (только для вузов — квалификация “системный программист”). Цель базовой подготовки, получаемой учащейся молодежью в общеобразовательной и профессиональной школе, — дать *обучаемым* сведения об информатике и вычислительной технике, заложить основы дальнейшего совершенствования в специальных областях применения вычислительной техники (таких, как проектирование микропроцессорных систем, автоматизация проектирования и управления и т.д.). Следующая ступень является начальным этапом специальной подготовки молодежи в области применения вычислительной техники и требует знания возможностей современной элементной базы средств вычислительной и микропроцессорной техники, умения использовать ее при создании различных автоматических приборов, устройств, электронных *игр и тренажеров*, программно-управляемых *моделей* и т.п. Освоение этой ступени осуществляется в кружках технического творчества молодежи, школах юных программистов и т.п. под руководством опытных

специалистов. Дальнейшее повышение уровня квалификации в области информатики и вычислительной техники осуществляется в системе высшего и среднего специального образования, на факультетах повышения квалификации специалистов и предусматривает последовательное освоение уровней базовой и специальной подготовки, предусмотренных требованиями к базовой и специальной подготовке специалистов различных специальностей. Эти требования устанавливают три уровня базовой и по два уровня специальной подготовки в следующих направлениях: микропроцессорная техника (МПТ); системы автоматизированного проектирования (САПР); автоматизированные системы управления технологическими процессами (АСУТП); автоматизированные системы научных исследований (АСНИ). Первый уровень базовой подготовки назначается студентам, в будущей профессиональной деятельности которых вычислительная техника служит вспомогательным средством. Специалисты этой категории по мере накопления прикладного программного обеспечения ЭВМ, относящихся к их предметной области, должны стать программируемыми пользователями, умеющими эффективно применять ЭВМ при выполнении своих служебных обязанностей. Второй уровень базовой подготовки назначается студентам, готовящимся к широкому использованию вычислительной техники и различных автоматизированных систем на ее базе. К этой категории относятся студенты, будущая профессиональная деятельность которых связана с различного рода проектированием, выполнением большого количества расчетов и т.п. Третий уровень подготовки назначается студентам, будущая профессиональная деятельность которых связана с проектированием вычислительной техники и программного обеспечения ЭВМ, а также различных автоматизированных систем, базирующихся на широком использовании средств вычислительной техники. Первый уровень специальной подготовки — для специалистов, являющихся пользователями соответствующих систем (оборудования со встроенными микропроцессорными системами, систем автоматизированного проектирования и т.д.). Второй уровень специальной подготовки предназначен тем, кто не только использует указанные системы, но и участвует в их разработке (проектирует оборудование со встроенными микропроцессорными системами, разрабатывает автоматизированные технологические процессы, участвует в создании пакетов прикладных программ САПР по своей специальности и т.д.).

Система непрерывной подготовки (СНП) включает рекомендации относительно объема и содержания подготовки студентов в зависимости от предъявляемых требований к их знаниям, навыкам и умениям, а также типовые учебные программы. Рекомендации устанавливают состав изучаемых дисциплин и примерное количество часов учебного времени, отводимого на их изучение. Типовые программы рекомендуемых для изучения учебных дисциплин состав-

лены ведущими специалистами вузов страны, накопившими значительный опыт подготовки специалистов соответствующего профиля. В основном теми же специалистами подготовлены к изданию комплексные учебные пособия, обеспечивающие базовую и специальную подготовку студентов вузов. При использовании учебно-методических документов и рекомендаций СНП для составления новых учебных планов подготовки специалистов необходимо: 1) пользуясь рекомендациями СНП, установить, какими уровнями базовой и специальной подготовки должны овладеть в процессе обучения в вузе студенты соответствующей специальности; 2) используя "Требования к подготовке специалистов в области применения вычислительной техники", составить текст части квалификационной характеристики специалиста, отражающей требования к знаниям, навыкам и умениям специалиста в области применения вычислительной техники; 3) учитывая рекомендации по определению объема и содержания подготовки специалистов в области применения вычислительной техники, определить состав учебных дисциплин, обеспечивающих базовую и специальную подготовку студентов, и примерное количество часов, отводимое на их изучение; 4) проанализировать содержание учебных дисциплин по типовым программам для устранения возможного дублирования материала и исключения из содержания подготовки разделов, изучение которых студентами данной специальности нецелесообразно; 5) принять решение о том, какие из рекомендуемых СНП дисциплин войдут в учебный план, а какие будут использоваться для формирования учебных дисциплин, включаемых в учебный план в качестве специальных; дисциплины базовой подготовки рекомендуется включать в учебный план без изменения названий и существенного изменения содержания; 6) распределить весь объем подготовки в области применения вычислительной техники по семестрам с учетом принципа непрерывности подготовки в течение всего периода обучения.

Элементы содержания подготовки студентов вузов в области применения вычислительной техники на уровне учебных дисциплин и их разделов представляют собой учебные модули, каждый из которых с течением времени может совершенствоваться, изменяться или заменяться, что обеспечит учебным планам определенный динамизм и возможность совершенствования в соответствии с изменяющимися условиями и требованиями научно-технического прогресса.

Н.М.Козлов, Ф.И.Перегудов.

НЕПРЕРЫВНОСТЬ ФОРМИРОВАНИЯ КОМПЬЮТЕРНОЙ ГРАМОТНОСТИ — последовательное развитие *компьютерной грамотности* от умений пользоваться компьютером в учебно-познавательской деятельности до умений профессионально решать практические

задачи посредством современных информационно-вычислительных систем. Одна из задач внедрения *компьютерной технологии обучения*.

Н.М.Когдов, А.Я.Савельев, В.А.Стаканов.

НЕЧЁТКАЯ ЗАДАЧА — задача, для которой соответствующая задача контроля является *проблемной задачей*. Иногда задачу называют нечеткой, имея в виду недостаточную определенность не критерия решенности задачи, а самого ее предмета, когда, напр., не вполне ясно, что, собственно, дано в задаче. *Учебная задача* выступает для *обучаемого* как нечеткая, в частности, тогда, когда он не может понять, что, собственно, в ней требуется. Если он неправильно понял задачу, то сформировавшаяся внутренняя задача может быть четкой, но она существенно отличается от предложенной (внешней), т.е. произошло переопределение последней (см. *Доопределение задачи*).

В учебном процессе нечеткость решаемых обучаемым задач, обусловленная то ли неясностью или противоречивостью заданий (вообще, неадекватностью *обучающих воздействий*), то ли несформированностью самоконтроля обучаемых, их неумением оценить успешность своих действий, нежелательна. Иное дело, когда нечеткость учебных задач предусматривается специально для стимулирования их самостоятельного уточнения обучаемыми. Примерами могут служить задачи, предлагавшиеся студентам академиком П.Л.Капицей, в частности, такая: "Перечислите факторы, которые сказываются на точности хода карманных часов. Оцените относительные значения этих факторов"

В условиях *компьютерной технологии обучения* Н.з. нередко предъявляются обучаемым в дидактических целях, тем более что в режиме интерактивного *диалога* с компьютером можно многократно уточнять условие задачи, "снимать" ее нечеткость различными способами.

Г.А.Балл.

НОСИТЕЛЬ ИНФОРМАЦИИ — устройство для записи и хранения *информации*. Н.и. должен обладать способностью изменять локальные физические состояния в соответствии с заносимыми данными, сохранять эти состояния и с помощью определенного механизма позволять воспроизводить ранее занесенные данные. Изготавливают Н.и. из запоминающих элементов или на основе запоминающей среды. Различают Н.и. одно- и многократной записи информации. В Н.и. однократной записи, применяемых во внутренних *запоминающих устройствах* (ЗУ) ЭВМ, запоминающими элементами служат плавкие перемычки или диоды, восстанавливающие связь электрическим пробоем, в Н.и. многократной записи — триггеры или конденсаторы, сохраняющие информацию.

Во внешних ЗУ в качестве Н.и. чаще всего используют *магнитные диски* и ленты. Основа магнитной ленты (полиэтилен-рефтолат) покрыта ферромагнитным слоем. Лента наматывается на катушки или помещается в кассеты (см. *Дискета*), где защищена от механических повреждений, что позволяет изготавливать ее более тонкой. При той же информационной емкости кассеты обладают значительно меньшими габаритными размерами и массой, чем катушки. Магнитная лента допускает плотность записи 32, 126, 356 и т.д. перемагничиваний (изменений направлений намагниченности) на миллиметр. Значение номинальной плотности записи определяется типом ленты и иногда указывается на рулоне.

Перспективны Н.и. на цилиндрических магнитных доменах. Под действием внешнего магнитного поля в материале домены перемещаются; направление намагниченности домена определяют двоичный "0" либо двоичная "1". В отсутствие магнитного поля домены не передвигаются, но и не разрушаются.

Кроме магнитных Н.и., все шире применяются оптические носители — оптические диски, плотность записи на которых в 10—40 раз выше, чем на магнитных дисках. Физической средой, хранящей информацию, здесь служит поверхность жесткого (пластмассового или из алюминиевого сплава) диска. Различают Н.и. термо- и магнитооптические. Термооптические Н.и. бывают с перезаписью (многократной записью) и без перезаписи информации (однократной записью). Записывают информацию мощным импульсным лучом лазера. На Н.и. без перезаписи луч выжигает углубления диаметром приблизительно 0,8 мкм, на Н.и. с перезаписью луч локально изменяет коэффициент отражения поверхности диска. Препятствует коэффициент отражения восстанавливается лучом лазера меньшей мощности. Информация с термооптического диска считывается лучом лазера такой мощности, которая не приводит к изменениям оптических свойств поверхности диска. Магнитооптические Н.и. бывают с перезаписью информации. Запись производится в магнитном поле. При этом перемагничиваются только те локальные участки диска, которые прогреваются лучом лазера. Информация считывается поляризованным маломощным лучом лазера. Магнитное поле поворачивает плоскость поляризации света, и поэтому плоскость поляризации луча лазера, отраженного от различных участков диска, поворачивается на различные углы. Изменение яркости отраженного термооптическим Н.и. или поворот плоскости поляризации отраженного магнитооптическим Н.и. луча улавливается электронной аппаратурой и преобразуется в электрические сигналы.

В.П.Козлова, В.Л.Леонтьев, З.Ш.Примакова, В.И.Сигалов.

ОБРАТНАЯ СВЯЗЬ В ОБУЧЕНИИ — механизм, обеспечивающий взаимосвязь между обучающей и учебной деятельностью. Трактовка

О.с.в о., ее структуры и функций существенно зависит от концепции *обучения*. Так, в рамках бихевиористических концепций обучения эта связь рассматривается как осн. механизм *учения*, выражающийся в подкреплении правильного *ответа*. В когнитивной психологии О.с.в о. и коррекция рассматриваются как *информация*, которая способствует учению, но не сводится к поощрениям. Осн. функции О.с.в о.: информирование *обучаемого* о допущенной ошибке; оказание ему помощи для устранения ошибки; повышение *мотивации* обучаемого с помощью суждений о результатах его деятельности. В зависимости от преобладания тех или иных функций различаются внешняя обратная связь и знание своего результата (условно — внутренняя обратная связь).

Е.Д.Маргулис.

ОБСИС (ОБУчающая СИСтема) — *авторская система*, основанная на страничной (блочной) разбивке учебного материала. Разработана в 1984-88 в Болгарии. В ОБСИСе различают пять типов страниц: информационные, графические, вопросы, задачи, группы. Информационные страницы предназначены для записи теор. и методических указаний. Объем каждой такой страницы определяется объемом *информации*, представленной на экране *дисплея*. Графические страницы играют вспомогательную роль и предназначены для представления графической информации, иллюстрирующей содержание страниц др. типов, к которым они присоединяются. В страницах типа “вопрос” *автор* задает текст *вопроса*, правильные и ожидаемые неправильные *ответы*. Страницы типа “задача” предназначены для формирования *задач*, задаваемых *обучаемому*. Каждая задача состоит из *текста*, параметров и вопросов. С задачей ассоциируются *программы*, описывающие ее решение. Страницы типа “группа” предназначены для объединения нескольких подобных вопросов или задач. Использование этого типа страниц позволяет осуществлять случайную выборку вопросов и/или задач, задаваемых группой.

Особенностью системы ОБСИС является то, что автор освобождается от необходимости писать программу, реализующую *стратегию* предъявления учебного материала. Вместо этого он задает ассоциативные связи между созданными страницами, которые и определяют требуемую стратегию. В результате один и тот же учебный материал можно использовать для реализации самообучения, экзамена и др. режимов *обучения*. ОБСИС реализована в системе ТУРБОПАСКАЛЬ и функционирует под управлением *операционной системы* MS DOS на компьютере типа IBM PC и совместимых с ними (напр. Правец-16).

О.К.Ковачев, Б.К.Шишеджиев.

ОБУЧАЕМЫЙ — тот, на которого направлен процесс обучения; в условиях компьютерной технологии обучения — непосредственный пользователь *автоматизированного учебного курса (АУК)*. При работе с АУК О., как правило, может управлять последовательностью его изучения, посылать при необходимости сообщения *автору* курса (эти сообщения помещаются в архив, откуда *диспетчер* может вывести их на алфавитно-цифровое *печатающее устройство* и передать автору). При этом О. может работать как индивидуально, так и в условиях класса при групповых занятиях (см. *Автоматизированная обучающая система*). В любом случае АОС обеспечивает независимую работу всех О. в индивидуальном темпе с каждым АУК, имеющимся в ее библиотеке.

О.П.Платонова.

ОБУЧАЕМЫХ КАТЕГОРИИ — группы *обучаемых*, объединенных на основе общности модели *знаний*. Отнесение обучаемого к той или иной категории должно быть основано на выявлении его психолого-дидактических параметров, включающих: индивидуальные особенности (характер, память, реакцию, восприимчивость к новой информации и др.); интеллектуальные возможности; творческие способности; систему общеобразовательных и специальных знаний; эрудицию; заинтересованность в результатах обучения; профессию (для учеников школ — предположительную будущую профессию). Эти параметры являются содержательным описанием *модели знаний* обучаемого. В *учебной информатике*, напр., можно выделить три О.к.: обучаемые, осн. объектом и продуктом профессиональной деятельности которых являются программные средства (профессиональные программисты); обучаемые, использующие *программы* как *средство решения* задач (функциональные специалисты); обучаемые, которые могут использовать компьютер на досуге, в быту и, возможно, в своей профессиональной деятельности, но чисто потребительски, не вникая в сущность и устройство компьютера, структуру *программного обеспечения* и т.д. (массовые, или конечные, пользователи).

В.Б.Доронин.

ОБУЧАЮЩАЯ ИГРА — *учебная игра с обучающим воздействием*. Представляет собой *программу действий преподавателя* или компьютерной *обучающей программы*, включающую учебную игру и ее психолого-педагогическое обеспечение — систему *обучающих воздействий*, построенную с учетом индивидуально-психологических особенностей *обучаемого* и предыдущих результатов игры. Таким образом, О.и. нужно рассматривать как совокупность учебной игры и соответствующей обучающей системы. Учебная игра может

входить в эту систему и рассматриваться как учебная задача особого рода или как особое обучающее воздействие.

Э.Л.Ивахненко, Е.Д.Маргулис, В.Цонева.

ОБУЧАЮЩАЯ ПРОГРАММА — прикладная программа учебного назначения (ПУН), в которой описываются подлежащие усвоению знания, умения и навыки, а также способы их формирования. Первоначально О.п. составляли основу *программированного обучения*. Ныне под О.п. зачастую понимают и др. прикладные ПУН, поддерживающие функцию обучения (*автоматизированные учебные курсы, учебные игры и т.п.*). О.п., реализуемые в *автоматизированных обучающих системах*, состоят из ряда фрагментов, каждый из которых содержит некоторую порцию учебного материала, отдельный вопрос, задачу или к.-л. иное задание, выполняемое *обучаемым*. Кроме того, в О.п. описывается способ *анализа ответов обучаемого* и средства сбора информации о результатах работы обучаемого с программой (истории обучения). Статистика о ходе обучения накапливается и обрабатывается пофрагментно, что позволяет учитывать качество усвоения учебного материала с целью корректировки процесса обучения. Для создания О.п. используются как универсальные языки программирования, так и специальные средства, напр., *авторские системы*.

С точки зрения структурных характеристик О.п. условно разделяют на линейные и разветвленные. В линейной О.п. фиксированные порции учебного материала объединены в виде последовательности уроков, порядок прохождения которых не зависит от результатов *контроля знаний* на предыдущем этапе. В разветвленных О.п. предъявление обучаемому очередной порции учебной информации зависит от истории обучения — степени усвоения материала на предыдущих этапах, количества и характера ошибок и т.п. В общем случае разветвленную О.п. можно представить как совокупность линейной О.п., содержащей осн. учебную информацию, и добавочных порций, выдаваемых обучаемому при наличии ошибок. С точки зрения функциональных характеристик различают адаптивные и неадаптивные О.п. Адаптивная О.п. обеспечивает возможность изменять способы изложения учебного материала в зависимости от изменений внутренних и внешних условий обучения (степени и скорости усвоения предыдущего материала, индивидуальных психофизиологических качеств обучаемого и т.п.). См. также *Педагогический программный продукт*.

А.Ф.Манако, Е.М. Сеница.

ОБУЧАЮЩЕЕ ВОЗДЕЙСТВИЕ — 1) Акт управления процессом обучения, 2) Составная часть метода обучения (способа управления учебной деятельностью). Различают О.в. основные (*учебные задачи*)

и вспомогательные (подзадачи, указания, наводящие вопросы, демонстрация фрагмента деятельности и т.д.). Вспомогательные О.в. выдаются *обучаемому* при возникновении у него затруднений. Наиболее важная характеристика О.в. — направленность: одни непосредственно устраняют ошибки, другие ориентируются на устранение причин, приводящих к этим ошибкам. Первый случай часто соотносят с прямым, а второй — с непрямым управлением учебной деятельностью. По модальности О.в. варьируются от прямых указаний и предписаний (преимущественно в первом случае) до пожеланий и эвристических рекомендаций (во втором случае). Анализ О.в., по мнению психолога Е.И.Машби относится к формальной стороне метода обучения и предполагает выяснение особенностей постановки (явной или неявной) учебных задач и вопросов, а также установление необходимой меры помощи при возникновении затруднений у обучаемых.

Е.Д.Маргулис.

ОБУЧАЮЩИЕ СИСТЕМЫ — программы учебного назначения, реализующие функцию управления познавательной деятельностью *обучаемого*. Различают тренирующие, собственно обучающие и сопровождающие О.с.

В тренирующих О.с. обучаемый получает знания, навыки и умения, необходимые для выполнения определенной работы, а учебные программы организуют адекватную среду для приобретения и закрепления требуемых навыков и умений. К данному типу программ относятся *тренажеры* и многие *учебные игры*. Характерные черты программ этого типа: наличие как формальной, так и аудиовизуальной модели изучаемого предмета; явное знание целей функционирования системы; наличие контроля действий обучаемого; наличие, как правило, жестких ограничений на время, выделяемое обучаемому для достижения поставленной цели.

В собственно обучающих системах обучаемый получает знания, навыки и умения в некоторой предметной области под управлением учебной программы; при этом программа принимает на себя все функции преподавателя по организации предъявления учебного материала, контроля его усвоения и диагностики ошибок обучаемого. Данный тип программ включает обучающе-контролирующие, управляющие, экспертные обучающие системы и частично учебные игры. Он характеризуется: наличием цели обучения; реализацией некоторого метода обучения, приводящего к достижению требуемой цели и определяющего стиль общения с обучаемым; комплексным решением задач учения, контроля и диагностики.

В сопровождающих О.с. пользователь работает в некоторой системе обработки информации, а программа следит за его деятельностью с целью оказания ему помощи при обнаружении

ошибочных или нерациональных действий. Сопровождающая система как бы “заглядывает через плечо” пользователя на экран, пытается понять, что он делает, оценить, как он это делает, и решить, стоит ли ему помочь, а если стоит, то как эту помощь предоставить. Программы этого типа, как правило, принадлежат к интеллектуальным О.с. Они имеют много общего с предыдущим типом программ, но отличаются тем, что не знают цели деятельности пользователя и вынуждены ее прогнозировать; менее коммуникабельны, чтобы не отвлекать пользователя от работы.

По гибкости взаимодействия с обучаемым различают О.с. с жестким сценарием и экспертные (интеллектуальные). Первые, как правило, являются порождением современных *автоматизированных обучающих систем* (напр., АОС-ВУЗ, АОС-М) и, несмотря на существенные недостатки, широко используются в учебном процессе, в то время как последние только начинают представлять практический интерес.

В.А.Петрушин.

ОБУЧЕНИЕ — целенаправленный вид деятельности, осуществляемой организатором учебного процесса (*преподавателем, обучающей программой*). Осн. дидактическими категориями, характеризующими О., являются *принципы обучения*, процесс О. (целенаправленный процесс взаимодействия преподавателя с обучаемым, в ходе которого осуществляются образование, воспитание и развитие обучаемых), содержание (система *знаний, умений и навыков*, которыми закладываются основы для формирования и развития личности человека), средства (все, что используется в учебном процессе для повышения его эффективности), организационные формы О. (отражающие особенности объединения обучаемых для организуемых преподавателем занятий, в процессе которых и совершается учебно-познавательная деятельность) и др.

О. можно описать как совокупность двух подсистем — обучающей и учебной деятельности. При этом первая подсистема выступает как управляющая по отношению ко второй. Механизмы О. осуществляют связь двух выделенных подсистем, т.е. обеспечивают функционирование системы О. в целом. Обучающую деятельность осуществляет педагог (или более широко обучающая система, в т.ч. и компьютер), учебную деятельность — обучаемый. Цели обучающей деятельности направлены на то, чтобы обеспечить достижение целей учебной деятельности. Но на цели обучающей деятельности значительное влияние оказывают и остальные цели педагога (проверить эффективность новой методики, достичь высокого статуса в педагогическом коллективе, добиться уважения со стороны обучаемых и пр.), а также цели самих обучаемых (могут не совпадать с учебными). Обучающая деятельность осуществляется посредством

обучающих воздействий. Имеются различные аспекты О.: как передачи знаний (опыта, культуры), как ссобой формы групповой (кооперативной, совместной, совместно-распределенной) деятельности обучающего и обучаемых, как решения *задач обучения*, как управления учебной деятельностью и т.д. Системное понимание О., как отметил укр. психолог Е.И.Машбиц, должно включать все эти аспекты.

Средства О., как и *средства решения задач* вообще, могут быть материальными объектами (орудия труда, *технические средства обучения*), материализованными (*знаки, схемы, чертежи* и пр.), идеальными (образы объектов, различные знания и пр.). Иногда идеальные средства разделяют также на знаковые средства (конкретные *знаки, модели*) и операциональные схемы действий, формулировки, приемы работы и пр. Средства деятельности различаются по ряду направлений: отношению к применяющему их субъекту, полноте охвата его функций, природе объектов, используемых в качестве средств, а также по видам опосредуемой деятельности. Совокупность средств О. и способов их применения характеризует *технологию обучения.*

При рассмотрении требований к психологической теории О. в контексте проектирования (см. *Проектирование психолого-педагогической*) важное значение имеют такие из них: 1) теории О. должны быть не только описательными (дескриптивными), но и предписывающими (прескриптивными); 2) указания (предписания) должны даваться в форме, допускающей их технологизацию; 3) теории О. и *учения* должны быть сопряжены, т.е. каждая из них должна относиться к деятельности обучающего и обучаемых, причем эти виды деятельности следует рассматривать во взаимодействии (Е.И.Машбиц).

В связи с *компьютеризацией* процесса О. резко возросли требования к технологизации научных знаний, в т.ч. знаний о процессе О. Возникла, в частности, необходимость пересмотра и уточнения многих традиционных психолого-педагогических и дидактических *понятий*, с тем чтобы они могли быть использованы на технологическом уровне, в частности, понятия метода обучения или способа управления учебной деятельностью. Примером, иллюстрирующим роль проблем технологизации О. как связующего звена между теор. концепциями и практикой О., может служить также подход к автоматизации учебного процесса, основанный на теории поэтапного формирования умственных действий. Согласно этому подходу процесс усвоения новых видов познавательной деятельности состоит из ряда качественно своеобразных этапов, определяющих конкретные требования к деятельности преподавателя и обучаемого, причём на каждом этапе имеются особые предпосылки для автоматизации деятельности. Осн. момент перехода от теор. проблем к технологическим связан с выявлением и анализом функций

преподавателя и обучаемых, характеристикой обучающих программ посредством описания функций, которые автоматизируются. Исходные теор. положения в контексте данного подхода позволяют выделить следующие осн. функции преподавателя: 1) предъявление и объяснение обучаемому усваиваемого материала; 2) показ деятельности, в которой обучаемый должен использовать этот материал; 3) управление процессом усвоения знаний обучаемым. Соответственно, деятельность преподавателя должна обеспечивать: а) создание положительных мотивов к учению, объяснение, показ и фиксацию формируемой деятельности и входящих в неё знаний; б) организацию и контроль деятельности обучаемых; в) передачу машине рутинной части учебной деятельности; г) составление и предъявление учебных знаний, соответствующих разным этапам усвоения, а также индивидуальным особенностям обучаемого и состоянию всей деятельности в данный момент. Однако реализация всех этих функций обеспечивается функционирование лишь осн. части обучающей программы. Необходимым является и получение сведений о состоянии познавательной деятельности в процессе О. Таким образом, как подчеркивает один из авторов теории поэтапного формулирования умственных действий академик Н.Ф.Талызина, эффективное О. невозможно без создания программы коррективки, реализующейся в зависимости от особенностей деятельности обучаемого. Эта программа является средством реализации теор. положений и осуществления *обратной связи в обучении*. Возникают здесь и специфические задачи, связанные с получением сведений, их фиксацией, обработкой и хранением, выработкой корригирующих воздействий, их реализацией и пр.

Е. Д. Маргулис.

ОБУЧЕНИЕ АЛГОРИТМИЗАЦИИ ЗАДАЧ — целенаправленное формирование у *обучаемого* математических понятий дискретного характера. Для автоматизированного О.а.з. наиболее широко распространены обучающие курсы с линейной последовательностью изучаемых разделов и разветвленные внутри каждого раздела. Осн. дидактические цели таких курсов: научить расчленять некоторый процесс на элементарные *операции* в заданной конечной системе операций и выстраивать элементарные операции в цепочку последовательных и/или повторяющихся операций так, чтобы их применение давало решение *задачи* за конечное число шагов. На основе сформулированных целей уточняется предметное содержание *обучения*, т.е. определяются базовые компоненты *знаний* и умений. При этом выделяются следующие разделы предметного содержания: интуитивное понятие *алгоритма*, способы описания алгоритмов, алгоритмические операции и алгоритмические приемы.

Интуитивное понятие алгоритма раскрывается на примере расчленения процесса решения неформализованной задачи на элементарные действия, описываемые математически. Уточняется это понятие на примере решения задачи, достаточно простой, но позволяющей явно проиллюстрировать особенности конструирования алгоритма. Излагаются также два осн. способа описания алгоритмов: словесного описания и блок-схемного описания. Подчеркивается разница между описанием алгоритма и процессом его реализации. Для конструирования алгоритмов вводится система алгоритмических операций, обладающих свойствами: а) алгоритмической полноты; б) методической целесообразности; в) отражения характера решаемых задач. В алгоритмически полную систему операций входят: присваивание, условный переход, или разветвление, останов. С учетом методической целесообразности операция присваивания расчленяется на три класса: установочное присваивание (установление начального значения заданным объектам); рекурсивное присваивание (приращение значений заданных объектов); формульное присваивание (вычисление по выражениям). Решение конкретных практических задач определяет необходимость включения в систему таких алгоритмических операций, как ввод исходных данных и вывод данных. Требование отражения характера решаемых задач выражается во введении в алгоритмические операции конкретных действий над данными. Для вычислительных задач это могут быть алгебраические действия, для задач обработки символьной информации — действия над строками и т.д. Усвоение алгоритмических операций, осуществляемое под управлением компьютера, не является достаточным для составления алгоритмов решения задач, а представляет собой подсистему знаний, необходимую для перехода к следующей стадии обучения. В множество базовых алгоритмических приемов, которыми должны владеть программисты начального уровня подготовки, как правило, включаются: табулирование функций, вычисление сумм и произведений, простейшие методы сортировок, обработка матриц, простейшие итерационные вычисления. Изучение таких алгоритмических приемов осуществляется на второй стадии обучения и основывается на использовании ставшей уже известной для обучаемого системы алгоритмических операций. Процедура обучения на этой стадии включает такие этапы: иллюстрацию процесса построения алгоритмов (синтез алгоритмов); прослеживание динамики работы алгоритмов (анализ алгоритмов); контроль знаний и умений обучаемого в построении алгоритмов. На этапе синтеза алгоритмов компьютер формирует некоторую учебную задачу, решение которой сводится к применению одного из перечисленных алгоритмических приемов, осуществляется построение плана решения задачи, а обучаемый имеет возможность отслеживать этот процесс с целью оценки собственных аналогичных действий при самостоятельном решении задачи, т.е. на

данном этапе компьютер формулирует задачу; обучаемый формирует алгоритм решения; компьютер иллюстрирует процесс формирования алгоритма; обучаемый соотносит свои действия с действиями компьютера. На этапе анализа алгоритмов осуществляется обучение пониманию работы алгоритма. При этом обучаемый прослеживает динамику функционирования алгоритма, предложенного компьютеру для различных учебных задач. Задачи отличаются исходными данными, которые определяют различные последовательности применения операций, составляющих алгоритм. Этап контроля знаний реализуется посредством составления алгоритмов, предложенных обучаемому и включающих базовые алгоритмические приемы. Анализ правильности решения осуществляется компьютером путем сравнения с имеющимися в его распоряжении эталонами.

Достижение целей взаимодействия в режиме О.а.з. и овладение алгоритмическими приемами позволяет перейти к освоению следующего этапа решения задач с помощью компьютера — программированию.

М.А.Краснов.

ОБУЧЕНИЕ ИНОСТРАННОМУ ЯЗЫКУ С ПОМОЩЬЮ КОМПЬЮТЕРА. Традиционной формой компьютерного преподавания иностранного языка являются *автоматизированные учебные курсы*, функционирующие на базе универсальных *автоматизированных обучающих систем*. Их действие основывается на жестких, заранее проработанных сценариях общения *обучаемого* с компьютером. Эти сценарии однозначно реализуют методику своих разработчиков и не допускают варьирования и развития со стороны *преподавателя*. Альтернативой этому подходу является применение в целях преподавания систем автоматической переработки *текста* (АПТ), в частности систем *машинного перевода* (МП), предлагающих широкие возможности с точки зрения методического творчества преподавателя, свободы общения с компьютером самого обучаемого и применения новых технологий. Ядро каждой системы АПТ составляет *автоматический словарь* (АС), выступающий как в роли базы орфографических и лексико-грамматических *данных*, так и в функции базы энциклопедических *знаний*. Простейшее использование АС для нужд преподавания осуществлено в Самаркандском университете, где применяется французско-русская версия системы МП СИЛОД (для ЕС ЭВМ). Здесь компьютер задает студенту формы французских слов (напр., прилагательных) вместе с заданием образовать от них производные слова (напр., наречия). Если задания выполнены правильно, то система, обратившись к АС, выдает русский перевод формы, образованной студентом. Если же студент выполнил задание неверно или допустил орфографическую ошибку, то предложенная им словоформа не будет найдена в словаре либо переведена

лексически или грамматически неверно. Аналогично используются системы исправления орфографических и грамматических ошибок, разрабатываемые фирмой IBM и японскими исследовательскими группами. Напр., с помощью англо-японского АС и *морфологического анализа* входного текста выявляются грамматические и орфографические ошибки, после чего обучаемому выдается перевод.

Подготовка специалистов, способных переводить и реферировать иноязычные тексты на современном уровне, требует использования уже не АС, а действующих систем МП и автоматического реферирования текста. Простейший способ применения систем МП в учебных целях — постредктирование пословно-пооборотного или лексико-грамматического перевода, выдаваемого компьютером. Этот подход позволяет проверить понимание обучаемым иноязычного текста и вырабатывать редакторские навыки в области родного языка. Подобным образом используется система ПЛАСТ. Однако постредктирование для непрофессионального переводчика (напр., школьника, студента или аспиранта) является сложной задачей, т.к. современные системы автоматического МП ориентированы либо на хорошее знание конкретной *предметной области*, либо на профессиональное владение языком. Поэтому для облегчения работы обучаемого при выполнении технического перевода система МП должна иметь дополнительные сервисные средства. Примерами таких средств служат компьютерное пособие по грамматике родного языка, автоматический корректор ошибок, средства для постредктирования перевода. В этом направлении адаптируется в ЛГПИ им.А.И.Герцена для нужд преподавания система *MULTIS*. На первом этапе создается автоматизированная лексико-грамматическая служба, состоящая из следующих функциональных блоков: морфолого-синтаксическая справка, содержащая конкретную грамматическую *информацию* по данной словоформе, информацию о возможностях актуализации данной лексической единицы в тексте (это позволяет устранять омонимичные конструкции); таблицы, дающие обобщенную информацию по затрагиваемой грамматической категории; терминологический справочник с пояснениями значения терминов и краткими сведениями по грамматике. Это справочное пособие использует в основном грамматическую информацию, содержащуюся в словарной статье автоматического словаря системы.

К.Р.Пиотровская

ОБУЧЕНИЕ НА БАЗЕ КОМПЬЮТЕРА — то же, что и *компьютерное обучение*.

ОБУЧЕНИЕ ОТЛАДКЕ ПРОГРАММ — завершающий этап *процесса обучения* практическому *программированию*; характеризу-

ется невозможностью его осуществления без использования реального компьютера, с помощью которого у *обучаемого* вырабатываются и закрепляются умения и навыки по *отладке программ*. Использование компьютера при О.о.п. может носить двоякий характер — в качестве рабочего инструмента и как обучающего устройства, совмещенного с рабочим инструментом отладки программ. В первом случае компьютер играет пассивную роль, и обучающие свойства его *программного обеспечения* определяются понятностью и точностью диагностических сообщений, выдаваемых при обнаружении ошибок в отлаживаемых *программах*. Взаимодействие между человеком (обучаемым) и машиной (компьютером) характеризуется тем, что машина, выполняя программу, переданную ей человеком, предъявляет ему результат выполнения, соответствующий данному варианту программы и определенному набору входных *данных*. На основании соотношения полученного и ожидаемого результатов выполнения программы человек формирует и задает машине *процедуру* управления отладкой — D , которая, с точки зрения задачного подхода к анализу взаимодействия в человеко-машинных системах, трактуется парой $D = (\{d_i\}, R_d)$, где $\{d_i\}$ — конечное множество отладочных операций, выполняемых машиной, R_d — отношение следования, определяющее последовательность выполнения операций $\{d_i\}$. Множеству $\{d_i\}$ принадлежат *операции* запуска программы на выполнение, отслеживание хода выполнения программы, отслеживание динамики изменения переменных, контроля индексов массивов путем сопоставления с объявленными границами, изменения извне текущих значений *данных* программы, корректировки *текста* программы и *данных* и др. При таком взаимодействии с компьютером собственно *обучение* осуществляется традиционными методами — изучением документации по системе отладки и консультаций у более опытных программистов, которые зачастую навязывают обучаемому свои навыки формирования процедур управления отладкой, возможно далекие от совершенных.

При использовании компьютера как автоматизированного обучающего устройства, совмещенного с рабочим инструментом отладки, его роль становится активной и заключается в управлении действиями обучаемого при отладке программ. Автоматизированная система обладает *алгоритмами* формирования процедур управления отладкой, которые отслеживают действия обучаемого и корректируют их при обнаружении рассогласования между фактическими действиями обучаемого и реализуемой в данный момент процедурой управления отладкой. Наличие развитой номенклатуры отладочных операций $\{d_i\}$ требует знания обучаемыми их возможностей и условий применения, что обуславливает включение в структуру системы, кроме процедур управления отладкой, также информационно-справочных и обучающих режимов по усвоению операций $\{d_i\}$.

Дидактические цели автоматизированного О.о.п. определяются необходимостью раскрыть осн. принципы отладки, указать влияние этих принципов на процесс разработки программ и качество программного продукта, показать пути практической реализации технологического подхода к отладке программ и обеспечить обучение осн. приемам и методам, применяемым на каждом этапе отладки программ.

Учебно-методическое и программное обеспечение для автоматизированного О.о.п. проектируются так, чтобы, с одной стороны, удовлетворить цели обучения, а с другой — учитывать практические потребности обучаемого по выполнению и отладке программ. Автоматизированные курсы для О.о.п. обычно создаются на базе интегрированных пакетов прикладных программ АОС, для которых обучающая часть реализуется средствами пакета автоматизированной обучающей системы, а исполнительная часть — средствами сопрягаемого пакета, представляющего собой диалоговую отладочную систему. Характерным примером реализации такого курса служит режим ОТЛАДКА, входящий в структуру многофункциональных автоматизированных учебных курсов АФРОДИТА и АИДА, созданных на базе интегрированного пакета АОС-ВУЗ/ПРИМУС.

В.Д.Рынгащ.

ОБУЧЕНИЕ С ПОМОЩЬЮ КОМПЬЮТЕРА — то же, что и компьютеризированное обучение.

ОБУЧЕНИЯ ФОРМАЛИЗАЦИЯ — формальное представление процедуры обучения. Обозначим эту процедуру через $\Omega_{об}$, а ее модель — обучающую программу — через $\Omega_{об}^*$.

Характерным примером обучающей программы $\Omega_{об}^*$ являются программы для ЭВМ, написанные на одном из языков программирования, ориентированных на автоматизацию обучения. Операторами $\{\omega^*\}$, составляющими $\Omega_{об}^*$, в этих языках являются операторы выдачи текстов (вопросов, команд, учебных задач), операторы приема ответов обучаемого, операторы контроля ответов (напр., оператор сравнения ответа с эталонным правильным ответом) и др. Примеры таких программ приведены в приложениях. Другим видом $\Omega_{об}^*$ являются обучающие программы, записанные на естественном языке и представляющие собой квазиалгоритмы, выполняемые человеком (напр., обучаемым), работающим с программированным учебником. И в первом, и во втором случае обучающая программа соответствует расширенному понятию программы решения абстрактной задачи.

А.М.Довгялло.

ОБЩЕНИЕ ЧЕЛОВЕКО-МАШИННОЕ — специфическое, опосредствованное компьютером общение между двумя субъектами — *пользователем* программных средств компьютера и программистом, создавшим эти средства. В психологии под общением понимают сложный многоплановый процесс установления и развития контактов между людьми, порожденный потребностями совместной деятельности и включающий обмен *информацией*, выработку единой *стратегии* взаимодействия, восприятие и понимание другого человека. Коммуникативный компонент общения предполагает обмен определенными результатами психической деятельности — усвоенной информацией, мыслями, суждениями, оценками, чувствами, установками. В интерактивном аспекте общение представляет собой средство, обслуживающее совместную деятельность субъектов. При этом оно само выступает как самостоятельная и специфическая форма активности субъекта, направленная на формирование субъект-субъектных отношений.

Традиционен также взгляд на общение как особую форму взаимодействия. Хотя не существует четких определений общения и взаимодействия, позволяющих однозначно дифференцировать эти понятия (зачастую они определяются друг через друга или синонимично), предполагается, что понятие взаимодействия носит более технический или инструментальный характер и выполняет функцию регуляции совместной деятельности, будучи лишенным аффективно-перцептивного аспекта общения. В процессе взаимодействия в результате воздействия субъектов друг на друга достигаются приспособление действий одного к действиям другого, *общность* в понимании ситуации, смысла действий и определенная степень солидарности или согласия между ними. Еще более очевидна несинонимичность понятий “общение” и “диалог”, хотя распространена трактовка *диалога* как формы речевого общения. В этой связи диалог представляет собой взаимодействие не субъектов, а некоторых выражаемых ими (или им, если это внутренний диалог субъекта) смысловых позиций относительно темы диалога.

Миним. единицей процесса общения можно считать реплику, сообщение или высказывание. Семантическое разграничение этих понятий весьма сложно. Реплика всегда отражает реакцию субъекта, в отличие от сообщения или высказывания, которые могут быть самоиницируемыми; кроме того, реплика носит более эмоциональный характер по сравнению с сообщением, имеющим больший содержательный смысл. В свою очередь, высказывание всегда имеет определенное лингвистическое оформление, в отличие от реплики и сообщения, которые можно выразить невербальными средствами (жест, мимика).

О.ч.-м., с одной стороны — это “отсроченное” общение, т.к. программист в реальном диалоге физического участия не принимает, ограничив свою роль проектированием реакций системы на возможное

развитие диалога. С другой стороны, возможности технического посредника чрезвычайно велики, так что он получает возможность автономного поведения, т.е. ему передаются определенные полномочия в *принятии решений*, что до некоторой степени делает его субъектом (точнее, квазисубъектом) общения. Понятие О.ч.-м. развивалось и углублялось по мере развития его средств: материальных (технические, аппаратное обеспечение) и идеальных (*программное обеспечение, язык*). Первоначально все "общение" сводилось к одностороннему и жесткому воздействию человека на ЭВМ (ввод данных, обсчет их по "защитым" в памяти ЭВМ программам, вывод результатов). При появлении *машинно-ориентированных языков* ассемблерного типа, впервые реализующих собственно концепцию *программирования*, возникает т.н. квазиобщение, предусматривающее, напр., обмен сообщениями о результатах диагноза или выявления ошибок. В дальнейшем с созданием *языков программирования* высокого уровня (см. *Уровень алгоритмического языка*), появляется упрощенный вид общения, ограниченный контролем и коррекцией синтаксической правильности программ; тогда же возникает проблема понимания, сводящаяся к способности корректно анализировать синтаксис сообщения. В техническом плане возможности общения облегчили клавиатурный ввод и дисплей.

В настоящее время создание новейших лингвистических средств, позволяющих реализовать *взаимодействие* *человеко-машинное* на *естественном языке ограниченном*, и чрезвычайно комфортных *устройств ввода-вывода информации* сочетается с возникновением содержательного общения (что реализуется, напр., в *экспертных системах*) и вызванной этим проблемой содержательного анализа и понимания системой сообщений *пользователя*.

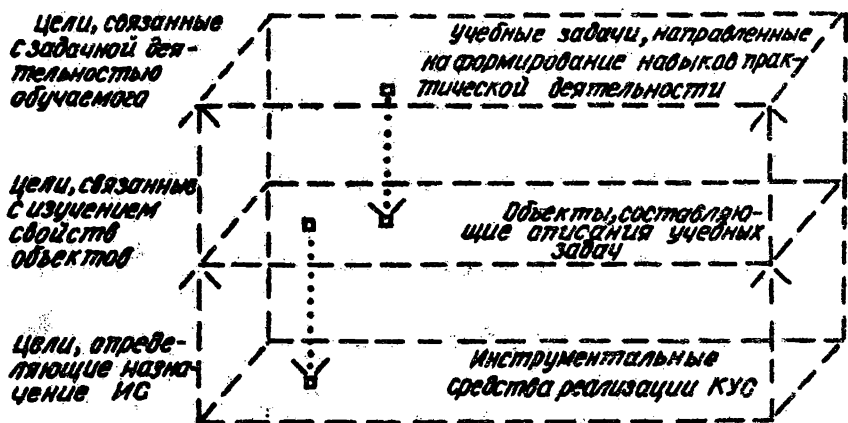
Е.Ю.Ксмисарова.

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ СРЕДСТВА поддержки учебного диалога — совокупность программно-педагогических средств, обеспечивающих учебное взаимодействие *обучаемого* с компьютером. Организация обучения с применением современных информационных технологий по *предметным областям* (ПО), связанном с практической деятельностью, требует реализации специальных *компьютерных учебных сред* (КУС), позволяющих воспроизводить операционную обстановку, соответствующую реальному производственному процессу. В такую учебную среду включают подходящую *имитационную модель* изучаемого процесса и на ее основе реализуют обучение. Однако имитационные модели входят только в предметную компоненту учебного процесса, ход которого задается последовательностью учебных целей как *обучаемого*, так и *преподавателя*. Реализация на основе *компьютерной технологии обучения* учебных систем моделирования различных процессов основывается на применении

КУС как средства обеспечения процесса совместного решения учебных и практических задач. Обучаемый и КУС при этом рассматриваются как *решатели*. Собственно учебная практическая деятельность основывается на классификационной схеме объектов изучаемой ПО. Классификационная схема строится в процессе изучения объектов ПО на основе ее *тезаурусно-сетевой модели* (ТСМ). Применение ТСМ позволяет на этапе планирования объектно-ориентированной учебной деятельности провести выделение множества элементов опорных понятий (МЭОП) по конкретной ПО.

Реализация в среде КУС структуры учебной деятельности массового пользователя основывается на следующих принципах: 1) программно-информационные средства должны быть ориентированы на структуру процесса решения задачи; при этом операции, применяемые к объектам задачи, должны быть хорошо описаны и понятны обучаемому; 2) обучаемому должна быть предоставлена возможность свободного обращения с объектами, используемыми в процессе решения задачи, при миним. использовании дополнительных языковых средств для описания объектов в КУС и применяемых к ним операций; 3) необходимо по возможности обеспечить в КУС конструирование обучаемым различных моделей изучаемых процессов; 4) учебные модели объектов должны быть адекватны уровню подготовленности обучаемых. Проектирование и описание структуры учебной и практической деятельности основывается на применении МЭОП и анализе связей и отношений между объектами ПО. Для этого осуществляется согласование множества целей, стоящих перед обучаемым, с множеством отношений и связей, описанных в ТСМ ПО. Основу ТСМ составляют родо-видовые отношения, функциональные свойства объектов и структурные отношения между объектами ПО. Дополнительно в нее включаются цели, связанные с выработкой у обучаемого способов действий по решению задач из данной ПО. Сюда следует отнести умения по корректной формулировке задачи из заданной ПО, построению плана ее решения, выделению необходимых свойств объектов, составляющих процесс решения задачи, и т.п.

Структура учебной деятельности реализуется в КУС на основе описания уровня взаимодействия с обучаемым. В этих целях можно использовать *многоуровневую модель обучающего диалога* (см. также *Учебный диалог*). Она базируется на выделении осн. классов целей взаимодействия (рис.). Эти классы целей иерархически связаны, что обеспечивает переход от одного класса целей к другому. Совместная учебная практическая деятельность в КУС основывается на описаниях совместно решаемых учебных практических задач по профилю изучаемой ПО; описаниях объектного представления модели предметной области; реализующих детализацию учебной деятельности обучаемого; выделении инструментальных средств (ИС), в среде



которых обеспечивается представление моделей объектов, включение их в описания наборов учебных задач и организация обучающего диалога. Выделение в структуре диалогового взаимодействия соответствующих уровней позволяет зафиксировать различные виды деятельности пользователя-обучаемого как решателя. Так, начало взаимодействия определяется профессиональными целями. Сюда следует отнести вопросы, связанные с формулировкой задачи, с определением терминологии, составляющей лексическую основу формулировки задачи и описания ее объектов, со структурой представления данных (исходных, промежуточных, результирующих). Точность формулировки задачи во многом зависит от квалификации пользователя, от того, как и какими средствами была определена и построена модель ПО задачи. Довольно распространенный способ построения модели — применение аналогии с уже построенными моделями. Дальнейшее диалоговое взаимодействие пользователя-обучаемого и КУС предполагает построение плана решения задачи. Диалог должен способствовать планированию действий партнеров, выработке общих правил, навыков и способов их применения в конкретной ситуации и организовывать эти правила в осознанную и выразимую структуру деятельности. Реализация обучающего диалога основывается на включении в деятельность пользователя соответствующих классов учебных задач. Описания учебных задач строятся при помощи моделей объектов изучаемой ПО. Построение модели должно вестись совместно в диалоге, т.к. КУС и обучаемый обладают различными средствами и возможностями по ее описанию. По этой причине обучаемый должен описывать объекты задачи средствами языковой среды, поддерживающей диалог. При этом в процессе описания возможны уточнение и переформулировка задачи в терминах выбранной или описанной модели ПО. Достичь этих

целей можно при условии, что в среде обучающего диалога реализуется: кооперативное распределение функций и подзадач, обеспечивающих успешное решение прикладных и учебных задач; целенаправленное управление деятельностью пользователя в процессе решения указанных задач. Иными словами, собственно обучающий диалог должен быть направлен на осознанное усвоение новых знаний и приобретение на их основе необходимых навыков по решению соответствующего класса задач. В этом случае эффективность решения учебной и/или познавательной задачи можно оценить по факту перехода для обучаемого данной прикладной задачи в рутинную. Продуктивность организации обучающего диалога определяется тем, что в его среде обеспечивается продвижение к целевому состоянию выбранных учебных задач.

Структура учебной деятельности основывается на описании МЭОП. При этом формируются механизмы описания разрешенных композиций из элементов МЭОП, а также применимые к ним действия. Обучаемый может сделать *запрос* о свойствах объекта и/или объектов из ПО, уточнить тип применимого к ним (нему) действия и т.п. Инструментальные средства создания описаний задачных и объектных представлений учебной деятельности могут включать автоматизированные словари обучения терминам, в среде которых можно представить в адекватной для восприятия форме необходимые типы учебных моделей объектов и/или понятий, дать их определения и описать все типы отношений между объектами. Формирование операциональных навыков по профилю профессиональной деятельности обучаемого обеспечивается за счет увязывания элементов опорных знаний ПО вокруг каждого промежуточного состояния процесса взаимодействия. Выбранные формы учебных моделей объектов и/или понятий требуют включения на этапе создания учебной автоматизированной среды редакторов различных типов: *текстового редактора*; *оконного графического редактора* с динамически встраиваемой системой примитивов; *экранного редактора*, описывающего вопросно-ответную структуру диалога с пользователем-обучаемым.

Применение многоуровневой модели обучающего диалога можно кратко рассмотреть на примере изучения робототехнических производств. Здесь уровень описания учебных задач определяется процессами, реализуемыми в конкретных производствах. Основу реализации моделей производственных процессов составляют объектные представления ПО. Так, для изучения автоматизированного производства, связанного с ротационной обработкой деталей, необходимо сформировать и реализовать модели автоматизированного склада, робокара, транспортной системы, палетизирующей станции, депалетизирующей станции, антропоморфного робота, робота типа "СКАМКО", поворотного стола, токарного станка, фрезерного станка. Связывание этих объектов в единый технологический процесс основывается на

реализации подсистемы управления, включающей: информационную диспетчерскую управляющую систему; систему управления складом и транспортной системой; системы автоматизированного проектирования изделия и технологического процесса; подсистемы управления отдельным робототехническим устройством. Набор учебных задач должен строиться в направлении наполнения подсистемы управления, т.к. именно подсистема управления включает различные типы моделей объектов из ПО "Автоматизированные производства". Представление самих же объектов из ПО может быть разным. Так, фирмы LEGO, FISHER разрабатывают модели-копии реальных объектов. С использованием моделей-копий реализованы и учебные кабинеты РОБКО. Здесь на основе моделей-копий изучается назначение того или иного станка, робота, устройства. Далее обучаемый решает ряд несложных учебных задач, напр., по компоновке и сборке макета реального производства или по созданию управляющих воздействий на эти модели-копии.

В большей степени понятие КУС сейчас используется при организации обучения пользователей промышленных систем. Примерами могут служить система MICROСIM, обеспечивающая обучение программированию на ассемблера языке, система АФРОДИТА, в среде которой реализуется обучение алгоритмизации задач с последующим программированием на Фортране, система САСНЕ, реализующая обучение по САПР. Средствами этих систем реализованы механизмы описания и анализа целей взаимодействия на основе представления понятийной структуры изучаемой ПО. Частным случаем КУС является компьютерная учебно-игровая среда, где используется игровая модель предметной области.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

ОПЕРАНД — объект, к которому применяется *оператор*. *О*. *А* релевантен для оператора *α*, если применение *α* к *А* может вызвать или предотвратить то или иное изменение объекта *А* либо к.-л. иного объекта. Напр., релевантными *О*. для оператора "поставить в повелительное наклонение" являются только глаголы.

В *программировании* *О*. — аргумент *операции*; грамматическая конструкция, обозначающая выражение, задающее значение аргумента операции; иногда — место, позиция в *тексте*, где должен стоять аргумент операции.

Г.А.Балл, В.А.Третьяк.

ОПЕРАТОР (от лат. operator — исполнитель) — предписание, согласно которому человек, компьютер либо иная активная система осуществляет непосредственное воздействие определенного типа. Напр., *О*. "смешать", "взбить", "закипятить" характерны для процессов приготовления пищи, *О*. "присвоить значение", "извлечь

квадратный корень”, “повторять следующие вычисления, пока...” — для вычислительных процессов. Применение *О.* к релевантному для него *операнду* представляет собой операцию. *О.* можно поставить в соответствие реализующую его подсистему активной системы (блок агрегата, орган живого организма и т.п.). Часто *О.* называют человека, выполняющего определенные операции (напр., кинооператор, оператор вычислительной машины)

В *программировании* *О.* отличаются от “машинных” команд. Термин “*О.*” используется для обозначения законченного действия, подчеркивает функцию предписания в языках высокого уровня (см. *Уровень алгоритмического языка*) как *О.* в матем. смысле, т.е. отображение текущего состояния памяти *программы* в др. состояние (*О.* присваивания) либо отображение состояния *программы* в одно из значений “ИСТИНА” или “ЛОЖЬ” (*О.* проверки условия).

Г.А.Балл, В.А.Третьяк.

ОПЕРАЦИОННАЯ СИСТЕМА — комплекс *программ*, организующих работу компьютера. В состав *О.с.*, как правило, входят: программа первоначальной загрузки; программы управления внешними устройствами компьютера; супервизор — программа распределения ресурсов компьютера и обработки ошибочных ситуаций; файловая система, поддерживающая хранение и поиск *информации* во внешней памяти ЭВМ; набор программ для тестирования и диагностики компьютера; сервисные программы. Иногда к *О.с.* относят все базовое *программное обеспечение*, включая *системы программирования* и *системы управления базами данных*. *О.с.* различаются между собой: 1) режимом работы: пакетный (все работы подготавливаются заранее в виде заданий, которые объединяются в пакеты; *О.с.* планирует порядок выполнения заданий, чтобы оптимально использовать ресурсы компьютера); реального времени (компьютер управляет работой некоторого объекта в реальном времени, а *О.с.* позволяет гибко реагировать на изменения в состоянии объекта); разделения времени (*О.с.* распределяет время центрального процессора и другие ресурсы между несколькими *пользователями*); персональный (*О.с.* обеспечивает удобство работы на компьютере для одного ее пользователя); 2) возможностью одновременной обработки нескольких программ — мультипрограммирование; 3) возможностью обслуживания нескольких центральных и/или специализированных процессоров — мультипроцессирование; 4) возможностью работы в сети.

В.А.Петрушин.

ОПЕРАЦИОННАЯ СРЕДА ДЛЯ ПРЕПОДАВАТЕЛЕЙ И СТУДЕНТОВ — комплекс средств, предназначенных для автоматизации труда *преподавателей* и студентов. Основу такой среды составляет

инструментальное программное обеспечение, включающее обычные элементы систем программирования (текстовые редакторы, графические редакторы, языки высокого уровня — см. *Уровень алгоритмического языка, базы данных* и др.) и специальные средства, обеспечивающие описание учебного материала и стратегий обучения, ориентированных на различные учебные предметные области: конструирование и технологическую поддержку учебных программ, автоматизированных учебных курсов, баз знаний, баз данных, экспертных обучающих систем; психолого-педагогические исследования; обработку статистических данных; научно-технические расчеты; моделирование и научные исследования; проектирование и другие инженерные работы; технологию программирования; информационно-справочное обслуживание. См. также Эскиз.

Ю.И.Лобанов.

ОПЕРАЦИЯ (от лат. operatio — действие) — элемент функционирования активной системы, состоящий в применении какого-либо ее оператора к релевантному для него операнду. О. описываются, напр., следующими предложениями: “Ударить кием по бильярдному шару” “Возвести число 2 в квадрат”. В этих предложениях выделенные разрядкой слова описывают операторы, оставшиеся — операнды. Различают О. индивидуальные (операнд — конкретный единичный предмет) и родовые (операнд — произвольный представитель некоторого класса предметов). В предельном случае родовая О. представляет собой применение соответствующего оператора к любому релевантному для него операнду. Таковы, напр., рассматриваемые в психологии общемыслительные О. анализа, синтеза, сравнения и т.д.

О. и их системы (в т.ч. процедуры) служат способами *воздействий*. Одно из важных отношений между воздействиями и обеспечивающими их О. можно раскрыть с помощью понятий об эффективных и квазиэффективных О. Эффективная О. обеспечивает совершенно определенное воздействие на некоторый предмет, квазиэффективная О. обеспечивает такое воздействие с вероятностью, достаточно близкой к единице. Об эффективности и квазиэффективности О. имеет смысл говорить только по отношению к определенному *решателю*.

Описывая функционирование реальных систем (напр., компьютеров), часто пренебрегают их отличием от идеализированных систем, осуществляющих только эффективные О. (напр., абстрактных цифровых автоматов). В этом смысле говорят об “абстракции безошибочности”. Однако при оценке надежности компьютеров подобная абстракция неправомерна. При переходе к рассмотрению человеческой деятельности сфера применимости абстракции безошибочности сужается. В связи с этим именно здесь весьма важно

понятие квазиэффективной О. (такие О. обычно описываются под названием элементарных).

Г.А.Балл.

ОПТИЧЕСКАЯ КОММУНИКАЦИЯ — способ взаимодействия с компьютером, при котором технические средства образуют совокупность отображающего, снимающего и регистрирующего оборудования интерактивного типа. Самое распространенное оборудование О.к. — индикаторное и отображающее, отображающее с *клавиатурой* и *световым пером*, графические регистрирующие приборы и др. Для автоматизации сбора *данных* используются счетчики первичных документов, которые делятся на счетчики простых *знаков*, записанных на точно определенных местах предварительно обозначенного формуляра; счетчики алфавитно-цифровых знаков (их без цифрового кода можно, в частности, записать на первичные документы при помощи специальных пишущих машинок со стилизованными знаками письма); счетчики рукописного *текста*, которые дают возможность считывать и распознавать написанные от руки алфавитные и цифровые знаки на предварительно обозначенных местах первичного документа (тип этих знаков может быть предписанным или произвольным); счетчики графических изображений, делающие возможным считывание графических знаков и первичных документов.

П.Горват.

ОРГАНИЗАЦИОННЫЕ ФОРМЫ КОМПЬЮТЕРНОГО ОБУЧЕНИЯ — инварианты *кооперативных взаимодействий* между преподавателем и обучаемым, а также между самими обучаемыми. Различают О. ф. к. о.: компьютерный сеанс — однократный непрерывный тематический сеанс работы на компьютере с целью решения отдельной *задачи*, усвоения заданной темы, закрепления конкретного умения; компьютерный практикум — объединенная темой и методической установкой серия компьютерных сеансов для решения задач, развития и закрепления навыков и умений; компьютерная лаборатория — объединенная темой и методической установкой серия компьютерных сеансов познавательной и исследовательской деятельности с целью изучения конкретного процесса или объекта; компьютерный проект — организованная индивидуальная или групповая деятельность студентов по созданию физического или интеллектуального продукта с интенсивным и комплексным использованием компьютеров; компьютерный курс — целостный учебный курс, органически использующий компьютер в объеме и пропорции, ограничиваемых соображениями методической и гигиенической целесообразности, и вписанный в компьютерную технологию обучения.

Компьютерный курс является синтезом всех форм учебного применения компьютеров, т.к. воплощает в себе все названные выше формы. Процесс создания компьютерного курса можно рассматривать как процесс подготовки и создания кинофильма или постановки пьесы в театре. Здесь также требуется прежде всего сценарий, который должен содержать описание способов совместного функционирования компьютера и обучаемого, компьютера и преподавателя, описание содержательной информации в терминах предметной области и образного представления этой информации. Как и при обычных формах учебного процесса, компьютерные занятия предполагают участие преподавателя в ролях ведущего консультанта или наблюдателя. Внедрение компьютеров в качестве средства обучения в системе образования создает возможности для расширения диапазона форм и вариантов учебного процесса. Это выдвигает на первый план концепцию учебно-методического комплекса, рассматриваемую как совокупность книг, документов, компьютерных программ, наглядных пособий, что в целом составляет информационный фонд обучаемых и преподавателей по учебной дисциплине. Таким образом создается возможность организовать новый вид обучения, который можно назвать обучением на основе баз знаний. См. также *Автоматизированная дипломная работа, Автоматизированная контрольная работа, Автоматизированная курсовая работа, Автоматизированная лекция, Автоматизированное лабораторное занятие, Автоматизированный зачет, Автоматизированный экзамен.*

А.Я.Савельев.

ОСВЕЩЁННОСТЬ — отношение светового потока, падающего на рассматриваемый малый участок поверхности, к площади этого участка. Единица измерения — люкс (лк). *O.* — один из наиболее важных показателей, от которого зависит уровень зрительного комфорта на рабочем месте, оснащённом видеотерминалом. Необходимо отличать *O.* от естественного и искусственных источников света.

При выборе оптимального уровня *O.* на конкретном рабочем месте нужно учитывать вид деятельности пользователя, наличие или отсутствие антибликового покрытия на поверхности экрана, макс. уровни яркости и контраста изображения на экране. *O.* на рабочем месте должна быть такой, чтобы при работе с изображением на экране, клавиатурой, печатным текстом, рукописью пользователь не испытывал зрительного дискомфорта. При размещении рабочих мест в помещении необходимо учитывать уровни естественной *O.* в различных точках данного помещения. Средний уровень естественной *O.* используемого помещения зависит от его пространственной ориентации (по сторонам света) и площади оконных проёмов. Для ограничения уровня естественной *O.* необходимо использовать шторы,

жалюзи и при необходимости светозащитные экраны на конкретном рабочем месте. Поскольку естественная *O.* в течение суток изменяется, рекомендуется применять источники искусственного освещения, которые должны располагаться так, чтобы избежать дискомфорта из-за *блескости* или отражений. Для устранения непрямой *блескости* и отражений от экранов светильники общего освещения должны быть локализованы относительно рабочих мест так, чтобы *дисплеи* располагались параллельно источникам освещения и окнам (и, если это возможно, между рядами светильников). Угол зрения к источнику освещения для пользователя должен составлять менее 50°.

Соотношение яркостей в непосредственной рабочей зоне изображения на экране, документации, клавиатуре и окружающих предметов (стен и др.) во избежание постоянной яркостной переадаптации зрительного анализатора пользователя должна быть не более 1:2. Светильники на *автоматизированном рабочем месте* у экрана дисплея, как правило, не рациональны. Они увеличивают и без того имеющиеся различия в яркости между экраном, документом и клавиатурой. *O.* на рабочем месте, оснащённом видеотерминалом, должна быть регулируемой. Оптимальный уровень *O.* должен находиться в пределах от 300 до 500 лк.

В.М.Бондаровская, Т.Дамянова, С.А.Миронченко.

ОТВЕТ — тип реплики в *диалоге*: суждение, выступающее как реакция на *вопрос*. В теории *задач* — знаковая модель результата решения *познавательной задачи*, соответствующего определенному вопросу. *O.* является правильным, если достигнута достаточная полнота (т.е. и объем, и адекватность) *информации* об искомом предмете (предметах), которого (которых) касается вопрос; не-правильным, если не достигнута ее достаточная адекватность; частичным, если при достаточной адекватности не достигнут требуемый объем информации. По отношению к псевдовопросу нельзя говорить об *O.* (правильном, неправильном или частичном). Правомерными реакциями на псевдовопросы могут служить лишь так наз. несобственные ответы — суждения, указывающие на не-правильность поставленных вопросов.

O. либо выбирается из ограниченного набора альтернатив (при закрытом вопросе), либо свободно конструируется с помощью различных знаковых средств (при открытом вопросе). Если распознавание выборочных *O.* не представляет затруднений, компьютерный анализ *O.*, конструируемых обучаемыми, — одна из актуальных проблем *компьютерной технологии обучения*. Эта проблема чаще всего решается путем выявления в *O.* *ключевых слов* или морфем (предполагается, что их употребление с достаточно большой вероятностью свидетельствует о правильности *O.*). При проектировании

учебного диалога важным требованием является установление (с учетом характеристик обучаемого и решаемой им задачи) оптимального времени ожидания его О. и выбор педагогически обоснованной реакции на нарушение временного режима диалога.

Г.А.Балл, Е.Д.Маргулис.

ОТКРЫТЫЙ УНИВЕРСИТЕТ — одна из форм высшего образования. Первый О.у. был организован в 1963 по инициативе Г. Вильсона, премьер-министра Великобритании; регулярное обучение началось в 1969. Часть О.у. функционируют как учебные заведения нового типа, другие являются традиционными учебными заведениями с *“открытой технологией” обучения*. Суть этой технологии состоит в *индивидуализации обучения* (во времени и пространстве) с использованием *технических средств обучения* (радио, телевидение, аудиовизуальные средства, видеосистемы) вместе с *компьютерной технологией обучения*. “Открытая технология” применяется прежде всего при изучении *информатики и вычислительной техники*. Лучший пример “открытой технологии” демонстрирует О.у. Великобритании. Ныне в мире насчитывается более 300 О.у. Наиболее крупные О.у. находятся в Китае, Индии и Таиланде. Организацию О.у. поддерживает ЮНЕСКО.

Д.Ковач.

ОТЛАДКА ПРОГРАММЫ — процесс отыскания и исправления ошибок в *программе*, когда сам факт их существования установлен (при *тестировании программы*); наименее изученный и наименее формализованный этап разработки *программного обеспечения*. Отладка сложных программ, кроме опыта и умения ориентироваться в отлаживаемой программе, требует хорошего знания используемых *инструментальных средств* (*компиляторов, редакторов связей, интерпретаторов* и т.д.), а также архитектуры конкретной ЭВМ и особенностей конкретной *операционной системы*. Все ошибки можно условно разбить на 5 типов в зависимости от этапа разработки программы, на котором они вносятся: ошибки в *описании задачи*; ошибки в *проектировании*; ошибки в *кодировании*; ошибки, связанные с неверным использованием конструкций *языка программирования*; физические ошибки (*описки, пропуск строк* и т.д.). Ошибки первого типа являются, как правило, следствием *неправильного понимания задачи и требований пользователя*, второго типа — следствием *неверной разбивки на модули и описания интерфейса* между ними, неверного выбора *алгоритмов* и структур *данных*. Эти ошибки специфичны для конкретной области применения *вычислительных средств*. Ошибки третьего типа, напротив, мало зависят от решаемой задачи (типичный пример такой ошибки — выполнение цикла на один раз больше или меньше, чем

нужно). Ошибки четвертого типа вызываются либо плохим знанием языка, либо неверным пониманием его конструкций. В любом случае следует стараться избегать ошибок еще на ранних стадиях работы над программой. Большую помощь в этом может оказать соблюдение принципов структурного и модульного *программирования*, повышение читабельности программ, а также единообразие кодирования сходных алгоритмов.

Большинство современных *систем программирования* включает средства диалоговой О.п., как правило, в терминах исходного *текста* (т.н. отладчики). Такие средства включают: пошаговое выполнение программ, установку точек прерывания, возможности просмотра и изменения данных, просмотра последовательности вызовов *подпрограмм* и передаваемых при этом параметров, возможность останова по условию или изменению значения некоторой переменной и т.д. Таковы, напр., отладчики RTD (язык *Модула-2*, фирма Logitech) и Codeview (язык *СИ*, фирма Microsoft). Использование таких отладчиков может существенно ускорить процесс О.п. и облегчить его.

И.Б.Романенко, Е.М.Синица.

ОТРАСЛЕВАЯ АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ (ОАСУ) — человеко-машинная система, обеспечивающая функционирование объектов управляемой системы, в которой сбор и переработка *информации*, необходимой для реализации функций управления, осуществляется с применением средств автоматизации и вычислительной техники. Методы и средства ОАСУ Государственного комитета по народному образованию (ГКО) образуют в системе народного образования (СНО) развивающуюся гибкую и адаптивную интегрированную организационно-технологическую и информационно-вычислительную среду, активно способствующую формированию в СНО наиболее благоприятных условий для реализации *стратегии* непрерывного комплексного совершенствования отраслевого управления. Главная цель создания ОАСУ ГКО — совершенствование отраслевого управления, направленное на повышение эффективности функционирования и сбалансированности развития СНО и ее объектов. Эта цель достигается поэтапным органическим встраиванием в существующую систему управления отрасли новых, преимущественно безбумажных, информационных технологий, базирующихся на широком использовании программно-технических средств вычислительной техники и информационно-коммуникативных систем, а также передовых методов управления и экономико-математических *моделей*, предназначенных для реализации функций управления СНО как автономно, так и в составе общегосударственной автоматизированной системы (ОГАС).

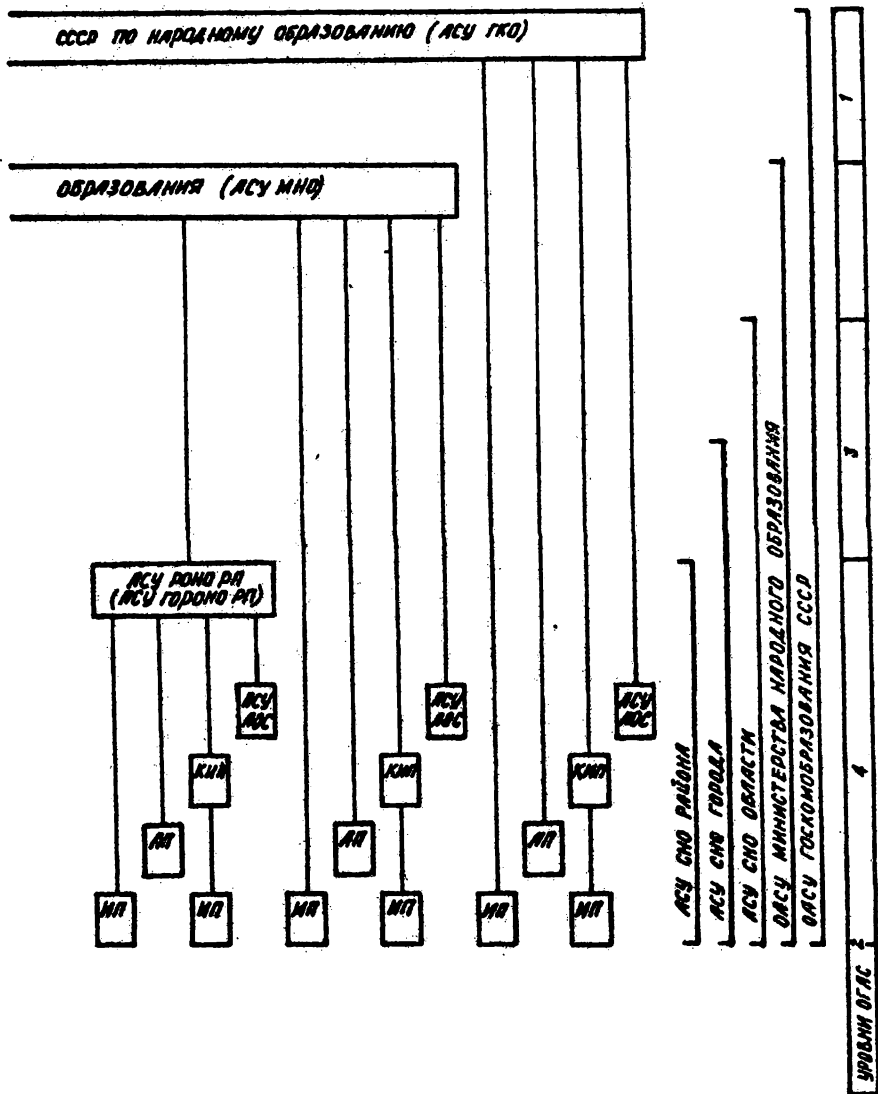
Организационно ОАСУ ГКО представляет собой многоуровневую иерархическую систему управления, обеспечивающую согласованное

(по уровням управления, целям, критериям, методам управления и процедурам обработки данных) совместное функционирование входящих в нее элементов. В основе организационного строения ОАСУ ГКО (рис.) лежит организационная структура СНО, отражающая отраслевой и территориальные аспекты управления и включающая шесть организационных уровней, из которых пять верхних взаимодействуют с соответствующими территориальными АСУ, входящими в один из трех верхних уровней ОГАС. При этом состав и функции элементов ОАСУ на различных ее организационных уровнях также различны.

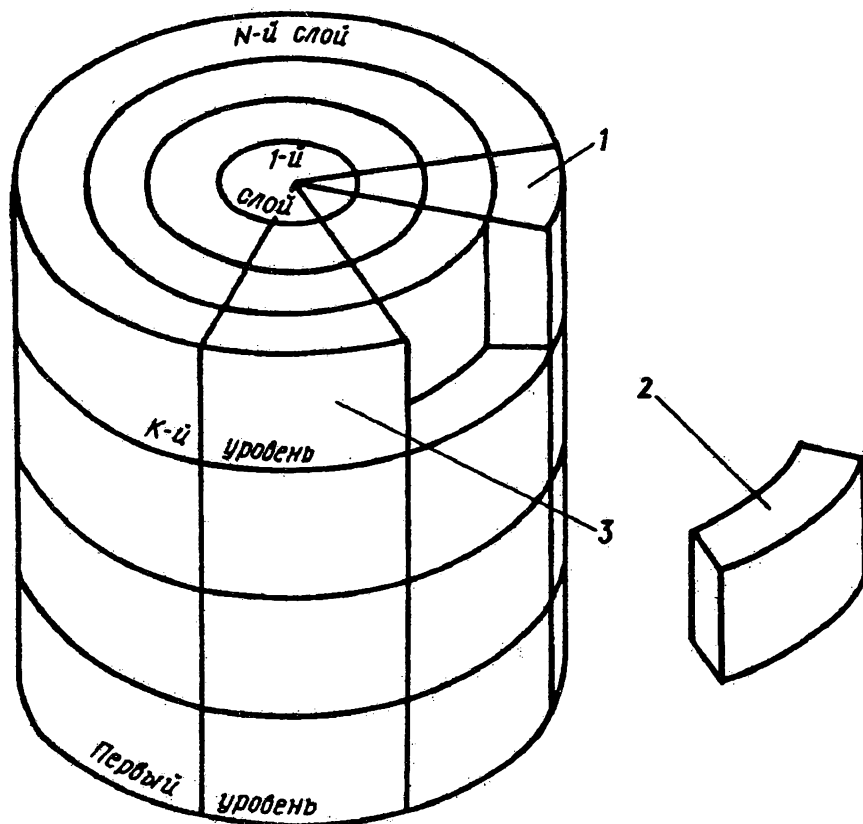
Системотехническую основу ОАСУ ГКО образует динамическая комплексная компьютерная модель отрасли, включающая структурно упорядоченную совокупность моделей объектов управления (МОУ), моделей окружающей среды (МОС) и моделей процессов управления (МПУ) — система моделей (СМ) ОАСУ. При этом СМ ОАСУ должна “жить” вместе с СНО, непрерывно развиваясь, видоизменяясь и уточняясь по мере развития, изменения и изучения СНО. МОС и МОУ представляют собой структурно упорядоченную совокупность существенных параметров объектов окружающей среды и объектов управления (учреждений народного образования — УНО), отражающих их цели и состояния (ретроспективное, текущее, желаемое и плановое). МПУ характеризуют содержательный и технологический аспекты отраслевого управления и представляют собой формализованное описание целей и функций СНО (как внешних, так и внутренних) по удовлетворению потребностей окружающей среды в конечном продукте СНО. Совокупность МПУ, входящих в СМ ОАСУ, отражает различные фазы процесса *принятия решений* в оргсистемах: прогнозирования и планирования развития СНО, оперативного управления функционированием отрасли, комплексного анализа ее деятельности, учета и отчетности о состояниях и характере деятельности СНО.

Если в качестве элементов СМ ОАСУ выбрать отдельные входящие в нее модели, а в качестве существенных сторон модельного представления строения СМ ОАСУ — аспекты, отражающие иерархию отношений в СМ ОАСУ, эволюционно-поэтапный характер ее развития, а также принадлежность различных множеств моделей СМ ОАСУ к тем или иным ее подсистемам, то строение СМ ОАСУ можно представить в виде так наз. цилиндра моделей (рис.), в котором все пространство внутри его объема разделено на K уровней, N слоев и L секторов, отображая соответственно первый, второй и третий из указанных аспектов модельного представления строения СМ ОАСУ.

Параметры и строения отдельных моделей, входящих в СМ ОАСУ, а также структура самой СМ ОАСУ загружаются в автоматизирован-



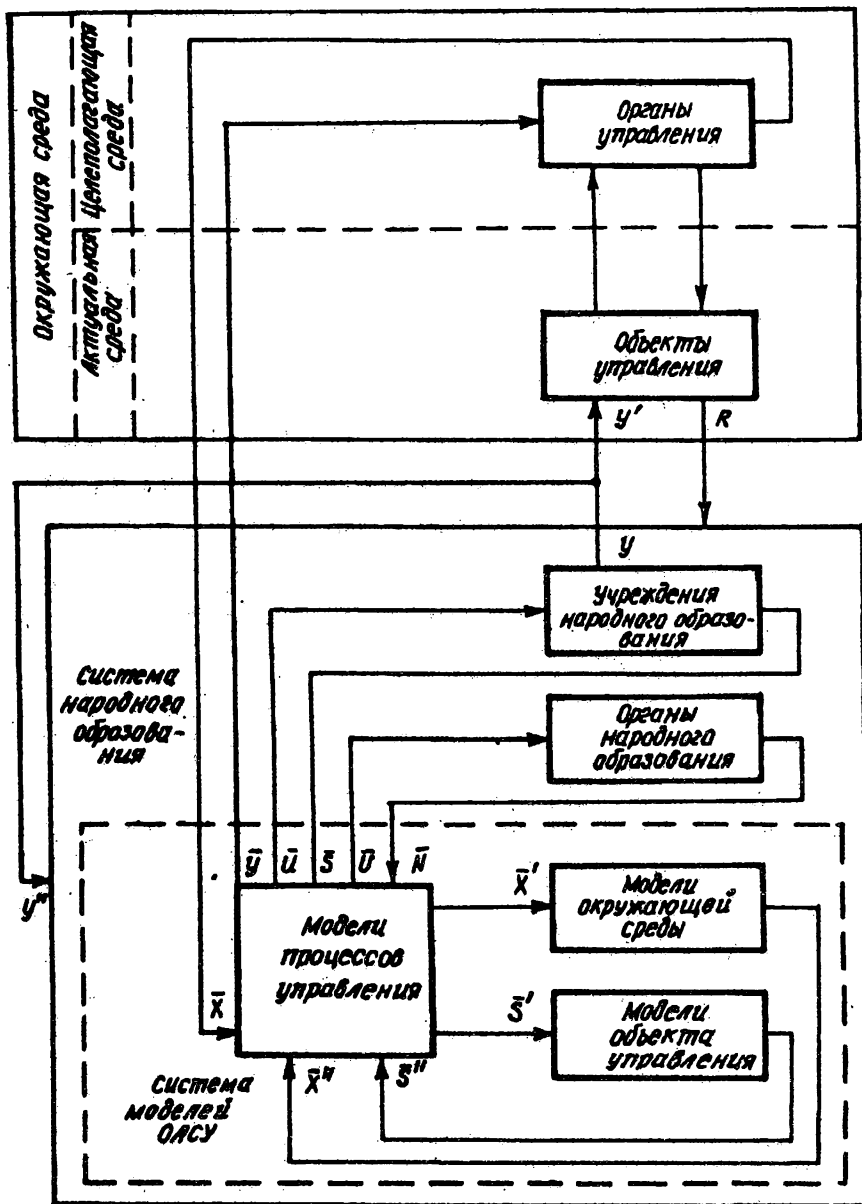
моделей (нормативов, классификаторов и др.). В свою очередь, библиотеки программ реализуют различные МПУ, обеспечивая решение функциональных задач ОАСУ и их агрегирование в функциональные подсистемы, комплексы задач и функциональные



блоки управления. Обобщенная схема взаимодействия СМ ОАСУ с элементами СНО и окружающей среды приведена на рис. с. 381.

В.Е.Быков.

ОЦЕНКА КАЧЕСТВА ОБУЧЕНИЯ СТАТИСТИЧЕСКАЯ — оценка степени освоения *знаний* путем статистической проверки. При О.к.о.с. сначала составляется программа из N контрольных *вопросов*. Затем для каждого *обучаемого* выбирается n вопросов. Определяется число m правильных *ответов*. В зависимости от доли правильных ответов $b = m/k$ принимается решение о выборе i -й, $i=1, \overline{k}$ оценки. Здесь величина k определяет макс. число баллов по принятой шкале оценок. При $k=2$ шкале оценок соответствуют двухбалльные оценки: “зачет” “незачет”. Следовательно, при чисто



статистическом подходе оценка качества обучения сводится к определению вероятности получения правильных ответов на случай-

ную выборку n вопросов из N . Если принять гипотезу, что обучаемый сможет ответить на M вопросов из N , то эта вероятность равна:

$$P(n, m) = \frac{C_M^m \cdot C_{N-M}^{n-m}}{C_N^n},$$

где C_M^m — число сочетаний из M по m . Качество навыков оценивается по скорости и точности выполнения требуемых операций. При достаточно большом времени наблюдений скорость выполнения операций C оценивается выражением

$$C = n / \sum_{i=1}^n \tau_i,$$

где τ_i — время выполнения операций в i -й попытке, n — количество попыток. Удобнее оценивать уровень владения операцией по среднему времени ее выполнения:

$$\tau_{\text{ср}} = \sum_i \tau_i / n.$$

Во многих практических случаях степень освоения умений по решению некоторого типа задач может оцениваться по совокупности значений среднего времени операций и доли правильно решенных задач.

Ю.И.Лобанов.

ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ (ППП) — комплект программ, ориентированный на решение совокупности однотипных задач в некоторой проблемной области. Способствует проникновению средств вычислительной техники в различные сферы человеческой деятельности. Напр., в промышленности широко применяются системы управления, в образовании — *автоматизированные обучающие системы*, в инженерном проектировании и научных исследованиях — системы автоматизации проектирования и научных исследований, в делопроизводстве — ППП автоматизации административной работы. Разработаны ППП для применения в быту (напр., для ведения дома, для игры и т.д.). На рынке программного обеспечения наиболее распространены ППП, ориентированные на автоматизацию административных работ (управленческой деятельности). Сфера делопроизводства и административного управления характеризуется большим количеством ППП, ориентированных на ведение персональных баз данных (напр., ППП dBase II), обработку текстов (Easy Writer II, PC-Text, Wordstar 2000), расчеты на основе электронных таблиц (VisiCalc), графику для коммерческих приложений, обмен данных по локальным вычислительным сетям.

ППП широко применяют в сфере образования. Возникающие здесь проблемы имеют, с одной стороны, много общего с проблемами других сфер человеческой деятельности (напр., задачи управления

учебными заведениями), а с другой — обладают определенной спецификой, обусловленной наличием процесса обучения (или управления познавательной деятельностью), вследствие чего развились два специфических для сферы образования класса ППП: *автоматизированные учебные курсы* и системы, ориентированные на автоматизацию их разработки (*авторские языки программирования, авторские системы* и т.п.).

Применение ППП предоставляет *пользователю* определенный уровень сервиса при решении однотипных задач. Однако в реальной ситуации пользователю необходимо иметь дело одновременно с набором различных задач. Напр., при корректировке данных в личной базе часто приходится составлять документ с учетом вносимых изменений и передавать этот документ по линиям связи. При этом возникает потребность в различных ППП. Для повышения уровня сервиса в таких случаях служат *интегрированные пакеты прикладных программ*, обладающие высокой гибкостью. Одним из самых популярных интегрированных ППП является Lotus 1-2-3. Ныне имеется значительное количество других интегрированных ППП. Многие из них используют метафору “рабочего стола”, которая реализуется на основе графического *многооконного интерфейса*.

В.И.Отенко.

ПАМЯТЬ ЭВМ — функциональная часть ЭВМ, предназначенная для запоминания, хранения (длительного или кратковременного) и выдачи входной и вспомогательной *информации, программ*, промежуточных или окончательных результатов. По характеру связи с *процессором* различают внутреннюю и внешнюю память. Внутренняя память составляет неотъемлемую физическую часть машины, непосредственно связанную с процессором, и все *данные*, хранящиеся в такой памяти, автоматически доступны этой машине. Внешняя память хранит информацию в форме, принятой для данной машины, но в отличие от внутренней может быть отделена от машины. В зависимости от организации вычислительного процесса и емкости П. ЭВМ подразделяют на основную и вспомогательную. В основной памяти хранится выполняемая программа и осн. часть относящихся к ней данных. Она всегда является внутренней. Рабочая память — часть осн. памяти, предназначенная для хранения промежуточных результатов. Оперативная память — часть осн. памяти, характеризующаяся высоким быстродействием (10^6 — 10^8 байт/с) и относительно небольшим объемом (10^5 — 10^7 байт); имеет постоянную связь с процессором; вся информация адресуется в байтах, словах или небольших группах слов. Оперативная память может иметь еще большее быстродействие и меньший объем (сверхоперативная память). Вспомогательная память (объем 10^7 — 10^8 байт) используется в качестве дополнения или расширения осн. памяти. Единицы

обращения — массивы (зона, страница) слов (10^2 — 10^3 байт). Вспомогательная память может быть внешней и внутренней. По способу обращения П. ЭВМ подразделяют на адресную, ассоциативную и магазинную. Адресная память — память с произвольным доступом (по адресу) к любой области памяти независимо от ее содержания. Ассоциативная память, кроме того, позволяет обращаться к ней по содержанию, т.е. по признаку, содержащемуся в слове, которое нужно прочитать. Магазинная память состоит из упорядоченных в одну сторону областей памяти (байтов или слов). Бывает двух типов: “первый вошел — первый вышел” (порядок записи и чтения один и тот же), и “последний вошел — первый вышел” (порядок чтения обратный порядку записи). По способу хранения информации память бывает постоянной и полупостоянной. Постоянная память допускает только чтение информации и используется для хранения стандартных программ, *операционной системы*, некоторых констант и т.д. Запись информации выполняется на предприятии-изготовителе, после чего любые изменения содержимого памяти становятся невозможными. Полупостоянная память, кроме чтения, позволяет также записывать информацию.

П. ЭВМ образуется из одного или нескольких *запоминающих устройств* (ЗУ), запоминающие элементы которых объединяются в ячейки. Ячейка служит для хранения машинного слова — стандартной порции информации, которую ЗУ может выдать или принять за одну операцию обмена информацией с другими устройствами. Длина машинного слова (т.е. число его двоичных разрядов) фиксирована в каждой модели ЭВМ. Это может быть 8, 16, 24, 32, 48, 60 и более бит. Ячейкам приписывают неповторяющиеся номера, которые принято называть адресами. По адресу в память помещается (записывается) или извлекается (считывается) машинное слово. При записи слова в ячейку прежнее содержимое этой ячейки уничтожается. При чтении слов из памяти извлекается копия хранящегося там слова; само содержимое ячейки остается неизменным.

Наибольшее количество единиц информации, которое может одновременно храниться в устройстве памяти, определяет емкость. Наиболее часто емкость оценивают количеством ячеек, каждая из которых способна хранить ровно одно машинное слово. Однако в машинах серии ЕС ЭВМ одна ячейка памяти содержит только один байт (восемь двоичных разрядов), и для хранения 32-разрядного машинного слова используются четыре подряд расположенные ячейки. Поэтому емкость памяти часто выражают в байтах, килобайтах, мегабайтах.

В.П.Козлова.

ПАСКАЛЬ — универсальный язык программирования, используемый для написания прикладных программ и как язык системного программирования. Создан швейц. ученым Н.Виртом в конце 60-х гг. как специальный язык для обучения. Используется практически на всех типах компьютеров. Обладает следующими функциональными возможностями и выразительными средствами: концептуальной целостностью, что позволяет создавать алгоритмически сложные программы; развитыми средствами представления структур данных, обеспечивающими удобную работу с числовой, символьной и битовой информацией; реализацией идеи структурного программирования, что делает программу наглядной и значительно упрощает ее отладку и сопровождение; специальными методиками создания трансляторов (метод “раскрутки”), упрощающими их разработку, что обусловило высокую степень стандартизации свойств различных реализаций П.; эффективностью оптимизирующих свойств трансляторов.

На микропроцессорной технике П. получил широкое распространение в составе авторских систем для программирования на П. Эти системы состоят из транслятора, сервисных и прикладных программ. Транслятор реализует стандартное подмножество языка и представляет ряд дополнительных возможностей, в частности работу с данными типа BYTE (последовательность бит) и STRING (строка символов). Графические и иные возможности реализованы в виде встроенных системных единиц (UNIT), являющихся функциональным эквивалентом пакетов на языке Ада.

В.А.Петрушин, В.А.Третьяк.

ПЕДАГОГИЧЕСКИЙ ПРОГРАММНЫЙ ПРОДУКТ — программные средства, используемые в педагогических целях их автором и/или педагогами-практиками. Разработана (в Научно-исследовательском институте содержания и методов обучения Академии педагогических наук под руководством академика В.М.Монахова) методическая концепция создания П.п.п. Согласно этой концепции предметным овеществлением информационных технологий в сфере образования становятся: информационный банк, пакеты программ учебного назначения (ПУН) и базы данных, информационно-поисковые системы и информационные сети. Осн. положения концепции: специфика содержания учебного предмета и ее учет в формулировке требований к П.п.п.; особенности взаимосвязи новых учебных программ как нормативных документов, программирующих осн. компоненты учебно-воспитательного процесса в массовой школе; методика отработки с помощью ПУН необходимого объема знаний и умений на уровне, зафиксированном в новых учебных программах; разработка конструктивных и продуктивных рекомендаций по созданию ПУН и их оценке, позволяющей выявить соразмерность их методической эффективности с экономическими затратами. Осн.

принципы разработки П.п.п.: многоступенчатость, реализация плана или сценария, активное участие *обучаемого* в его создании (уточнение сценария). Другой класс программ — инструментальные программные средства: они не реализуют конкретный сценарий, но позволяют создать “дружественный” *интерфейс* для общения с компьютером. Такие средства разделяются на два типа: 1) используемые в учебном процессе в качестве инструмента для исследования и средства решения *задач*; 2) используемые для создания *обучающих программ*. Первый тип ориентирован на обучаемого, расширяя набор средств его деятельности, второй тип — на разработчика учебных компьютерных программ (пример — система “РАДУГА”, созданная в Московском государственном педагогическом университете). При разработке и внедрении информационной технологии *обучения* различают пять этапов. На первом этапе отдельные П.п.п. на некоторых уроках выступают как средство обучения. На втором этапе *преподаватели* обращаются к П.п.п. вследствие их положительного влияния на методику обучения. На третьем этапе образуется методика, включающая усовершенствованную традиционную методику и фрагменты информационной технологии в виде пакетов ПУН. На четвертом этапе экспериментально проверяется и систематизируется методика или ее фрагменты. На пятом этапе корректируются цели и содержание обучения по большинству школьных предметов, динамика и дидактические принципы учебно-воспитательного процесса. Эта корректировка происходит преимущественно в аспекте их расширения за счет более продуктивной методической системы обучения. Это завершающий этап разработки информационной технологии, элементы которой начинают отрабатываться со второго этапа. Осн. дидактико-методические задачи, обосновывающие разработки и внедрения информационной технологии обучения: развитие методики продуктивного использования П.п.п. на уроках; создание методики разработки П.п.п., органически входящих в учебно-воспитательный процесс; поиск и отработка направлений использования возможностей информационной технологии (эта задача условно наз. информатизацией методической науки).

Е. Д. Маргулис.

ПЕДАГОГИЧЕСКИЙ ПРОЦЕССОР — см. *Вуйсор*.

ПЕДАГОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ — совокупность *обучающих программ*, используемых в *компьютерной технологии обучения* прежде всего для подготовки сценария *обучения*. Учебный материал на этом уровне представлен в виде последовательности определенных доз, порций и

описания процесса управления усвоением материала. Доза учебного материала завершается обычно одним или несколькими заданиями, которые *обучаемые* выполняют и затем вводят полученные результаты в компьютер. В зависимости от степени правильности *ответов* обучаемый после выполнения задания и ввода ответов либо переходит к изучению следующих доз, либо возвращается к предыдущим.

Н.М. Розенберг.

ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА, внешние устройства — устройства, выполняющие внешние функции машинной обработки информации: подготовку или преобразование ее к виду или форме, удобной для ввода в ЭВМ или вывода из ЭВМ, хранение информации и др. К П.у. относят *устройства ввода* — *вывода информации* и устройства подготовки данных (УПД), применяемые для записи данных на *носители информации* с целью последующего ввода в ЭВМ. УПД, как правило, применяются на средних, больших и супер-ЭВМ. С развитием *терминальных сетей* и внедрением ПЭВМ области применения УПД сокращаются.

В.И. Сигалов.

ПЕРСОНАЛЬНАЯ ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ПЭВМ), персональный компьютер — *микро-ЭВМ* для индивидуального (персонального) пользования. Отличается большим запасом готовых программных средств прикладного характера, малогабаритными *накопителями информации* значительной емкости на сменных *носителях информации*, малыми габаритами, массой, энергопотреблением и высокой надежностью. ПЭВМ могут использовать лица, не обладающие специальной подготовкой в области вычислительной техники и программирования, на местах; не требующих специального оборудования. Различают ПЭВМ бытовые (ПМ1), учебные (ПМ2) и профессиональные (ПМ3—ПМ5) (табл. 1). ПЭВМ включает базовый комплект, *периферийные устройства* и *программное обеспечение* ЭВМ. В базовый комплект входят основной и дополнительный (при необходимости) *микропроцессоры*, *оперативное* и *постоянное запоминающие устройства*, средства подключения периферийных устройств, *клавиатура* и *дисплей*. Набор периферийных устройств чаще всего составляют *печатающее устройство*, *накопители на гибком и жестком магнитных дисках*. Программное обеспечение должно включать *операционные системы*, *системы программирования*, а также *пакеты прикладных программ*, выполняющие обработку текстовой, табличной и графической информации, управление *базами данных*, трансляцию с языков *программирования* Бейсик, Фортран, Паскаль, Си, работу в *локальной сети*, контроль работоспособности ПЭВМ. Основные характеристики

Таблица 1

Основные параметры ПЭВМ

Параметр	ПЭВМ				
	ПМ1	ПМ2	ПМ3	ПМ4	ПМ5
Разрядность основного микропроцессора, бит	8;16	8;16	8;16	16;32	32
Быстродействие, млн операций/с, не менее	0,5	0,5	1,0	1,0	2,0
Емкость оперативного запоминающего устройства, М, не менее	0,064	0,064	0,512	1,024	2,048
Емкость накопителя на гибком магнитном диске, М, не менее	0,36	0,36	1,0	1,0	1,0
Емкость накопителя на жестком магнитном диске, М, не менее	—	—	10,0	20,0	40,0
Потребляемая мощность, Вт, не более	30	40	100	150	200
Масса, кг, не более	5	7	10	15	20

ПЭВМ единой системы (СНГ) и фирмы IBM (США) приведены в табл. 2.

Таблица 2

Основные характеристики ПЭВМ

Модель	Основной процессор	Сопроцессор	Емкость памяти, М		
			оперативной	накопителя на гибком магнитном диске	накопителя на жестком магнитном диске
ПЭВМ единой системы (СНГ)					
ЕС 1840	K1813BM86	—	0,256—0,64	2x0,32	—
ЕС 1841	K1810BM88	K1810BM87	0,5—2,0	2x0,36	10,0
ЕС 1842	K1810BM88	K1810BM87	1,0—3,0	2x0,72	20,0
ЕС .85	i80386	i80387	4,0—8,0	2x3,0	40,0
ПЭВМ фирмы IBM (США)					
IBM PC	i8086	i8087	0,256	2x0,36	—
IBM PC XT	i8088	i8087	0,512	0,36	10,0
IBM PC AT	i80286	i80287	0,512—4,0	1,2	20,0

Модель	Основной процессор	Сопроцессор	Ёмкость памяти, М		
			оперативной	накопитель на гибком магнитном диске	накопитель на жестком магнитном диске
IBM PS/2					
M30	i8086	i8087	0,64—2,0	0,72	20,0
M50	i80286	i80287	1,0—7,0	1,44	20,0
M60	i80286	i80287	1,0—15,0	1,44	44,0—70,0
M80	i80386	i80387	1,0—22,0	1,44	44,0—115,0

В.И.Сигалов.

ПЕЧАТАЮЩЕЕ УСТРОЙСТВО, принтер — устройство, автоматически печатающее на бумаге (или ее заменителе) информацию, выводимую из электронной вычислительной машины. Дает возможность получать распечатки текстовой информации, графическую интерпретацию данных, диаграмм и графиков и т.д. Устройство отличается многочисленными вариантами шрифтов и способами оформления текста. Можно, напр., выделять слова, печатая их в разрядку или подчеркивая, определять размеры полей, расстояние между линиями, изменять толщину букв, создавать и укреплять на бумаге образы, проекты устройств, печатных плат, получать разноцветные распечатки.

По принципу вывода символов различают П.у. с посимвольным выводом (таким, как у пишущей машинки — строка печатается знак за знаком), с построчной и постраничной печатью, по способу печати — П.у. ударные (мозаичные, построчные, розетковые и др.) и безударные (термические, потоковые, лазерные, постраничные электростатические, магнитографические и т.д.). В ударных П.у. изображение символа на бумаге получается посредством удара рычага с литерой или головки по красящей ленте и переноса отпечатка на бумагу. В безударных П.у., напр., термопринтерах, изображение получают на специальной термочувствительной бумаге, которая изменяет свой цвет под действием прижимаемых к ней нагревательных элементов. Символ изображается точками; число точек соответствует числу нагревательных элементов. Термопринтеры имеют небольшие размеры, могут располагаться на столах вместе с дисплеем и центральным процессором. В ударных построчных П.у. скорость печати достигает 1500 строк в минуту, в построчных безударных П.у. — термопринтерах — 1000 строк в минуту. К быстродействующим безударным (8—30 страниц в минуту) с очень высоким качеством печати относятся лазерные П.у. Применение их целесообразно в оперативной полиграфии.

В.П.Козлова, Э.Ш.Примакова, А.Рафб.

ПЛ/1 (от англ. Programming Language/One — язык программирования/первый) — универсальный язык программирования. Создан в 1956 в США. В обучении применяется как инструментальный язык для создания *автоматизированных учебных курсов и автоматизированных обучающих систем*. Сочетает в себе блочную структуру и динамическое распределение памяти Алгола (см. *Алгол-60, Алгол-68*), структуру записей и средства ввода—вывода Кобола и арифм. операции Фортрана, а также располагает дополнительными средствами: обработкой строк, списков и др. Важным свойством ПЛ/1 является возможность инициирования асинхронного выполнения *задач* и управления процессом решения, основываясь на таких факторах, как временные задержки или завершение другой задачи. В ПЛ/1 также реализована идея родовых функций, что означает использование одного имени функции для различных типов входных *данных*. К важной особенности ПЛ/1, отличающей его от других языков программирования, относится широко используемая концепция условий по умолчанию, позволяющая *пользователям* указывать не все атрибуты данных, функций и операторов. Другой его особенностью является то, что пользователь может работать с ограниченным множеством средств языка, которые необходимы для решения его задачи.

Ныне ПЛ/1 на микропроцессорной технике не применяется. Однако он послужил основой для создания языка программирования XPL, который, в свою очередь, является родоначальником целого семейства языков (ПЛ/3, ПЛ/65, ПЛ/М-80, ПЛ/М-86, ПЛ/М-286), призванных заменить языки ассемблерного типа.

В.А.Третьяк.

ПЛАН ДИАГНОСТИКИ — последовательность действий программы учебного назначения по определению текущего состояния знаний *обучаемого*. Исходные *данные* процесса планирования — описание предметных знаний и набор типов диагностических заданий. Формирование такого плана заключается в пошаговом построении *диагностического теста*. На каждом шаге генерируется последовательность действий, которую можно реализовать одним из имеющихся диагностических модулей системы. Каждый шаг генерации определяется согласно дереву предполагаемых состояний. Выбор шагов начинается с корня дерева и продолжается до постановки диагноза. Он осуществляется так, что диагноз ставится за миним. время. См. также *Планирование учебной деятельности*.

В.В.Съедин.

ПЛАН КОНТРОЛЯ — последовательность действий программы учебного назначения по определению качества знаний *обучаемого*. В качестве исходных *данных* процесса планирования выступают

текущее состояние знаний обучаемого, подлежащее контролю, и последовательность типов заданий. Процесс планирования сводится к поиску оптимальной последовательности таких действий обучаемого, что достигается оптимальное значение показателя качества обучающей системы — времени опроса обучаемого и выставления оценки. См. также *Планирование учебной деятельности*.

В.В.Съедин.

ПЛАН ОБУЧЕНИЯ — последовательность действий программы учебного назначения по переводу обучаемого из некоторого начального состояния в желаемое конечное состояние в процессе обучения (см. *Планирование учебной деятельности*). В процессе планирования использует схему условий достижения цели, т.е. логические правила выбора очередности заданий при формировании плана учебных действий. Различают несколько типов заданий: теория, иллюстрация, пример, *вопрос, задача*. Введение в описание предметной области типа действия и схемы условий достижения цели позволяет реализовать различные методы обучения. Процесс планирования заключается в выборе оптимальной последовательности действий обучаемого, таких, что достигается оптимальное значение показателя качества обучающей системы — времени обучения.

В.В.Съедин.

ПЛАН ТРЕНИРОВКИ — последовательность действий программы учебного назначения по совершенствованию в процессе обучения некоторых параметров текущего состояния модели обучаемого. Процесс планирования осуществляется на основании модели обучаемого, схемы условий достижения цели, перечня параметров, которые необходимо улучшить, и заключается в выборе оптимальной последовательности таких действий обучаемого, что достигается желаемое значение параметров состояния обучаемого. См. также *Планирование учебной деятельности*.

В.В.Съедин.

ПЛАНИРОВАНИЕ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ — формирование программой учебного назначения плана учебных действий обучаемых на основании информационных модели предметной области, модели обучаемого, модели ресурсов обучающей системы, методов и целей обучения. Модель ресурсов обучающей системы включает перечень имеющихся программных обучающих модулей при организации учебной деятельности на основании плана учебных действий (различают четыре вида их — *план обучения, план тренировки, план контроля, план диагностики*) — последовательности действий обучаемого под управлением обучающей системы, соответствующей переводу обучаемого из некоторого начального

состояния S_0 в желаемое конечное состояние S_k в процессе обучения, трактуемого как многошаговый процесс, на каждом шаге которого осуществляется перевод обучаемого (посредством действия d_i за время $t(d_i)$) из состояния S_i в состояние S_j , где S_i, S_j — элементы некоторого множества допустимых состояний. Под состоянием S_i понимается конкретный набор *знаний, умений и навыков*, приобретённых обучаемым априори, S_j — набор *знаний, умений и навыков*, которые он может получить при взаимодействии с обучающей системой (реализующей управляющее действие d_i) на основании предъявленного учебного материала и априорных знаний. Если S — полный набор *знаний, умений и навыков*, которые может приобрести обучаемый при взаимодействии с обучающей системой, то любое подмножество a из S_k , покрываемое S , может быть целью планирования. Оптимальная последовательность состояний обучаемого $S_0, S_1, S_2, \dots, S_k$, достигаемая при реализации обучающей системой плана учебных действий, трактуется как цель обучения. Назначение плана учебных действий — генерация индивидуального обучающего курса (осуществляемая при отображении плана на множество обучающих моделей системы) и предъявление *информации* о содержании курса и видах учебной деятельности по требованию *пользователя*.

В.В.Съедин.

PLATO (по имени древнегреческого философа Платона) — одна из первых *автоматизированных обучающих систем*. Разработана в середине 60-х гг. в США. Проект PLATO включал как разработку технических средств для применения компьютера в *обучении* (в первую очередь *интерактивных терминалов*), так и создание *программного обеспечения*. В рамках проекта были разработаны *дисплеи* на плазменных трубках, обладающие хорошей разрешающей способностью (512 на 512 одноцветных точек), с возможностью вывода *текста, графических изображений и анимации*. Кроме того, дисплей предоставлял возможность выбора точки на экране путем простого нажатия пальцем в искомое место экрана (*тактильный ответ*). Программное обеспечение включало *систему программирования для авторского языка программирования Тьютор* и набор *сервисных программ*. В системе PLATO написано большое количество учебных программ. Имеются реализации системы на различных типах компьютеров. Язык *Тьютор* породил целое семейство авторских языков, которые ныне активно используются при создании *программ учебного назначения*.

В.А.Петрушин.

ПЛОТТЕР — то же, что и *графопостроитель*.

ПЛЭНЕР (англ. PLANNER — произвольная аббревиатура от “A Language for Proving Theorems” — язык для доказательства теорем) — функционально-императивный язык программирования, предназначенный для разработки сложных интеллектуальных систем. Разработан в 1971 амер. ученым К.Хьюитом. В области обучения может применяться для создания лингвистических процессоров и экспертных обучающих систем.

П. является развитием языка Лисп, дополняя его встроенными механизмами ассоциативной выборки и дедукции. Осн. типы данных — атомы и формы. Форма — это выражение, которое можно вычислить и получить значение. Выражение — общее название атомарных и списковых выражений (единственный вид конструкций, используемых в П. для изображения программ и данных). Формы бывают простые и сегментные. Форма наз. простой, если ее значением является одно выражение, и сегментной, если ее значением является последовательность выражений. Типы простых форм: атомы (значение атома — атом); “.” — переменные; (.X — значение X); “:” — глобальная константа; (:Y — значение Y); “(L)” (L — список); значение списка (x,y,...,z) — список (значение x, значение y,..., значение z); “[P]” (P — список, значение функции); значение формы [функция] — значение функции. Типы сегментных форм: “!.” (сегментированное значение); если X=[a(bc)], то значение !.X=a(bc); “!:” (сегментированное значение указанной константы); “<” (сегментированное значение функции); значение формы <функция> — ее сегментированное значение.

Кроме традиционных для Лиспа управляющих структур, в П. существуют оператор перехода (GO) и оператор возврата из блока (RETURN), а также режим возвратов, моделирующийся следующими осн. функциями: AMONG, ALT — различные определения развилки; FALL — возврат к развилке в случае неуспеха; MESS, GATE, UNFALSE — вырабатываются логические значения при неуспешном завершении функции; PERM — условное уничтожение всех развилки; STRG — условное уничтожение развилки внутри аргумента; FP — именование развилки. Образцы создаются на основе форм и бывают двух типов: простые и сегментные. Простые образцы сопоставляются только с одним выражением, сегментные — с последовательностью выражений. Возможно также сопоставление образца с образцом.

Дедуктивный вывод реализован с помощью аппарата вызова процедуры по образцу, т.е. некоторой плэнерной процедуре ставится в соответствие образец. Такая конструкция наз. теоремой. Теоремы вызываются с помощью операторов, определяющих их тип: ACHIEVE — целевые теоремы; DRAW — записывающие теоремы; CHANGE — вычеркивающие теоремы. Интерфейс с пользователем осуществляется на уровне функций ввода—вывода.

В.А.Третьяк.

ПОДПРОГРАММА — программа, для которой существует предписание вызова в другой программе. На содержательном уровне реализуется алгоритм решения некоторой частной по отношению к решаемой задачи.

В.А.Третьяк.

ПОЗНАВАТЕЛЬНАЯ ЗАДАЧА — задача, решаемая в процессе познания или его технического моделирования. Чаще всего о П.з. говорят в тех случаях, когда предусматривается приобретение субъектом информации, рассчитанной на длительное хранение в его памяти (либо в памяти общества, если речь идет о профессиональной деятельности ученого). Вместе с тем термин “П.з.” употребляется и в более широком смысле — в связи с рассмотрением любых задач, решаемых посредством восприятия, мышления, воображения и др. познавательных процессов. В рамках общей теории задач П.з. трактуется еще шире — как задача усовершенствования знания, принадлежащего любому решателю (а не только человеку). Точнее, в любой П.з. требуется усовершенствовать информацию, которую несет это знание о моделируемой им системе, т.е. об объекте познания. Это усовершенствование можно трактовать как обеспечение достаточной полноты упомянутой информации, причем она должна обладать, во-первых, достаточным объемом и, во-вторых, достаточной адекватностью, т.е. соответствием объекту познания.

Знание, служащее предметом П.з., можно описать как систему взаимосвязанных компонентов-моделей двух типов. В исходном состоянии предмета задачи только компоненты первого типа несут достаточно полную информацию о соответствующих компонентах объекта познания; информация, которую несут компоненты второго типа, является недостаточно полной. Компоненты объекта познания, моделируемые компонентами-моделями первого и второго типов, принято называть соответственно известными и неизвестными объектами. Рассмотрим простой пример П.з. — задачу, сформулированную так: “В прямоугольном треугольнике ABC длина гипотенузы AC составляет 10 см, а длина катета AB — 6 см. Найти площадь треугольника”. Предмет этой задачи в его исходном состоянии можно представить с помощью таблицы; каждую ее строку можно отождествить с одним из компонентов-моделей, составляющих этот предмет. Вместе с тем каждая строка таблицы описывает компонент объекта познания (треугольника), моделируемый соответствующим компонентом-моделью.

В рассматриваемой задаче длина гипотенузы AC и длина катета AB описаны достаточно полно — это известные объекты. Недостаточно полно описаны длина катета BC и площадь треугольника ABC — это неизвестные объекты. В строках таблицы, описывающих эти объекты, имеются незаполненные (обозначенные вопроситель-

Наименование объекта, моделируемого компонентом-моделью	Математическая характеристика объекта	Единица измерения	Численное значение
Длина гипотенузы AC	положительное число	см	10
Длина катета AB	То же	—"	6
Длина катета BC	" "	—"	?
Площадь треугольника ABC	" "	см ²	?

ными знаками) клетки. Связи между моделями известных и неизвестных объектов обеспечиваются в данном случае теоремой Пифагора и формулой для вычисления площади прямоугольного треугольника. Благодаря этим связям информация о неизвестных объектах содержится не только в их моделях, входящих в предмет задачи (прямая, или эксплицитная информация), но и в связанных с ними моделях известных (и других неизвестных) объектов — косвенная (имплицитная) информация. Требования П.з. состоят в переводе всех или некоторых из компонентов предмета задачи — моделей неизвестных объектов — в разряд моделей известных объектов. Неизвестные объекты, к моделям которых относится такое требование, — это искомые объекты. Всякая П.з. требует пополнения содержащейся в некотором знании (в предмете этой задачи) прямой информации об искомым объектах. Неизвестные объекты, не являющиеся искомыми в рассматриваемой задаче, могут быть искомыми в ее подзадаче (такова в приведенном примере длина катета BC). Неизвестные объекты подобного рода наз. промежуточными, или вспомогательными неизвестными. В П.з. часто моделируются и такие неизвестные объекты (в матем. задачах это так наз. неопределенные неизвестные) которые не являются искомыми ни в одной из ее подзадач, но используются для установления связей между известными и искомыми объектами (точнее, между их моделями).

В результате решения П.з. ее предмет оказывается содержащим достаточно полную (т.е. обладающую достаточными объемом и адекватностью) прямую информацию об искомым объектах. Если же обеспечивается достаточный объем, но не достигается достаточная адекватность указанной информации, то имеет место псевдорешение рассматриваемой задачи (обычно в таких случаях говорят о нечетком или ошибочном решении). Решение П.з. может быть формально описано как превращение некоторой высказывательной функции в истинное высказывание, а псевдорешение — как превращение ее в ложное высказывание.

Степень представленности компонентов П.з. в ее формулировке может быть различной. Это проявляется, в частности, в том, что хотя известные объекты достаточно полно (в отличие от неизвестных) представлены (смоделированы) в П.з., вовсе не обязательно, чтобы они были представлены столь же полно в ее формулировке, где они могут лишь подразумеваться. Вместе с тем далеко не все компоненты формулировки в общем случае существенны для решения задачи. П.з. могут решаться либо без доступа к внешней информации (в этом случае в процессе решения может использоваться только информация, заключенная в самом решателе и в формулировке или иной предъявленной решателю модели задачи), либо такой доступ может быть предоставлен. Пополнить прямую информацию об искомом объекте можно разными путями, в т.ч.: а) используя связи между компонентами-моделями, входящими в состав предмета задачи, и преобразуя, благодаря такому использованию, косвенную информацию об искомым объектах в прямую; б) извлекая недостающую прямую информацию из системы, моделируемой предметом задачи, т.е. из объекта познания; в) генерируя недостающую прямую информацию. При решении любой П.з., независимо от возможного использования др. путей, путь "а" всегда требуется. Во многих случаях, в частности для решения матем. задач, в принципе достаточно использовать только этот путь. Сопоставляя выделенные пути решения П.з. с особенностями психических процессов, можно констатировать, что преимущественное использование пути "а" характерно для мышления, пути "б" — для восприятия, пути "в" — для воображения. Этот последний путь при прочих равных условиях может играть тем большую роль в решении П.з., чем больше для каждого искомого объекта может быть найдено приемлемых (удовлетворяющих требованию П.з.) результатов решения. Ситуации, когда их может быть много, характерны для задач, решаемых в процессе творческой деятельности, особенно в области искусства и техники. См. также *Вопрос, Закрытые и открытые задачи, Трехкомпонентная задача.*

Г.А.Балл.

ПОКАЗАТЕЛИ КАЧЕСТВА ПРОЦЕССА ОБУЧЕНИЯ — набор параметров (показателей, критериев) концептуальной модели состояния знаний *обучаемых* (см. *Концептуальная модель процесса обучения*), характеризующих степень достижения целей *обучения*. Значения этих показателей определяются расчётно-аналитическим, опытно-регистрационным или экспертным способом. При оценке их величины можно использовать шкалы наименований, порядка, интервалов и отношений. При обработке результатов измерений необходимо следить за допустимостью статистических операций над данными различных шкал.

Интегральным критерием качества может служить степень близости результатов работы с системой к максимально возможным

$$n = (R_{\max} - R_k) / R_{\max},$$

где R_{\max} — максимально возможная величина вектора результатов, а R_k — расстояние от точки, соответствующей полученным результатам, до цели в критериальном пространстве. Степень овладения знаниями, умениями, навыками определяется по результатам выполнения учебных заданий или специального тестирования. Оценка качества обучения проводится в четыре этапа: уточнение цели оценки; выбор набора частных показателей качества; оценка значений частных значений качества; оценка значения интегрального показателя.

Ю.И.Лобанов.

ПОЛЬЗОВАТЕЛЬ — специалист, использующий в своей работе пакеты программ различного назначения. В области компьютерного обучения принято выделять типы П. по их отношению к автоматизированным учебным курсам (АУК): разработчик автоматизированных обучающих систем (АОС), автор АУК, диспетчер АОС, преподаватель, обучаемый.

В.А.Третьяк.

ПОНИМАНИЕ — 1) Способность постичь смысл и значение чего-либо, и достижения вследствие этого определенного результата. 2) При взаимодействии человеко-машинном определение целей пользователя, преследуемых при высказывании данного текста, или отношение, связывающее систему, задачу и знания. Включает в себя узнавание языковых средств, осмысление содержания и направлено на раскрытие связей и отношений реальной действительности. В более узком смысле П.— это установление соответствия между семантическими и грамматическими системами отправителя и получателя сообщения. П.— необходимая предпосылка общения, т.к. отсутствие корректного восприятия и адекватной реакции на каждое сообщение собеседника приводит к замене диалога совокупностью перемежающихся монологов, семантически не связанных между собой. При человеко-машинном взаимодействии наиболее сложна проблема П. компьютером сообщений пользователя. В отношении к компьютеру однозначная трактовка термина “П.” на концептуальном уровне не найдена, но существует ряд релевантных определений. Понятность сообщения определяется как синтаксическая, семантическая, прагматическая недвусмысленность сведений, требующая общности языка, общности знаний коммуникантов об окружающем мире, а также о себе и о партнере. На уровне системного анализа можно зафиксировать также требования: система должна обладать знаниями о языке (*модель языка*), знаниями

энциклопедического характера (*модель окружающего мира*), знаниями о партнере (*в обучающих системах — модель обучаемого*), знаниями о себе (*модель системы*), т.е. саморефлексией.

На операциональном уровне возможна трактовка П. текста как возбуждения этим текстом узлов *семантической сети*, отражающих используемые в тексте *понятия* и отношения и *выстраивающихся* в определенные *фреймы* и сценарии. Специфика проблемы П. при *компьютерном обучении* связана с необходимостью анализа *высказывания обучаемого* в разных аспектах *обучения*: правильность результата, усвоение способа решения, логика рассуждения, достижение ближайших и отдаленных целей обучения. Требуется способность диагностировать возможные причины ошибки и выбрать путь коррекции, адекватную форму и меру помощи. При обучении П. имеет иерархическую структуру, задающую подчиненность решающей (предметной) подсистемы педагогической подсистеме, управляющей обучением и включающей существенную психолого-педагогическую *информацию*: цели и методы обучения, история и модель обучаемого.

Глубина П. характеризуется возможностями системы в плане решения определенного круга задач и способностью к использованию тех или иных знаний. По соотношению решаемых задач и используемых знаний выделяют семь уровней П.: от информационного поиска без изменения формата внутренних знаний, с использованием лишь сведений о синтаксисе и семантике языка общения, до генерирования новых видов *представления знаний* и *операций* для решения новых задач в различных *предметных областях*, с усвоением нового знания и модификацией внутреннего представления знаний. Среди реально существующих наиболее глубоким уровнем П. обладают *экспертные системы*, системы с автоматическим модифицированием и расширением запаса эвристических правил и др. В обучающих системах необходима большая, чем в обычных, глубина П., т.к. оно опосредуется ситуацией решения *учебной задачи* и характеристиками ситуации обучения (цели, методы, история обучения, модель обучаемого). Обучающая система даже самой узкой специализации и ограниченных задач обучения должна реализовать по крайней мере содержательное П., т.е. использовать содержательные определения объектов и отношений окружающего мира.

Понимание системой сообщений пользователя можно улучшить различными способами. К их числу относятся: входной фильтр, осуществляющий автоматический анализ и коррекцию грамматических, синтаксических и прочих ошибок; задание классов семантической эквивалентности, охватывающих все лингвистическое разнообразие возможных входных сообщений, что снимает зависимость П. от формы сообщения; введение некоторых градаций П. от полного и безусловного до вероятностного, приблизительного. От обучающих систем требуется большая гибкость в сфере П., чем от систем общего

назначения, поскольку предложенный обучаемым путь решения, не предусмотренный в системе, может оказаться не ошибочным, требующим обязательной коррекции, а нетривиальным, требующим включения в базу знаний системы.

Е.Ю.Комиссарова.

ПОНЯТИЕ — фрагмент знаний, представляющий собой обобщенную модель определенного элемента предметной области. В П. фиксируются общие и существенные признаки объектов и явлений, представление их классов. С помощью П. осуществляется распознавание, обобщение, структурирование информации, поступающей в решатель, связывание ее с информацией, хранящейся в системе. Формирование П. является важной проблемой искусственного интеллекта и относится к методам представления внешнего мира.

Выделяют несколько подходов к формированию П. Прежде всего это индуктивное обобщение моделей элементов системы. П. формируются также путем выделения и обобщения моделей ситуаций, включающих элементы мира, относящиеся к данному П. Так формируются абстрактные понятия. П. строятся также путем выделения пересечений, объединения, связывания различных П. отношениями. При решении задач этот процесс определяется целями и подцелями задач. Разновидностью такого процесса является формулирование определений, с помощью которых вводятся научные и технические П. Множество элементов мира, определяемое П., образует объем П. Существуют объединительные (присущие всем или многим элементам объема П.) и разделительные (те, которые за пределами объема П. не встречаются вообще или встречаются редко) признаки элементов объема П.

Одной из важнейших является операция по распознаванию принадлежности объектов объему П. Обучение П. часто отождествляется с обучением правилам опознавания объектов. Действия по опознаванию могут быть представлены в процедурной или декларативной формах. В первом случае задается последовательность операций и действий, приводящих к распознаванию объектов, во втором — перечень и взаимосвязи признаков, характеризующих объекты П.

Современное понимание роли П. в формировании баз знаний и решении задач значительно расширяет ранее распространенное представление о П. лишь как об инструменте распознавания образов, причем необходимо учитывать, что при изменении соответствующей научной теории может изменяться и уточняться содержание П. Работа с П. и их предварительный анализ взаимосвязаны. Анализ включает информационный анализ предметной области, ее структуризацию, учет информационных потребностей пользователей. Один из вопросов, возникающих при этом, — чем обуславливается структуризация

П, и где границы такой структуризации. При разработке *обучающих систем* необходимо принимать во внимание, что исходная проблема задается П., которое необходимо сформировать у *обучаемого*, — цель *обучения*, и перечнем П., которые уже известны, — *знаниями* обучаемого. Ответ обучаемого может быть и новым понятием. См. также *Система понятий*.

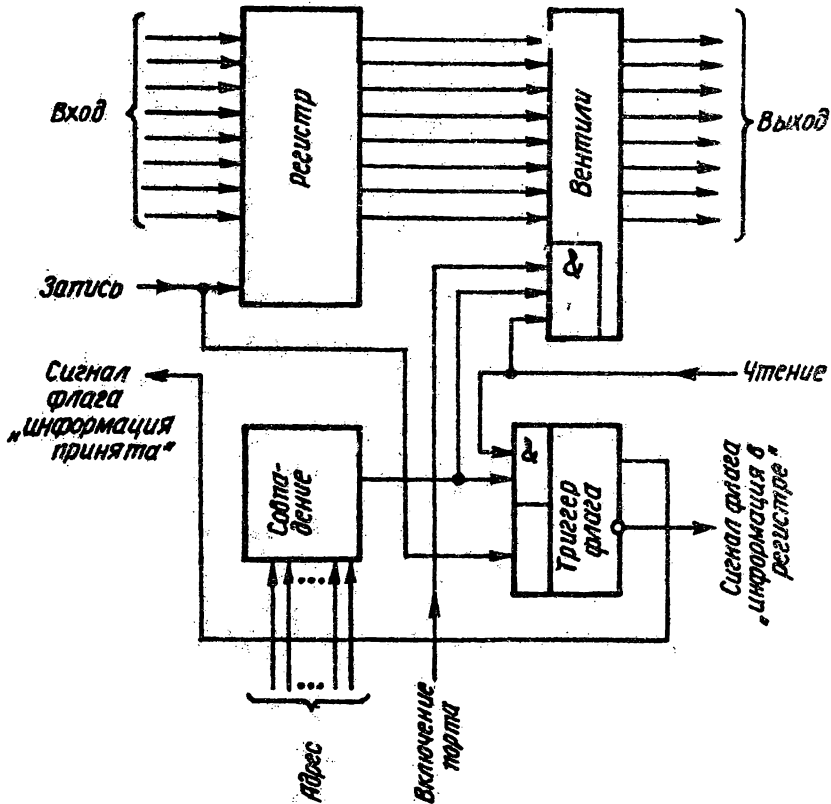
А.Ф. Манако.

ПОНЯТИЙНО-ТЕРМИНОЛОГИЧЕСКАЯ ИГРА — *учебная игра*, с помощью которой осваивается понятийная структура *предметной области* на базе ее *тезаурусно-сетевой модели*. Стимулирует самостоятельную познавательную деятельность, расширяет представления о сфере применения как имеющихся, так и новых *знаний*, позволяет преодолеть ограниченность понятийно-терминологических ресурсов.

Функции П.-т.и.: формирование и развитие профессионально необходимых теор. знаний и умений; активизация поиска связей, противопоставлений, аналогий между изучаемым *понятием* и известными фактами и явлениями, актуализация системы имеющихся знаний, их систематизация и переконструирование; соотнесение базовых понятий с ситуациями практической деятельности, выявление их взаимосвязи и взаимообусловленности; выявление новых, ранее неизвестных *обучаемому*, характеристик понятий, областей их применения. Средствами компьютерной поддержки П.-т.и. являются автоматизированные словарные системы (см. *Автоматизированный словарный комплекс учебного назначения*).

С.Л. Васильченко.

ПОРТ (франц. La porte — дверь) — устройство для обмена *информацией* между ЭВМ и *устройствами ввода — вывода информации*. В простейшем случае — регистр, группа вентиляей, элемент совпадения *адреса* и триггер флага, соединенные между собой соответствующим образом (рис.). Различают П. ввода и вывода. П. действует в режиме *диалога*. *Процессор* выставляет адрес данного П. Адрес элементом совпадения превращается в сигнал, разрешающий открывать вентили и устанавливать триггер в исходное состояние. Передающий объект записывает информацию в регистр и перебрасывает тем самым триггер. Принимающий объект обнаруживает, что триггер флага переброшен и, следовательно, в регистре — достоверная информация, после чего открывает вентили, принимает информацию из регистра и перебрасывает триггер в исходное состояние. Передающий объект, обнаружив, что триггер флага в исходном состоянии (это означает, что информация принята), записывает в регистр следующую порцию информации, перебрасывая триггер, и цикл повторяется. Количество циклов и порядок их следования определя-



ются программой работы ЭВМ. Возможен П. без триггера, но в этом случае синхронизация работы внешнего устройства и центрального процессора ЭВМ должна быть обеспечена иным способом.

В.Л.Леонтьев.

ПОЭТ (Программа Обработки Экономических Текстов) — промышленная диалоговая система для анализа естественных языковых экономических текстов. Словарь, семантическая сеть и база данных (БД) системы предназначены для описания слов, объектов внешнего мира и связывающих их отношений. Комплекс обрабатывающих программ осуществляет процедуры распознавания входной информации, генерирует ответы на языке общения, организует доступ к словарю, семантической сети и БД. Знания о языке общения разделяются на лингвистические и проблемные. Лингвистические знания хранятся в основном в различных зонах словаря системы (морфологической, синтактико-семантической и т.д.), проблемные

знания — в семантической сети. Различают абстрактную сеть (хранит информацию о множестве состояний *предметной области*, выраженных через общие *понятия*) и конкретную сеть (конкретизирует понятия абстрактной сети). Входное предложение анализируется на этапах *морфологического анализа, синтаксического анализа и семантического анализа*, причем на этапе семантического анализа синтаксическая структура входного сообщения трансформируется в семантический граф, состоящий из множества вершин-понятий, связанных между собой через вершины-события и характеристики. На этапе семантической интерпретации определяется *контекст* запроса, извлекается информация из конкретной сети, формируются обращения к БД за числовой информацией и определяется способ обработки этой информации (суммирование, вычисление процента и т.д.). Система П. предназначена для генерации из ее модулей конфигураций для конкретных применений. В ней используются языки *ПЛ/1* и *ассемблера язык*.

О.Н.Золотопул.

ПРЕДМЕТНАЯ ОБЛАСТЬ (ПО) — часть реального мира, в которой поставлена и будет решаться *задача*, отражающая объекты, атрибуты этих типов объектов и связей между ними. Ключевым моментом является описание ПО и *знаний* о ней. Основу ПО составляют: предметы (все то, что может быть как-то воспринято, представлено, названо исследователем), которые могут находиться в различных состояниях, определяемых различными значениями свойств объектов; воздействия на предметы, т.е. превращения одного предмета в другой; параметры. Элементами ПО, отражающими общепринятые формальные средства, являются: абстрактные объекты; *алгоритмы*; конструкторы алгоритмов; утверждения об объектах; доказательства (выводы утверждений); конструкторы доказательств (*стратегии*).

Описания ПО условно можно назвать знаниями системы; для решения конкретной задачи требуются знания о конкретной ПО. Выделяют два типа ПО. К первому типу относятся ПО, в которых выделена четкая система *понятий* и достаточно полный набор свойств, а объекты, с которыми приходится иметь дело, имеют четкую структуру и хорошо описываются в выделенной системе понятий и свойств. Если же объекты ПО не имеют четкой структуры и/или в них отсутствует выделенный набор свойств, позволяющих описывать все объекты, то такие ПО относятся к плохо структурированным.

А.Ф.Манако.

ПРЕДСТАВЛЕНИЕ ЗНАНИЙ — способ формального выражения *знаний* человека, позволяющий хранить и обрабатывать знания в системах *искусственного интеллекта*. При П.з. в *компьютерной*

технологии обучения рассматриваются различные виды знаний, представимые для машинной обработки: учебные знания (см. *База знаний учебного назначения*), лингвистические знания (см. *Лингвистическое обеспечение, Лингвистический процессор*), экспертные знания, знания инструментальной системы (см. *База знаний инструментальной системы*). Существуют различные методы представления знаний, реализуемые посредством программных средств представления и обработки знаний (см. *Инструментальные средства для представления знаний*). Создаются специальные структуры П.з. (см. *Учебные структуры знаний*), связанные с построением фреймов. П.з., относящееся к некоторой конкретной области, базирующееся на учебных структурах знаний, несет в себе главным образом информацию о содержательной стороне дела и структуре понятий (см. *База знаний, Метазнания, Приобретение знаний, Язык представления знаний*).

С.П.Кудряцева.

ПРЕДСТАВЛЕНИЕ КОНЦЕПТУАЛЬНЫХ ЗНАНИЙ — формализованное описание *концептуальных моделей* изучаемых объектов и явлений. При формализации каждый элемент концептуальных моделей описывается семиотической схемой:

$$Z = \langle N, E, I \rangle,$$

где N — имя объекта; E — внешнее описание объекта (множество схем внешних отношений); I — внутреннее описание объекта (множество схем внутренних отношений). Внешнее описание модели определяет положение объекта в пространстве и времени. Внутреннее описание отражает связи элементов модели. При описании используется схема тернарного отношения $S = X * Z * Y$, где X, Y, Z — множества входных, выходных и внутренних (соответственно) состояний объекта. В каждый момент времени элемент системы находится в состоянии $z \in Z$ и реализует некоторую внешнюю функцию $f : X * Z \rightarrow Y$. После выполнения операции элемент изменяет собственное состояние в соответствии с переходной функцией $f : X * Z \rightarrow Z$. Объекту, рассматриваемому как система, ставятся в соответствие внешние и внутреннее описания. Внешнее описание определяет объект как подсистему (элемент) системы высшего ранга. Внутреннее описание содержит *информацию* о внутренней структуре системы и ее функциях.

Ю.И.Лобанов.

ПРЕПОДАВАТЕЛЬ — тот, кто осуществляет преподавание, обучение. В роли П. может выступать и *обучающая система*. В условиях *компьютерной технологии обучения* П., как правило, является специалистом по учебному предмету, для которого написан

автоматизированный учебный курс. Его задача — помогать обучаемым на занятиях в дисплейном классе.

О.П.Платонова.

ПРИКЛЮЧЁНЧЕСКАЯ ИГРА — развлекательная игра, характеризующаяся наличием сюжета и правил, предполагающих преодоление различных препятствий для достижения игровых целей. См. также *Учебная игра*.

М.Илиева, В.Цонева.

ПРИНТЕР (от англ. printer — печатать) — то же, что и *печатающее устройство*.

ПРИНЦИП ДОСТУПНОСТИ ОБУЧЕНИЯ — принцип обучения, обеспечиваемый соответствием содержания и объема усваиваемых знаний и умений уровню предшествующей подготовки (общеобразовательной и профессиональной) и познавательным особенностям обучаемых. При реализации П.д.о. необходимо следовать от легкого к трудному, от известного к неизвестному, от простого к сложному. Доступность обучения — категория историческая. Разработка новых методов, приемов и средств обучения приводит к тому, что определенный учебный материал, ранее считавшийся недоступным для определенной группы обучаемых, становится доступным для усвоения. Или аналогично материал, доступный ранее группе обучаемых данного возраста, ныне может быть усвоен учащимися более младшего возраста. Так, обобщение передового учебно-воспитательного опыта свидетельствует, что содержание учебного материала, обеспечивающего овладение основами *компьютерной грамотности*, доступно школьникам 7—8-х классов, а не только 9—10-х, как это считалось ранее. Общее правило адресации выбора средств *компьютерной технологии обучения* для учебных заведений всех уровней состоит в том, что ее методы и средства в основном соответствуют целям и содержанию подготовки каждой из категорий обучаемых по *информатике* и вычислительной технике и, следовательно, определяются профессионально-квалификационными требованиями.

Н.М.Когдов, Н.М.Розенберг, А.Я.Савельев, В.А.Стаканов.

ПРИНЦИП НАГЛЯДНОСТИ — принцип обучения, реализуемый путем предъявления модели изучаемого объекта или процесса в визуальной форме, наиболее четко раскрывающей существенные связи и отношения объекта. Эти связи и отношения можно акцентировать компьютерными средствами (цветом, миганием, звуком и пр.). В практической работе учебных заведений используют многие виды наглядности. Одна из них — существенная, частным

случаем которой является экспериментальная наглядность (напр., ознакомление обучаемых с работой электронно-лучевой трубки или станка с программным управлением). Важное значение при компьютеризации имеют такие виды наглядности, как картинная и картинно-динамическая, звуковая, символическая, графическая. Компьютерная технология обучения сочетает многие виды наглядности, и задача состоит в оптимизации их использования. Когда речь идет о познании, научном или учебном, следует говорить о наглядном образе как единстве чувственного и рационального. В профессиональной подготовке особую роль играет символическая и графическая наглядность. Прежде всего речь идет о специфических особенностях содержания профессионально-технического образования, в котором важную роль играют принципиальные и монтажные схемы (профессии, связанные с электро- и радиотехникой) или кинематические схемы (профессии, связанные с металлообработкой и машиностроением).

Особенности проявления П.н. в условиях компьютерного обучения весьма многообразны. Одна из них связана со взаимодействием обучаемых и картинки на дисплее, включающей вербальный текст, графики, кино- и диафрагменты, фрагменты мультфильмов и др. Изображения на экране электронно-лучевой трубки можно подчеркнуть приемами специфически компьютерной техники. Несколько иначе проявляется П.н. при работе печатающего устройства, редактировании полученного с его помощью текста. Важное значение при этом имеют формы рационального сочетания слова преподавателя и средств наглядности. Только при тесном взаимодействии слова и наглядных образов повышается роль каждого из них. Чем труднее материал для усвоения, тем шире должны использоваться приемы обеспечения наглядности.

Н.М.Когдов, Н.М.Розенберг, А.Я.Савельев, В.А.Стakanов.

ПРИНЦИП НАУЧНОСТИ ОБУЧЕНИЯ — принцип обучения, выполняющийся, если отбор содержания учебного материала по данному предмету, как и методы его усвоения, адекватны современному научному знанию. В соответствии с П.н.о. обучаемые должны овладеть современными способами и приемами научного поиска и профессиональной деятельности. Применительно к компьютерной технологии обучения целесообразно в обучающей программе для ЭВМ использовать такие методы, как системный анализ, моделирование и т.д. Из П.н.о. вытекает требование проблемно-деятельностной постановки и внедрения компьютерного обучения, направленного преимущественно на использование вычислительной техники для формирования интеллектуальных и моторных умений и навыков, соответствующих задачам обучения в данной предметной области и профессионально-квалификационным требованиям. Такая

организация обучения, реализуемая на основе моделирования изучаемых явлений и систем в соответствии с П.н.о., формирует у обучаемого качества исследователя, конструктора, испытателя, способствует развитию аналитического, логического мышления, формированию умений системного анализа технических, экономических, социальных и комплексных (управленческих) проблем, умений находить эффективные методы их решения.

Н.М.Когдов, Н.М.Розенберг, А.Я.Савельев, В.А.Стаканов.

ПРИНЦИПЫ ОБУЧЕНИЯ — представленные в наиболее общем виде основные идеи, правила, требования к процессу обучения, выполнение которых обеспечивает его эффективность. Среди П.о. наибольшее значение имеют *принцип доступности обучения, принцип наглядности, принцип научности обучения, принцип сознательности и активности обучаемых, принцип системности, принцип учета индивидуальных различий обучаемых*, принципы связи с жизнью, воспитывающего характера обучения, прочности знаний и умений. Под влиянием научно-технического прогресса и социальных изменений П.о. непрерывно изменяются, уточняются, переосмысливаются и дополняются. Одним из важных факторов такого совершенствования в середине 80-х гг. стало интенсивное внедрение информационных технологий в теорию и практику обучения.

Е.Д.Маргулис.

ПРИНЦИП СИСТЕМАТИЧНОСТИ И ПОСЛЕДОВАТЕЛЬНОСТИ — *принцип обучения*, выполнение которого означает, что логика содержания обучения должна следовать систематичности и последовательности. В каждом учебном предмете средней или высшей школы имеется система взаимосвязанных *понятий, суждений, закономерностей, вытекающих одно из другого*. Идеи систематичности реализуются уже на уровне учебного плана, поскольку на данном этапе должна быть обеспечена систематичность и последовательность расположения учебного материала по предметам, годам и семестрам. Более детализированно эта задача решается на последующих уровнях реализации содержания образования — уровнях учебной программы. Систематичности и последовательности расположения учебного материала способствует обеспечение внутрипредметных и межпредметных связей. В результате у *обучаемых* будет формироваться целостное представление об объеме знаний и умений. На тактическом уровне задачи обеспечения систематичности и последовательности были решены в 60 — 70-х г. в условиях широкого распространения *программированного обучения, программированных учебных материалов*. *Обучающая программа* представлена четкой последовательностью кадров (шагов, фрагментов учебного материала), располо-

женных так, что n -шаг служит частным основанием для $n+1$, $n+2$, шагов, а сам тесно связан с $n-1$, $n-2$, шагами.

Н.М.Розенберг.

ПРИНЦИП СИСТЕМНОСТИ — принцип обучения, реализующий общенаучный метод системности, логику системного раскрытия объектов и явлений. Из П.с. вытекает требование выделять в объектах или явлениях, предъявляемых средствами компьютерной технологии обучения, осн. структурные элементы, их свойства и существенные связи между ними, позволяющие целостно представлять данный объект; при этом деятельность обучаемого должна отражать логику системного анализа объекта.

Н.М.Когдов, А.Я.Савельев, В.А.Стakanов.

ПРИНЦИП СОЗНАТЕЛЬНОСТИ И АКТИВНОСТИ ОБУЧАЕМЫХ — принцип обучения, выполнение которого означает, что обучаемые сознательно относятся к обучению средствами обучающей программы. Во многом определяет результативность компьютерной технологии обучения (КТО). Сознательность предполагает обязательное понимание, осмысление учебного материала. Мерой сознательности является не запоминание учебного материала, а умение разумно применять знания и умения для решения вначале достаточно стандартных, а затем и продуктивных, творческих задач. Сочетание сознательности и активности, апелляция к ведущим мотивам учебной деятельности обеспечивает развитую познавательную активность, единство мышления и учебной деятельности. При использовании человеко-машинных систем выполнение П.с.и а.о. проходит весьма специфично. Обучающая программа компьютера ставит перед обучаемым задачи, а операторную часть действия выполняет компьютер, причем это делается применительно к различным учебным предметам. Совершенно очевидно, что полное обеспечение сознательности и активности определяется качеством обучающей программы. Уровень этого обеспечения зависит от создания проблемных ситуаций, ставит обучаемых перед необходимостью аргументировать, составлять планы, тезисы, аннотации и др. Понятие сознательности и активности в обучении тесно связано с понятием самостоятельности обучаемых. П.с.и а.о. призван формировать у обучаемых опыт творческой деятельности. Известно, что концепция содержания образования, выдвинутая группой московских дидактов, включает опыт творческой деятельности как один из четырех обязательных элементов содержания. Если ставится задача формирования стандартных типов умений, то развитие познавательной активности осуществляется по готовому алгоритму для КТО. Если же предполагается формирование умений решать эвристические задачи, то обучаемому предоставляют

возможность самостоятельно построить алгоритм действий; компьютер в таком случае выступает в роли “эксперта”, “контролера”.

Н.М.Розенберг.

ПРИНЦИП УЧЁТА ИНДИВИДУАЛЬНЫХ РАЗЛИЧИЙ ОБУЧАЕМЫХ — *принцип обучения*, предусматривающий необходимость построения обучения с учетом индивидуальных различий обучаемых, влияющих на эффективность усвоения содержания образования. Речь идет о темпе усвоения, общих и специальных способностях, интеллекте, предшествующей подготовке. В таком понимании П.у.и.р.о. включает требования учета и последующего формирования мотивационного, эмоционально-волевого и интеллектуального компонентов личности. Необходимость учета индивидуальных различий неизменно постулируется в теор. работах, однако реализовать его в условиях практического обучения довольно трудно. Положение улучшилось с применением *программированного обучения*, появлением программированных учебников и обучающих машин, обеспечивающих определенный уровень индивидуального обучения. Уровень индивидуального обучения значительно возрос при обеспечении *компьютерной технологии обучения*. Реализуемый средствами КТО как в условиях самостоятельной, так и в условиях коллективной учебной работы, П.у.и.р.о. предусматривает автоматизацию индивидуального планирования обучения, учета прохождения его этапов контроля усвоения. Важнейшим условием эффективного машинного обучения является обеспечение адаптивных способностей *автоматизированных обучающих систем* на основе рефлексивного управления. При этом управлении *обучающая система* автоматически формирует *модель обучаемого* (напр., в виде структуры усвоенных умений, установленных особенностей восприятия, мышления, памяти) и на этой основе оптимизирует план и тактику обучения, темп и характер *диалога*, меняет методы *мотивации* и способов *предъявления информации*.

Н.М.Когдов, Н.М.Зозенберг, А.Я.Савельев, В.А.Стаканов.

ПРИНЦИПЫ ПРИМЕНЕНИЯ СРЕДСТВ ОБУЧЕНИЯ — *правила*, позволяющие квалифицированно применять средства обучения (СО) в учебном процессе с учетом их дидактических возможностей. По аналогии с *принципами обучения* П.п.с.о. даются условные названия, частично отражающие их содержание: “принцип обусловленности”, “принцип необходимости”, “принцип соответствия”, “принцип информативности”.

Принцип обусловленности. Средства обучения следует применять в учебном процессе, для которого четко определены условия на входе, цели на выходе, а также подцели и способ применения на промежуточных этапах. Понятие “условия на входе” лаконично отражает уровень *знаний, умений и навыков, необходимый*

обучаемому для успешного восприятия нового учебного материала. “Цели на выходе” — цели данного учебного занятия, соответствующие целям итогов обучения (см. *Целевое обучение*), если СО используются без учета предварительных знаний обучаемых и определенности целей обучения. Применение в таких условиях СО носит формальный характер, т.к. у преподавателя нет реальных возможностей четко определить их роль и место. Поэтому здесь СО могут применяться либо как самоцель (преподаватель просто стремится показать, что он использует СО), либо, в лучшем случае, по интуиции преподавателя (преподаватель предполагает: “Вероятно, в этих условиях можно применить такое-то СО”). Принцип обусловленности снимает эту неопределенность: еще до выбора конкретного СО должны быть определены условия, без которых такой выбор целенаправленно сделан быть не может. Существенно, что П.п.с.о. только в комплексе полностью определяют обстоятельства выбора СО, а каждый принцип в отдельности определяет лишь какой-то один аспект.

Принцип необходимости: СО должны применяться только в соответствии с реальными потребностями учебного процесса на данном конкретном этапе, для достижения конкретного педагогического эффекта. Если преподаватель спланировал учебный процесс на основе первого принципа, он четко представляет, чего хочет добиться от обучаемых при изложении того или иного вопроса. Напр., необходимо, чтобы студент знал формулу, требуемую для расчета к.-л. величины. Для достижения этой цели, очевидно, достаточно доски и мела. Если студент должен запомнить формулу, нужно, чтобы последняя была записана в конспект и, если запоминанию поможет некоторое мнемоническое правило, его вместе с формулой следует изобразить на доске, показать на плакате и т.п. Для понимания физического смысла закона недостаточно запоминания выражающей его формулы; преподавателю следует решить, какими средствами наиболее доступно и убедительно можно раскрыть содержание закона. В одном случае будет достаточно плаката, слайда, в ином — не обойтись без показа явлений, подчиняющихся рассматриваемому закону, в динамике: нужны кинофильм, лекционные демонстрации или натурный показ. Следовательно, принцип необходимости указывает на то, что в условиях учебного процесса СО должны быть подчинены целям обучения. Иначе говоря, СО следует использовать только тогда, когда они действительно необходимы из дидактических соображений. Конкретизация целей применения СО определяет не только содержание предъявляемой с их помощью информации, но и длительность использования СО.

Принцип соответствия. СО выбираются в соответствии с их дидактическими, эргономическими и техническими возможностями, удовлетворяющими учебный процесс на данном этапе. Поскольку СО должны удовлетворять конкретным целям обучения, выбор их следует производить так, чтобы он в наибольшей степени соответствовал этим целям. Одно и то же понятие можно проиллюстрировать по-разному: и рисунком на доске, и слайдом, и фрагментом кинофильма и т.п., однако степень учебной эффективности будет зависеть от того, соответствует или не соответствует данное СО решаемой задаче обучения. Выбор СО с учетом их эргономических свойств сводится к установлению соответствия между выбранным СО и условиями учебной работы обучаемых. Под соответствием понимается такой выбор СО, при котором работа обучаемых проходила бы в комфортных условиях: достаточная освещенность, контрастность, слышимость и т.п. Учесть технические возможности СО — значит выбрать СО с учетом удобства его эксплуатации. Напр., если необходимо сделать выбор между применением слайдов и плакатов — средств, имеющих близкие информативные возможности, предпочтение отдают слайдам. Однако, если занятие проводится в аудитории, не имеющей затемнения, целесообразно применение плакатов.

Принцип информативности. Методика применения СО должна быть такой, чтобы обеспечить максимум обучающей информации при минимуме помех. Следовательно, используя СО, нужно освободить их от «шумового» наложения, т.е. от всего постороннего, являющегося помехой для восприятия осн. информации, предьявляемой с их помощью. Напр., преподаватель располагает несколькими плакатами, но в процессе изложения материала использует лишь отдельные элементы информации из этих плакатов, подавляющий же объем содержащейся в них информации не используется. В этом случае второстепенная информация создает значительные помехи в усвоении основной, как бы теряющей среди той, которая для данного случая не столь важна. Аналогичная ситуация возникает при демонстрации слишком заполненных экранов в КТО.

При рассмотрении П.п.с.о. не акцентируется к.-л. конкретная учебная дисциплина, т.к. по отношению к учебным дисциплинам эти принципы универсальны. Чтобы рассмотренные принципы давали возможность решать соответствующие им задачи, их необходимо сопоставить с конкретным содержанием учебного материала и условиями учебного процесса. Если это условие будет выполнено, преподаватель может ответить на вопросы: как подготовить учебный материал для рационального использования СО в учебном процессе — принцип обусловленности; какое СО выбрать в зависимости от целей обучения (одновременно решается вопрос и о продолжительности использования СО) — принцип необходимости; как конкретизировать выбор СО в соответствии с условиями его приме-

нения — принцип соответствия; как пользоваться СО, чтобы обеспечивать максимум обучающей информации — принцип информативности.

А.Е. Денисов.

ПРИНЯТИЕ РЕШЕНИЙ — выбор наиболее предпочтительных решений. Перед П.р. необходимо получить исходную *информацию* и переработать ее, после чего — проанализировать информацию, полученную в результате принятого решения, и объяснить целесообразность выбранного решения.

Для функционирования процесса П.р. необходимо существование двух подсистем: *базы знаний* (БЗ) по *предметной области*, обеспечивающей распознавание исходных *данных*, и БЗ правил, осуществляющих классификацию результатов анализа в терминах возможных решений. Для обеспечения П.р. необходимо: наиболее точное определение терминов и *понятий*, используемых лицом, принимающим решение (ЛПР); умение четко сформулировать *задачу* и находить *алгоритм* ее оптимального решения; умение находить экспертов и сотрудничать с ними для более быстрого решения задачи. ЛПР должен подходить к П. р. с точки зрения получения наилучшего решения, т.е. должна быть создана некоторая иерархия целей задач в соответствии с критериями, влияющими на выбор решения.

Процесс П.р. является многоэтапным. Рассмотрим его в применении к *обучению*. На первом этапе предьявляется осн. учебная информация, формулируется сущность проблемы (этап семантического анализа проблемы). На втором этапе ставятся *вопросы*, на которые должен ответить *обучаемый*, применяются тесты для определения степени незнания. На третьем этапе предьявляется дополнительная информация, уточняющая постановку вопросов и формулировку проблемы. Далее происходит диалоговая корректировка или уточнение гипотезы в процессе моделирования, проблемная подсказка. На следующем этапе — пошаговое сопровождение решения в режиме *диалога*, оценка правильности результата.

Развитие человеко-машинных методов П.р. связано с проблемой интеллекта и предусматривает построение цепочки *моделей* "потребность — мотив — цель — принятие решения". Обычно строится модель только последнего элемента данной цепочки "цель — принятие решения". На практике ограничиваются рассмотрением простейшего низшего звена, дав ему название "вопрос-ответ". *Ответы* могут выражаться на разных языках с разной степенью обобщения или детализации. Новое *знание* (ответ), отличное от знаний ЛПР, можно найти, генерируя ответы при помощи перебора множеств ответов прецедентов по внешним критериям, априори указанным ЛПР, т. е. указываются некоторые характеристики ответа,

критерии, по которым находится ответ. Понятие "ответ" весьма емко, но все же дискретно.

Одним из распространенных способов проверки правильности свободно конструированных ответов для *обучающих систем* является дескрипторный контроль, т.е. контроль по *ключевым словам*. Описание множества правильных ответов включает перечень дескрипторов (слов или устойчивых словосочетаний), связей между ними и схемы допустимых ответов. Проще всего описываются ответы на вопросы о перечислении предметов, обладающих некоторым свойством, либо об описании системы признаков, служащей для опознания явления или предмета. Ответ считается верным, если совпадает хотя бы одно название. Для управления самостоятельным исправлением неправильного ответа лучше применять наводящие фразы или подсказки, а не ограничиваться простой констатацией ошибок. Желательно предусмотреть возможность многократного ответа на задаваемые вопросы.

А.Ф.Манако.

ПРИБРЕТЕНИЕ ЗНАНИЙ — процесс передачи и преобразования опыта по решению *задач* от некоторого источника *знаний* (экспертов, специальной литературы, *базы данных*, личного опыта *разработчиков систем*) в *решатель*. Знания, которые должны быть получены, могут представлять собой набор специальных фактов, *процедур*, оценок и правил *принятия решений* в определенной *предметной области*. Выделяют три этапа П.з.: 1) обнаружение неполноты *информации* и выявление новых знаний, устраняющих эту неполноту; эксперт и инженер по знаниям в ходе *диалога с автором* разработки выявляют новые знания и формулируют их в некотором удобном виде, напр., на *языке представления знаний* (ЯПЗ); 2) введение в систему новых знаний; осуществляется в ходе диалога: если новые знания выражены на естественном языке, то они должны быть преобразованы в правило, представленное на ЯПЗ; 3) объединение новых знаний с имеющимися; новое правило объединяется с *базой знаний* (БЗ); эта задача почти не автоматизирована в имеющихся системах — эксперт убеждается в правильности правила и его согласованности с другими правилами БЗ в результате решения тестовых задач.

Процессы передачи и преобразования, необходимые для П. з., в решателе можно автоматизировать полностью или частично лишь в некоторых специальных случаях. Поэтому для осуществления связи эксперт — решатель чаще всего требуется третий участник (инженер по знаниям). Способы П.з. в ИОС и ЭОС (см. *Системы, основанные на представлениях знаний*): формирование знаний "вручную"; взаимодействие эксперта и инженера по знаниям; взаимодействие эксперта непосредственно с решателем; приобретение знаний непосредственно

редственно из литературы. При создании систем, основанных на представлениях знаний (ИОС, ИИСС, ЭС, ЭОС), одно из узких мест — извлечение и структуризация *экспертных знаний*. На этом этапе разработки принимают участие эксперт и инженер по знаниям, осн. задачами которого являются извлечение знаний эксперта, формализация знаний в терминах выбранного языка представления, “упаковка” их в соответствующем виде и ввод в систему.

Обычно структуризация и первоначальное извлечение и накопление экспертных знаний выполняются неформально, путем интервьюирования эксперта, однако такой подход имеет ряд недостатков: отсутствие структуры диалога; наличие между экспертом и инженером по знаниям языкового барьера, связанного с применением экспертом профессиональной терминологии; необходимость переработки инженером по знаниям большого объема новой информации; трудоемкость прямой активизации экспертом своих знаний для решения соответствующих проблем. Далекое не всегда эксперту удается адекватно выразить даже те свои знания, которые можно вербализовать. В самом же процессе П.з. и формирования БЗ можно выделить следующие стадии: идентификация проблемы; концептуализация знаний; формализация знаний; представление знаний на ЯПЗ; реализация БЗ; тестирование БЗ. Для автоматизации процесса П.з. используют автоматическое обучение (в этом случае системы обучаются как знаниям о ПО, так и эвристикам для решения задач) и игровые методы. Для формирования БЗ применяются *экспертные игры*, в частности метод извлечения знаний основан на идее диагностической *игры*.

В.В.Колос, С.П.Кудрявцева.

ПРОБЛЕМНАЯ ЗАДАЧА (от греч. *προβλημα* — преграда, трудность, задача) — в широком смысле то же, что и *нерутинная задача*, т.е. такая, к решению которой *решатель* приступает, не обладая *алгоритмом* или *квазиалгоритмом* ее решения. Термин “П.з.” употребляется и в более узком смысле, соответствующем точке зрения на проблему как на способ фиксации диалектического противоречия. В педагогическом плане проблемность (нерутинность) для *обучаемых* решаемых ими задач нужно оценивать отрицательно, если она является следствием недостаточного усвоения тех способов действий, которыми *обучаемые* должны владеть на данном этапе учебного процесса. В остальных случаях ее следует считать нормальным явлением. Часто ее проектируют специально, особенно в системе *проблемного обучения*.

Г.А.Алл.

ПРОБЛЕМНАЯ СРЕДА компьютеризированная — специализированная *учебная среда*, предназначенная для предоставления

обучаемому компьютерных средств создания и моделирования конкретной ситуационной обстановки, проблемы, явления в определенной предметной области. Объединяет средства в данной предметной области, методы, способы и знания с целью обучения и самообучения по решению задач с помощью компьютера. Дает возможность проводить различные действия, способствующие обучению, такие, как использование *текстовых редакторов*, моделирование процессов, получение справок, *принятие решения*, автоматизация графической деятельности по черчению и т.д. Включает следующие компоненты (или часть их, в зависимости от учебно-научного направления): подсистему управления обучением; подсистему моделирования; *систему управления базами данных и базами знаний*; подсистему организации вычислений; подсистему *машинной графики*.

И. Панков.

ПРОБЛЕМНОЕ ОБУЧЕНИЕ — 1) Целостная психолого-педагогическая теория, раскрывающая закономерности организации учебного познания и формирования творческой личности. 2) Психолого-дидактическая система содержания, методов, форм и средств развивающего обучения, обеспечивающая оптимизацию поисковой познавательной активности *обучаемых* в процессе индивидуального и совместного овладения ими *знаниями* и способами действия. Центральным звеном развития идеи П.о. явилось представление о возможности смоделировать в реальном обучении полноценный творческий процесс *обучаемого*, создав систему проблемных ситуаций. Поэтому главное отличие П.о. от традиционного (объяснительно-иллюстративного) заключается в содержании и характере познавательной деятельности *обучаемых*, предусматривающей диалектическое единство учебных проблем, *проблемных задач* и проблемных вопросов в учебном процессе.

Теор. разработки П.о. характеризуются большой вариабельностью. Это во многом обусловлено тем, что понятие "П.о." ученые относят к разным категориям, вкладывая в него различное содержание. Так, одни считают его принципом дидактики, другие — методом обучения, третьи — новым типом учебного процесса, четвертые — системой в организации обучения, пятые — особым подходом к обучению и т.п. Все указанные обоснования сущности П.о. имеют научный смысл, однако не столько сами по себе, сколько в системе единой теор. концепции. Одновременно они свидетельствуют о широких адаптивных возможностях П.о. к внедрению в практику учебно-воспитательного процесса.

Методологическое основание П.о. составляет принцип проблемности, реализация которого затрагивает все осн. звенья учебного процесса: проблемное структурирование содержания образования, проблемный способ его преподавания и проблемный характер

познавательной деятельности обучаемых. Универсальным фактором, объединяющим эти звенья в психолого-педагогическую систему является учебная проблемная ситуация. Она непрерывно участвует в процессе перехода обучаемого от неполного знания к более обобщенному и системному, а ее полный функциональный цикл (возникновение → становление → разрешение → снятие) всегда представляет собой психологически заверченный этап учебного процесса, т.е. творческого усвоения учеником *понятия* или закона. Помимо сущностных признаков (противоречивость, проективность, полифункциональность, динамичность, диалогичность, продуктивность), учебная проблемная ситуация характеризуется рядом закономерностей: законченной последовательностью четырех этапов функционирования и их строгой направленностью на развитие полноценного продуктивного процесса, наполняемостью каждого этапа соответствующим психологическим содержанием (удивление, осмысление, интеллектуальный конфликт и т.п.) и цикличностью в функционировании взаимообусловленной совокупности познавательных противоречий, а также многими другими. Однако стержнем любой учебной ситуации являются внутренние проблемные ситуации обучаемых, возникновение которых основывается на познавательных смысловых противоречиях, осмысливаемых обучаемым в ходе непрерывного взаимодействия с объектом познания и субъектом преподавания. Эти противоречия находят конкретное выражение в форме личностных проблем, задач и вопросов, являющихся не «компонентами» познавательного процесса, а его функциональными образованиями. Причем решающее значение в организации П.о. имеет качество поисковой познавательной активности обучаемых. Уже на этапе возникновения субъективной потребности в новых знаниях эта активность выступает осн. звеном, из которого впоследствии развиваются познавательная *мотивация* и духовные интересы личности; она предопределяет продуктивность познавательного процесса и темп умственного развития обучаемого. Следовательно, в системе П.о. главными являются условия: возникновение познавательной потребности у каждого обучаемого, формирование познавательной активности, актуализация опорных знаний и развитие познавательной мотивации, осознание, принятие и решение личностной проблемы или задачи, инициация продуктивных процессов мышления, приводящая к обнаружению субъективно новых, обобщенных знаний.

В зависимости от особенностей учебного предмета и уровня интеллектуального развития обучаемых в дидактике различают четыре уровня П.о.: обычной, полусамостоятельной, самостоятельной и творческой активности. В педагогической психологии существует иное распределение П.о. на уровни, в основу которого положены закономерности решения проблемных задач, охватывающие как уровень актуального развития обучаемого, так и зону его ближайшего

развития. Исходя из этих закономерностей, система проблемно-поисковых методов П.о. должна включать четыре группы методов: демонстрационное решение проблем учителем (диалогическое проблемное изложение, демонстрационный эксперимент и др.), самостоятельная поисковая деятельность обучаемого (исследование, решение практической проблемы), коллективная поисковая деятельность обучаемых (учебный диспут, дискуссия), совместная поисковая деятельность преподавателя и обучаемых (эвристическая беседа, деловая игра). П.о. реализуется также при помощи системы учебных средств: *познавательных задач*, проблемных вопросов, проблемно построенного содержания учебника и пр. При этом одним из ведущих условий эффективности внедрения уровней, методов и средств П.о. является высокое профессиональное мастерство преподавателя и, в частности, его умение вести равноправный *проблемный диалог* с обучаемыми.

А.В.Фурман.

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК — *алгоритмический язык*, предоставляющий средства для решения прикладных задач в конкретной проблемной области. Основу П.-о.я. составляют проблемно-ориентированные конструкции, формализующие знания в данной предметной области. По способу представления и обработки данных П.-о.я. делятся на *императивные языки*, *функциональные языки* и *декларативные языки*, по технологической направленности — на *языки спецификаций*, *языки проектирования* (как правило, создаются при разработке больших программных систем) и *языки программирования*.

В.А.Третьяк.

ПРОБЛЕМНЫЙ ДИАЛОГ — особый тип *диалога*, характеризующийся ситуативно ограниченными интеллектуально-волевыми возможностями одного или обоих партнеров в преодолении обнаруженной проблемы и потому требующий от них познавательной и речевой активности для разрешения возникшей проблемной ситуации.

Между проблемностью и диалогичностью творческих процессов существует тесная взаимосвязь. П.д. так же, как проблема и мыслительная *задача*, берет свое начало в проблемной ситуации. Но если возникновение и решение проблемных ситуаций характеризует содержательно-процессуальную сторону развития продуктивного мышления, то П.д. — формально-динамическую, обуславливающую возрастание качества мыслительного процесса и его темпа. Показателями этого являются конкретные результаты совместного поиска как субъективное открытие неизвестного знания, а также желание партнеров решать проблемные ситуации повышенной трудности. Большая ситуативность и неполная заданность композиции П.д.

фактически предопределяют постоянное наличие проблемности при выполнении партнерами задач. Центральное звено этой проблемности составляют познавательные-смысловые противоречия, являющиеся источником активности, интереса и развития обоих партнеров. Продуктивность мышления здесь выражается через идеи, предположения, гипотезы, признаки *понятий*, доказательства и т.п. При рассмотрении диалога *обучаемого* с компьютером следует отметить, что для компьютера П.д. не возникает, т.к. ему не свойственна проблемность в ее субъективно-личностном преломлении (он работает в режиме спроектированной *обучающей программы*).

Системообразующим фактором возникновения диалогических взаимоотношений и одновременно развития продуктивной познавательной активности партнеров являются проблемно-диалогические ситуации. В *обучении* они детерминируют проблемный способ овладения обучаемыми научными знаниями, определяют уровень их поисковой активности, обуславливают возникновение новых психических образований: мотивов, целей, приемов мышления, убеждений и пр. Важно, что в процессе П.д. осуществляется не просто овладение истиной в процессе разрешения проблемно-диалогической ситуации, но и выявляется ее многогранность, поскольку каждый партнер имеет свою собственную точку зрения, специфическое видение проблемы, что и обуславливает в конечном счете систематическое столкновение известного и неизвестного, привычного и непривычного знания. Существенной особенностью П.д. является то, что в нем на первый план выступает продуктивный *внутренний диалог*. Внешний диалог стимулирует и дополняет внутренний, находится с ним в диалектическом единстве, т.е. ускоряет появление новых психических образований. Поэтому П.д. — основа развития творческого мышления, феномен культуры и одновременно наиболее действенный инструмент развивающего и воспитывающего воздействия на формирующуюся личность.

В дидактике П.д. — один из главных методов *проблемного обучения*. Его осн. содержание составляет взаимообусловленная активность обучаемых и *преподавателя* в поиске объективно истинного знания. При этом учебная проблемность активизирует процессы продуктивного понимания, а диалогичность реализует прогрессивную направленность их развития, что в совокупности обеспечивает действенность приобретаемых обучаемыми знаний. Одновременно происходит овладение субъективно новыми приемами познавательной деятельности, средствами группового мышления. И чем больше обучаемые включены в преобразовательный поиск теор. содержания понятий в процессе диалогических взаимоотношений, тем более обобщенными и системными знаниями овладевают и тем интенсивнее формируется их способность к сотворчеству.

А.В. Фурман.

ПРОГНОЗИРОВАНИЕ В ОБУЧЕНИИ — составление прогноза деятельности *обучаемого* в процессе *обучения*. Направлено на повышение качества подготовки специалистов путем введения оперативной коррекции в *алгоритмы управления учением*, напр., в алгоритм реализации учебных процедур с помощью средств адаптивных *автоматизированных обучающих систем* (АОС), а также формирования рекомендаций по оптимизации организационного обеспечения учебного процесса. Прогноз вузовского учебного процесса основывается на трех группах *данных*, отражающих довузовский период обучения, период обучения в вузе, данные о практической деятельности специалистов после окончания вуза. Реализация алгоритмов прогноза результатов процесса обучения и параметров трудовой деятельности будущих специалистов дает возможность: организовать профотбор среди абитуриентов; подобрать оптимальный контингент студентов в микроколлективе; определить степень успешного выполнения учебных заданий; прогнозировать успеваемость обучаемых по отдельным предметам; выдавать рекомендации каждому обучаемому о мерах, необходимых для повышения оценки по данному предмету; выдавать рекомендации по выбору сферы деятельности в рамках профессии в зависимости от индивидуальных особенностей каждого обучаемого; оптимизировать учебный процесс на основе прогноза показателей научной, общественной и учебной работы студента; индивидуализировать обучение; производить сравнительную оценку качества подготовки студентов; производить оценку эффективности используемых АОС; выдавать рекомендации по оптимальному распределению выпускников на основе максимума вероятности успешного выполнения ими данного вида практической деятельности.

В.Н.Сороко.

ПРОГРАММА — набор предписаний на *алгоритмическом языке*, реализующий *алгоритм решения задачи* на компьютере. П. принято разделять на вызывающие и вызываемые (*подпрограммы*). Вызывающей наз. П., в которой используется предписание для вызова другой программы П., на которую не существует предписаний вызова в других П., наз. *главной*, или *основной*. На содержательном уровне П. различаются по типу *предметной области*, в которой они решают определенные *пользователем задачи*, напр., П. управления технологическими процессами, П. диагностики (в медицине, машиностроении и др.), *программы учебного назначения* и др.

В.А.Третьяк.

ПРОГРАММА УЧЕБНОГО НАЗНАЧЕНИЯ (ПУН) — программа для ЭВМ, реализующая педагогическую функцию *учения* или *обучения* при взаимодействии с учащимся. Функции учения соответ-

стует класс ПУН, наз. *учебными средами*, функции обучения — класс, наз. *обучающими программами*.

К настоящему времени сложилась эмпирическая классификация элементарных классов ПУН. Так, класс учебных материалов на машинных носителях информации представляет собой *тексты* и *рисунки*, которые при необходимости можно вывести на печать. С помощью демонстрационных программ обучаемому предъявляется текстовая, графическая и звуковая информация, но они не контролируют процесс ее усвоения. Информационно-справочные программы служат для хранения фактографической информации по различным отраслям знаний и предъявления ее по запросу обучаемого. Вопросно-ответные программы также предназначены для информационного обеспечения обучаемого, но в отличие от предыдущего класса предполагают наличие естественного языкового интерфейса с обучаемым и механизма вывода новых фактов из имеющихся в наличии. Измеряюще-контролирующие программы осуществляют обучение способом управления нештатными внешними устройствами ЭВМ (датчиками, станками, роботами и т.п.), имитационные программы служат для изучения основных структурных и функциональных характеристик некоторого реального явления или процесса путем экспериментов над его моделью, представленной ограниченным числом параметров. Моделирующие программы предоставляют в распоряжение обучаемого осн. конструктивные элементы и операции над ними для представления некоторого реального или воображаемого явления или процесса. Учебно-игровые программы — моделирующие программы, игровая форма которых способствует развитию дополнительной мотивации к освоению изучаемого предмета. Инструментальные программы позволяют учащимся производить вычисления, обработку текстов, графических изображений и т. п. Языки программирования, разработанные для целей обучения (ЛОГО, Паскаль, Бейсик), представляют собой важный подкласс инструментальных программ. Решатели служат для автоматического решения задач в заданной предметной области, условия и/или данные для которых задает учащийся. Диагностические программы предназначены для диагностирования, проверки и оценки знаний, умений и навыков обучаемого, обучающе-контролирующие программы — для управления процессом учения, основываясь на текущем уровне знаний обучаемого и целей обучения (предполагают усвоение новой информации), тренирующие программы — для закрепления пройденного учебного материала и выработки у обучаемого необходимых умений и навыков. Экспертные обучающие системы позволяют гибко управлять познавательной деятельностью обучаемого на основе моделирования его состояния знаний, знаний экспертов об изучаемом предмете и методах обучения (экспертных знаний в области педагогики и психологии). Управляющие программы служат для управления познавательной деятельностью одного обучаемого или

группы обучаемых в классе и/или учебном заведении (кроме предписаний по работе с компьютером, такие программы содержат отсылки обучаемого для изучения раздела книги, просмотра видеofilmа, проведения натурального эксперимента, дискуссии с другими обучаемыми и т.п.).

В.А.Петрушин.

ПРОГРАММИРОВАНИЕ — 1) Разработка программ решения различных задач на электронных вычислительных машинах. В широком смысле под П. подразумевается весь процесс создания программы (анализ требований, разработка проекта и реализация программы); в узком смысле — составление и отладка программы. При изучении П. широко используют компьютерные технологии; разработаны программы учебного назначения различных классов, применяемые для изучения как общих понятий и методов П., так и конкретных инструментальных средств (прежде всего, языков программирования). Некоторые распространенные языки программирования (напр., Бейсик, Паскаль), основанные на самых различных концепциях и имеющие широкие области применения, были разработаны именно как учебные.

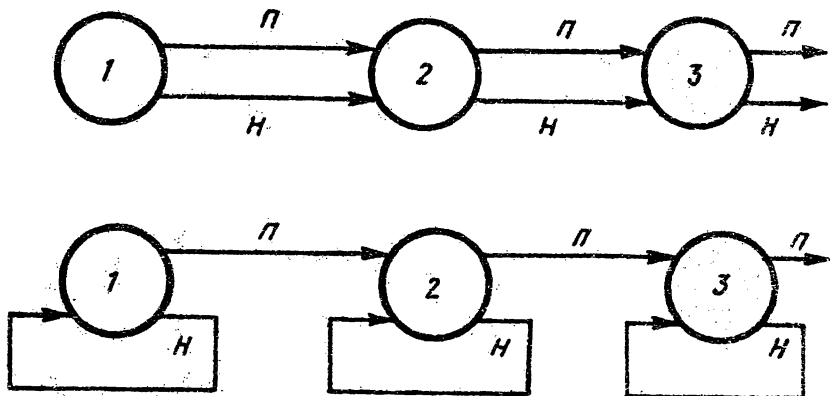
2) Наука, занимающаяся разработкой методов и средств формализации процессов обработки информации, представления их в виде алгоритмов и получения программ для ЭВМ. Осн. направления — семантика языков программирования, доказательство правильности программ, преобразования программ, создание спецификаций программ, методология П. и др. Использует аппарат матем. логики, алгебры, теории алгоритмов.

3) Матем. П. — раздел прикладной математики, занимающийся изучением задач отыскания экстремумов функций на заданных множествах и разработкой методов решения этих задач. В зависимости от свойств оптимизируемых функций и множеств допустимых значений выделяют линейное программирование, квадратичное программирование, выпуклое программирование, целочисленное программирование и др. разделы матем. программирования.

В.А.Бардадым.

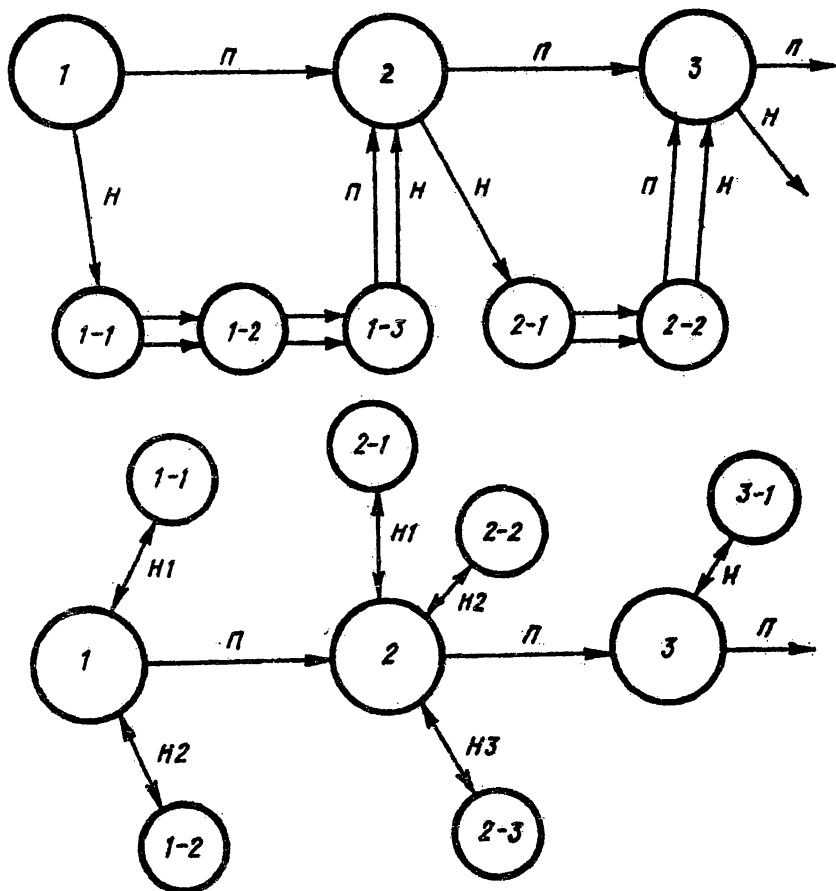
ПРОГРАММИРОВАННОЕ ОБУЧЕНИЕ — тип обучения, осуществляющийся в соответствии с обучающей программой, задающей не только знания, умения и навыки, которыми должны овладеть обучаемые, но и способ организации их учебной деятельности. Реализации этого существенного свойства П.о. служат его осн. внешние характеристики: а) расчленение учебного материала на отдельные порции (дозы); б) интенсивный обмен информацией между обучающей программой и обучаемым, предусматривающий, в частности, его достаточно частые ответы на задаваемые вопросы и информирование

его о степени правильности этих ответов. Эти характеристики обеспечиваются с помощью учебных книг, специальным образом составленных и оформленных (так наз. программированных пособий), технических устройств (от простейших механических и электромеханических до компьютеров), а также комбинаций этих средств. В П.о. применяются как линейные (рис.), так и разветвленные (рис.на с.422) обучающие программы. В первом случае последовательность



проходимых порций одна и та же для всех обучаемых, во втором — зависит от ответов, которые дает обучаемый. При этом в *адаптивных обучающих программах* учитывается не только последний ответ, но и некоторая история обучения.

Программирование создает возможности для повышения эффективности обучения благодаря тщательному продумыванию содержания и логической структуры учебного материала при составлении обучающей программы, воплощению в ней удачных методических находок, обеспечению активной самостоятельной работы обучаемых, повышению степени *индивидуализации обучения* (в частности при использовании разветвленных, особенно адаптивных, программ). Эти возможности реализуются только при высоком качестве обучающих программ, их психологической и методической обоснованности. П.о. часто сочетается с родственным ему направлением совершенствования учебного процесса — стандартизованным (тестовым) *контролем знаний* и умений, осуществляющимся с помощью контролирующих программ — наборов заданий (вопросов), для которых заранее выделены и оценены возможные ответы (либо их типы). Обучающие и контролирующие программы нередко реализуются посредством одних и тех же *технических средств обучения*.



П.о. (как сочетаемое с традиционным, осуществляемым преподавателем, так и используемое для самобразования) получило распространение начиная с 50-х гг. 20 в., вначале в США, а затем и в других странах. Во многих случаях оно оказалось весьма успешным. Это касается, в частности, обучения языкам программирования и решения задач с помощью компьютеров. Вместе с тем не оправдались предположения о том, что с помощью П.о. удастся обеспечить качественный скачок в области народного образования. Они были ошибочны прежде всего методологически, поскольку игнорировали системный характер процессов развития образования, ведущую роль социально-экономических факторов. Весьма скромные успехи П.о. объясняются также примитивностью использовавшейся обычно технической базы, не позволявшей, в

частности, контролировать свободно конструируемые ответы обучаемых. Ориентация на такую базу изначально ограничивала дидактический потенциал П.о., исключая, напр., *диалог*, управляемый обучаемым. Подобные ограничения устраняются в *компьютерной технологии обучения*, которая, реализуя П.о. (и стандартизованный контроль), обладает также рядом других возможностей. Успешному ее развитию будет способствовать полноценное использование опыта П.о., а также результатов исследований (психолого-педагогических, психолого-кибернетических, методических), выполненных в его рамках.

Г.А.Балл.

ПРОГРАММИРОВАННЫЙ УЧЕБНИК — учебник, разбитый с учетом элементов *обучающей программы* на отдельные небольшие части или разделы, усвоение которых необходимо для дальнейшего чтения. Изучение каждого раздела сопровождается примерами и *вопросами*, требующими правильных *ответов*, благодаря чему непрерывно осуществляются самоконтроль и самопроверка усвоения. Отсутствие рядом с обучаемым *преподавателя* компенсируется самоконтролем. П.у. сочетает в себе различные методы активизации познавательной деятельности *обучаемого*. Определение параметров П.у. (размера, дозы учебного материала, шагов и их количества и т.п.) возможно лишь в процессе длительных педагогических экспериментов. Осн. задача П.у. — дать учебную *информацию* в лаконичной форме с определенной логической последовательностью, в минимум времени сообщить максимум полезных сведений с обязательной оптимальной управляемостью и самоуправляемостью процессом *обучения*. По объему П.у. не должен превышать обычный (эксперименты показывают, что объем П.у. целесообразно уменьшить на 10—15%). Поскольку П.у. может исключить из процесса обучения преподавателя как непосредственного источника информации, оставляя за ним функции консультанта и руководителя самостоятельной работы обучаемого, особое внимание следует обратить на контрольные задания (вопросы) с учетом схемы программирования — линейной, разветвленной, комбинированной (см. *Программированное обучение*).

Главное отличие П.у. от традиционного заключается в том, что наряду с фактическим материалом П.у. должен иметь дидактический и справочный материал, обеспечивающий систематическую обратную связь (см. *Обратная связь в обучении*) от обучаемого к П.у. и возможность управления процессом усвоения новых *знаний* обучаемым в зависимости от результатов работы с книгой. Архитектоника П.у. (строение произведения, соотношение частей, глав, доз и т.п.), композиция параграфов, карточек контроля должны предусматривать диалоговую дидактическую *деловую игру*: П.у. (т.е. автор) — обучаемый или П.у. — обучаемый вместе с преподавателем (при

групповом обучении). При этом следует обратить внимание на рациональное использование *текста*, рисунков, таблиц, выполнение самостоятельных заданий. В П.у. должен быть представлен такой информационный и обучающий материал, чтобы он вызывал интерес к диалоговой дидактической игре.

Объем дозы (шага) информации в П.у., их количество для систематической обратной связи, обеспечивающей управляемость процессом усвоения материала, может быть от 1/3 страницы до полной страницы книжного листа. Количество доз на один печатный лист — от 30 до 40, в т.ч. 4 — 5 контрольных карточек. Вопросы, которыми заканчивается каждая доза информации, должны содержать элементы диалога обучаемого с П. у. и относиться не только к тексту, но и к чертежам, конкретным деталям и узлам на рисунках. Следует избегать постановки вопросов, на которые имеются прямые ответы в программных дозах; вопросы должны активизировать работу обучаемых, контролировать и стимулировать запоминание, прочное усвоение и обобщение изучаемого материала (с обязательной обратной связью). Расположение карточек контроля знаний в П.у., их количество, форма и включение в них иллюстраций, формул и т.п. играет важную роль в управлении процессом обучения либо самостоятельной работе с книгой. Рекомендуется контрольная карточка с пятью вопросами с конструируемыми ответами, чем стимулируется работа обучаемого по пяти элементам знаний. В контрольных вопросах карточек используется, как правило, до 5 альтернатив. Методически обоснованной постановкой вопросов и ответов (альтернатив) можно не только активизировать учебный процесс, но и стимулировать у обучаемых направленность мыслительной деятельности, делать самостоятельные сравнения, выводы и т.п. В карточках необходимо предусматривать: диалоговые вопросы для усвоения терминологии; вопросы и задания умственных действий (эвристические) для осмысления и сознательного усвоения, развития мышления и формирования умений и навыков; вопросы, рассматривающие детали, узлы, монтаж и демонтаж устройств на лабораторно-практических занятиях, а также при изучении теор. материала; задания на повторение пройденного материала, формул осн. законов и явлений; выдачу типовых задач и примеров. Возможны и другие варианты программирования ответов. По сравнению с обычным учебником П.у. требует большего количества иллюстративного материала, участвующего в программировании мыслительной деятельности обучаемого. Иллюстрации помогают формировать понятие о той или иной машине, детали, если они поданы в такой последовательности: рисунок (или фотография), чертеж (или кинематическая схема). В П.у. следует использовать иллюстративный материал, обеспечивающий практическую наглядность в обучении. Для него необходимо применять многоцветную печать, позволяющую полнее программировать чертежи, схемы и т.п., оптимально активи-

зировать мысль обучаемого. Нужно чаще использовать символику, т.е. совокупность знаков, обращающих внимание обучаемого на определение места содержания текста. Исключительное значение имеет компоновка книжных полос (текст, формулы, иллюстрации). От этого во многом зависят качество восприятия, скорость запоминания и усвоения учебной информации.

При внедрении *автоматизированных обучающих систем* (АОС) П.у. повышают продуктивность и полезную отдачу при работе. Их должны создавать высококвалифицированные авторы на основе пробных глав, применяя длительные педагогические эксперименты, — таково одно из главных требований к П.у. для использования в *компьютерной технологии обучения*. Определенная проблема при внедрении компьютеров в учебный процесс — ввод в машину с помощью *клавиатуры*. Стандартная клавиатура серийных компьютеров не имеет специализированных символов, необходимых для учебного процесса. Это затрудняет диалог обучаемого с компьютером. Длительная работа обучаемых у экрана *дисплея* утомляет их, что снижает активность познавательной деятельности и ухудшает восприятие учебной информации. В связи с этим в качестве осн. *банка знаний* целесообразно использовать П.у., позволяющий дифференцировать компьютерный диалог в зависимости от успешного выполнения заданий обучаемым. На 60—70% сокращается время сеанса (контакта общения) индивидуальной работы обучаемого с компьютером. С П.у. обучаемый работает дома, а с компьютером — на уроке в учебном заведении. Машина выдает задание обучаемому и обеспечивает контроль усвоения знаний с элементами тренажа и *учения*.

Примерный педагогический сценарий работы обучаемого с использованием персональных компьютеров по курсу “Детали машин”: самоподготовка обучаемых к очередному занятию по П. у. от 2 до 4 часов; занятие в дисплейном классе в режиме диалога с АОС (диалог обучаемый — компьютер ведется на естественном языке посредством клавиатуры и экрана не более одного академического часа). Этапы занятия у экрана дисплея: регистрация обучаемых; контроль выполнения предыдущего задания (предварительно выясняется время, затраченное обучаемыми на изучение материала, обращение их за консультацией к преподавателю, проверка ответов на контрольные карточки П.у.); проверка усвоения учебного материала темы, для чего компьютер предлагает последовательно ответить на 5 вопросов, включающих теор. материал, решение примера, использование теории на практике и т.д. с разветвленной схемой программирования вариантов ответов обучаемого; итоговая оценка знаний по проработанному заданию, которая сообщается компьютером по пятибалльной системе (при неудовлетворительной оценке знаний по проработанному заданию компьютер дает указание изучить необходимые разделы по П.у. и повторить работу с

компьютером); выдача компьютером задания на самоподготовку с П.у. по следующей теме; опрос обучаемых по ведению занятия с компьютером (уточняется ведение конспекта, утомляемость при работе, затраты времени на самоподготовку и т.п.). Аналогично может использоваться П.у. в рамках КТО по любой учебной дисциплине.

А.М.Довгялло, И.И.Мархель, В.И.Отенко.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ — совокупность программ, позволяющих использовать ЭВМ для требуемого применения (решения задач, управления процессом обучения и т.д.). Различают П.о. базовое (системное), неразрывно связанное с техническими средствами ЭВМ, и проблемно-ориентированное (специализированное), позволяющее использовать ЭВМ для решения прикладных задач. Базовое П.о., поставляемое вместе с ЭВМ, включает операционные системы, системы программирования для универсальных языков, системы управления базами данных, интерфейсные программы (графические редакторы, текстовые редакторы, электронные таблицы, интерпретаторы команд пользователя и сервисные программы для работы с файлами). Проблемно-ориентированное П.о. представляет собой совокупность программ для решения определенного комплекса задач; оно не поставляется вместе с ЭВМ, а приобретает отдельно в виде программных систем, библиотек стандартных программ (подпрограмм) и/или пакетов прикладных программ.

В.А.Петрушин.

ПРОГРАММНЫЕ ТРЕНАЖЁРЫ — то же, что и *тренирующие программы*.

ПРОДУКТЫ ДЕЯТЕЛЬНОСТИ — результат деятельности, дифференцирующийся на прямой и побочный. Прямой П.д. является превращенной формой предмета деятельности; в нем можно выделить также запланированный (желаемый) и неудавшийся П.д. Побочные П.д. различаются по своей отнесенности к тому или иному структурному моменту деятельности. При анализе обучающей и учебной деятельности особое значение имеет различие фактического результата, прямого и побочного П.д. Прямой П.д. — это результат, соответствующий осознанной цели субъекта, решающего задачу. Побочный П.д. — та добавка к прямому П.д., которая в совокупности с ним образует фактический результат деятельности. Деятельность наз. учебной в том случае, если ее прямому продукту соответствуют учебные приобретения, т.е. достижение определенных учебных целей.

Е.Д.Маргулис.

ПРОДУКЦИОННЫЕ МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ — методы *процедурного представления знаний*, основанные на использовании некоторого множества правил продукций, где каждое правило — преобразование, задающее пару: условие применимости (предусловие) → действие. Предусловие описывает ситуацию как перечисление условий, необходимых для существования данной ситуации. Если предусловие истинно, то выполняется действие продукции. В *программе* продукция выступает как тройка: имя продукции, условие применимости, *оператор*. Различают прямые продукции, описывающие процесс перехода от исходного состояния к целевому, и обратные, с помощью которых происходит разбивка цели на подцели (осуществляется обратный процесс). Правила продукций являются естественным средством *представления знаний* о решении *задач*, т.к. непосредственно воспроизводят отдельные шаги умозаключений, выполняемых при решении этих задач человеком-экспертом. В то же время при возрастании числа правил отслеживать их взаимное влияние в *базе знаний* (БЗ) становится затруднительно, резко возрастает время обработки множества правил. Использование продукций для представления знаний — конкретный метод организации БЗ в виде трех подмножеств: фактических, процедурных и управляющих знаний. Продукционные системы содержат множество правил, *базу данных* и вычислительную процедуру. Здесь главная проблема — порядок выбора правил.

Осн. достоинства П.м.п.з.: универсальность метода *программирования*; естественная модульность организации знаний; независимость каждой продукции от содержания др. продукций, что обеспечивает простоту и легкость спецификации, модификации и расширения продукционных знаний. П.м.п.з. перспективны для реализации их на параллельных компьютерах. Их можно использовать при наличии определенных условий (напр., описываемая *предметная область* должна допускать определенный уровень формализации, число взаимодействий между выделенными базовыми *понятиями* должно иметь верхний практический предел, не должно быть конфликтующих подцелей и т.д.). В противном случае необходимо (наряду с продукциями) вводить новое представление, задающее структуру на множестве правил.

П.м.п.з. в *автоматизированных обучающих системах* целесообразно применять при моделировании отдельных этапов процесса *обучения*. Напр., в системе САКИО эти методы используются при выборе оптимального режима обучения в зависимости от состояния *модели обучаемого* (режим обучения определяется такими характеристиками *обучающих воздействий*, как объем и форма подачи учебного материала, вид *контроля знаний*, стилистические параметры учебных *текстов* и т.д.), а также при моделировании ситуации урока для выбора очередного задания в зависимости от оценки предыдущего.

О.Н.Золотопуп, В.В.Колос.

ПРОЕКТИРОВАНИЕ — вид деятельности, направленной на создание идеальных *моделей* объектов, которые предположительно по некоторым параметрам превосходят реально существующие. Различают процесс и продукт П. Процесс П. носит непрерывный и итерационный характер, включает акты *принятия решения* и многократное моделирование объекта П. (на основе определенных представлений об искомом объекте, научных и инженерно-технических *знаний*, воображения, интуиции и т.д.). Продукт П. — это идеальный объект особого рода, именуемый проектом; представляет собой практическую модель искомого объекта. По мнению укр. психолога Е.И.Машбица, особенности проекта заключаются в том, что: а) он описывает еще не существующие в реальности объекты (т.е. любое описание имеющегося объекта не может рассматриваться как проект); б) в принципе, может быть реализован с помощью имеющихся или доступных в перспективе средств; в) должен быть нормативным, воспроизводимым и обеспечивать доступный уровень выполнения соответствующих операций. Будучи важным этапом в создании объекта, обладающего определенной общественной значимостью, проект вместе с тем имеет и самостоятельную ценность, отделяясь от деятельности по его реализации. Это обусловлено тем, что ряд проектов, итеративным образом восходящих от абстрактного к конкретному, позволяют совершенствовать видение объекта, делают его более адекватным, уточняют проблемы (их постановку и средства решения), которые необходимо решить при создании следующих проектов и их практической реализации. См. также *Проектирование психолого-педагогическое*.

Е.Д.Маргулис.

ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ДИАЛОГА — *проектирование*, направленное на разработку проекта интерактивного графического *интерфейса* пользователя с компьютером. Одним из осн. положений методологии П.г.д. является проведение аналогии между диалогом *пользователя* с компьютером и человеческим общением. При этом “словами” графического языка являются изображения объектов и действия (напр., нажатие функциональных клавиш на *клавиатуре*, выбор элементов картинки селектором и т.д.). Такой подход позволяет применять в процессе П.г.д. *понятия* и навыки из практики человеческого общения и теории языка. В частности, для языка графического *диалога* сохраняют свою значимость многие атрибуты языка общения человека с человеком. К их числу прежде всего относятся: естественность для пользователя (отсутствие чрезмерной ориентации на компьютер); эффективность (т.е. возможность давать компьютеру *команды* оперативно и кратко); полнота, понимаемая как возможность выразить любую идею в данной области исследования; естественность *грамматики*, подразу-

мевающая небольшое число простых и легких для изучения правил; расширяемость как метод превращения полного языка в эффективный путем предоставления средств для определения новых выражений, являющихся комбинацией существующих. Более подробное рассмотрение языка графического диалога позволяет констатировать, что в действительности в нем можно выделить два языка. На одном из них пользователь формулирует команды и данные для компьютера, а на другом компьютер отвечает пользователю. Первый язык выражается в действиях пользователя с разнообразными диалоговыми устройствами, в то время как второй — графически, посредством точек, отрезков, цепочек литер, цветов, сплошных областей, из которых формируются изображения и сообщения. Несмотря на существенные отличия в способах выражения, оба языка состоят из таких осн. частей: концепции, семантики, синтаксиса, лексики. В соответствии с этим определяются осн. этапы П.г. д.: концептуальное, семантическое, синтаксическое и лексическое проектирование.

На этапе концептуального проектирования объектами проектирования являются ключевые прикладные понятия, входящие в пользовательскую модель. Цель этого этапа заключается в определении объектов, взаимосвязей между объектами (классами объектов) и операциях над объектами (классами объектов). Иными словами, концептуальное проектирование заключается в построении пользовательской модели графического диалога. При семантическом проектировании определяются функциональные характеристики языка. Фиксируются необходимая информация для операций, возможные семантические ошибки и способы их устранения, результаты каждой операции. На этапе синтаксического проектирования разрабатывается последовательность, в которой должны появляться элементарные единицы языка. Для входного языка — это грамматика; в выходном языке понятие последовательности расширяется включением пространственных и временных характеристик. В синтаксис выходного языка включаются правила для построения двух- и трехмерных изображений, а также правила, описывающие любые изменения формы во времени. Этап лексического проектирования связан с разработкой способов формирования знаков входного и выходного языков на основе имеющихся аппаратных примитивов или лексем. Цель его — привязка аппаратных возможностей к аппаратно-независимым знакам входного и выходного языков. При проектировании сверху вниз привязка к физическим представлениям или реализациям откладывается на последний этап проектирования.

При П.г.д. существенным моментом является создание протокола диалога, определяющего, как протекает во времени диалог пользователя с компьютером. Протокол диалога описывает, как выполняются команды пользователя: сразу с предъявлением результатов или

с предварительным группированием в пакет и последующим выполнением. В протокол входят также времена отклика и реакция компьютера на ввод пользователем лексических, синтаксических и семантических конструкций.

Одна из предпосылок достижения успешных результатов при П.г.д. — учет человеческих (эргономических) факторов интерактивной системы. Накопленный к настоящему времени опыт разработки и теор. исследований в области П.г.д. позволяет сформулировать принципы, с помощью которых можно создать графический интерфейс с хорошими эргономическими характеристиками: обеспечение обратной связи; помощь пользователю при изучении системы; возможность возвращения в предыдущее состояние и исправления ошибок; управление временем отклика; соблюдение цельности системы (отсутствие исключений и особых условий); структуризация изображения; минимизация объема запоминаемой пользователем информации при работе с системой.

В.И.Отенко.

ПРОЕКТИРОВАНИЕ МОБИЛЬНЫХ АВТОМАТИЗИРОВАННЫХ ОБУЧАЮЩИХ СИСТЕМ — 1) Совокупность процедур, ориентированных на решение задачи создания проекта мобильной автоматизированной обучающей системы (АОС). 2) Этап жизненного цикла мобильной АОС, на котором происходит разработка ее проекта. Задача П.м. АОС включает следующие осн. подзадачи: выбор класса операционных обстановок, в рамках которого будет осуществляться перенос разрабатываемой АОС; определение структуры АОС; разработка пользовательского интерфейса; разработка структур данных и алгоритмов функционирования компонентов АОС; разработка интерфейсов между компонентами АОС; определение операционно-зависимых и операционно-независимых компонентов; разработка плана переноса АОС; разработка плана тестирования системы; проверка соответствия проектных решений спецификации АОС (если таковая создавалась). Главной методологической принцип П.м. АОС состоит в решении указанных подзадач относительно выбранного класса операционных обстановок, тогда как традиционное проектирование осуществляется применительно к особенностям одной операционной обстановки, образуемой конкретным компьютером и его программным обеспечением. Результатом проектирования является проект мобильной АОС, представляющий собой описание процесса обработки информации системой относительно класса операционных обстановок, в рамках которого будет осуществляться перенос данной системы.

Основу структурной организации мобильных АОС составляет модуль, который в общем случае может содержать определение нескольких *процедур* и ассоциированных с ними структур данных. Это позволяет использовать модуль не только как средство процедурной абстракции, но и как средство описания абстракции данных. Определяемые спецификацией АОС *понятия* на этапе проектирования представляются посредством модулей. Совокупность модулей АОС разбивается на два подкласса. Один подкласс составляют операционно-зависимые модули, т.е. модули, реализация которых может потребовать использования специфических функциональных свойств конкретной операционной обстановки выбранного класса обстановок. К другому подклассу относятся операционно-независимые модули, реализация которых не требует использования особых свойств к.-л. операционной обстановки из данного класса, достаточно использовать общие свойства, присущие всем операционным обстановкам этого класса. Операционно-зависимые модули предназначены для инкапсуляции (сокрытия) тех или иных особенностей операционных обстановок.

Основу проекта мобильной АОС составляют описания модульного и операционно-зависимого расслоений, в которые включаются: описание мобильного пользовательского интерфейса; описание межмодульных интерфейсов; описание каждого модуля, включающее описание структур данных и ассоциированных с ними процедур обработки, а также описание операционной зависимости. Существенными компонентами проекта мобильной АОС являются план переноса и план тестирования. План переноса предназначен для достижения максимально возможного уровня эффективности переноса и определяет его процедуру, т.е. последовательность переноса и инструментальные средства его осуществления.

План тестирования используется для проверки соответствия реализаций мобильной АОС исходным требованиям, спецификации и проекту. Инструментальное средство создания проекта мобильных АОС — язык мобильного проектирования.

Д.Н.Батанов, В.И.Отенко.

ПРОЕКТИРОВАНИЕ ПСИХОЛОГО-ПЕДАГОГИЧЕСКОЕ — вид проектирования, осуществляемый на основе психологических закономерностей обучения и воспитания (принципов педагогической психологии). В качестве предмета исследования выделено и разрабатывается укр. психологом Е.И.Машбицем. Предполагается, что в процессе П.п.-п. все элементы системы обучения анализируются во взаимосвязи и проектируется процесс обучения в целом. При компьютерном обучении эти условия приводят к

проектированию обеих подсистем проектирования во взаимосвязи; поэтому все компоненты *обучающей системы* необходимо анализировать в рамках указанных видов деятельности. Проектирование *обучающих программ* рассматривается как многоуровневый процесс, в котором выделяются уровни: концептуальный, технологический, операциональный и уровень реализации (систематизация Е.И.Машбица).

На концептуальном уровне должны быть описаны те концептуальные представления, которые кладутся в основу при анализе обучения (напр., при рассмотрении его как совокупности двух подсистем — обучения и учения), анализируются психологические механизмы и *принципы обучения*, общие принципы построения обучающих программ, способа управления *учебной деятельностью* (построение *обучающих воздействий*, поля самостоятельности, меры помощи и т.д.).

На технологическом уровне способ управления должен быть описан на микроуровне, т.е. конкретизируются требования к содержательной и формальной стороне обучения, к системе умственных действий обучаемых, к полю самостоятельности, к соотношению между явным и неявным управлением и т.д.

На операциональном уровне процесс обучения должен быть описан как решение *задач обучения*. Здесь прежде всего необходимо описать способы распределения функций между обучающим и компьютером и затем осн. способы (пути) управления учебной деятельностью. Проект обучающей системы на данном уровне является основой для разработчиков системного *программирования* обучающей системы.

На уровне реализации выделяются два подуровня — педагогической и программной реализации. Первый из них содержит систему обучающих воздействий, результатом выполнения этого подуровня является проект-сценарий, определяющий, как действует обучающая программа в каждый момент обучения (см. также *Учебная игра*). На уровне программной реализации сценарий переводится в *программу* для компьютера.

Все уровни проектирования могут выполняться итеративно, т.е. последовательно, в которой они перечислены, не является жестко фиксированной, уровни не рассматриваются в качестве этапов проектирования. Каждый уровень может многократно уточняться по мере уточнения и совершенствования проекта обучающей программы.

Е.Д.Маргулис.

ПРОЕКТИРОВАНИЕ СВЕРХУ ВНИЗ, аналитическое проектирование — метод разработки *программного обеспечения* (в т.ч. *программ учебного назначения*), в основе которого лежит систематическое осуществление функциональной декомпозиции и модульного расслоения данных. Для задачи T П.с.в. производится по следующей схеме: 1) оценивается возможность достаточно четкого и простого представления решения данной задачи при помощи базисных конструкций того языка, на котором будет осуществляться *программирование*; 2) если результаты оценки приемлемы, приступают к написанию *программы*, в противном случае вводят понятия T_1, \dots, T_n , такие, что каждое из T_1, \dots, T_n хорошо описано и применение этих понятий в соответствии с их описанием позволяет написать удовлетворительную программу решения задачи T ; 3) каждое из понятий T_1, \dots, T_n рассматривается как новая задача, к которой применяется последовательность шагов 1—3. Результатом П.с.в. является программа решения задачи T с модульным расслоением, в котором каждому модулю соответствует вводимое при проектировании понятие. При этом модуль рассматривается не только как средство процедурной абстракции, определяющее некоторое действие, но и как средство абстракции данных, описывающее совокупность *процедур* и ассоциированных с ними структур данных.

В.И.Отенко.

ПРОЛОГ (от лат. PROgramming in LOGic — логическое программирование) язык логического *программирования*, содержащий и некоторые нелогические средства. Был создан в 1971; применяется в области *искусственного интеллекта*. Осн. идея П. состоит в том, что нужно грамотно и подробно на точном логическом языке описать условие задачи, и тогда решение получается как результат некоторого рутинного процесса, который выполняется на компьютере. Обычно условие задачи представляет собой множество формул специального вида в так наз. языке предикатов 1-го порядка; одна из формул выделена и наз. целью, остальные — посылками. В этом случае упомянутый процесс состоит в построении доказательства цели из посылок в *исчислении логическом*, единственным правилом вывода которого является так наз. правило резолюции. Написать программу на П. означает, во-первых, описать *предметную область* (ПО) конкретной задачи в виде объектов и отношений между ними и, во-вторых, дать формулировку задачи, т. е. поставить *вопрос*, на который система должна дать *ответ*, либо указать императив, содержащееся в котором требование она должна выполнить. При описании ПО в языке П. можно использовать такие структуры данных, которые позволяют описывать простые объекты, составные выражения и списки.

Пролог-программа представляет собой совокупность хорновских дизъюнктов (формул вида $\lceil A1 \vee \lceil A2 \vee \dots \vee \lceil Ak \vee B1$ (где $A_i, B1$ — атомарные формулы), среди которых выделен один (цель). Исполнение ее — это применение правила резолюции. Результат П.-программы — значение, которое получает некоторая входящая в цель переменная после успешного завершения доказательства цели. Базисная вычислительная модель П.-программы — разбивка задачи на подзадачи, унификация и резолюция. Некоторые версии П. (напр., *МПролог*) имеют средства поддержки модульных программ. В П. есть, хотя и весьма ограниченные, средства для модификации *базы знаний* (БЗ), однако саму программу (*процедуры* и структуры данных) модифицировать трудно.

П. может выступать в качестве *языка представления знаний*. Декларативный синтаксис П. в сочетании с процедурной семантикой позволяет комбинировать *декларативные представления знаний* и *процедурные представления знаний*. Кроме того, аксиомы П. легко воплощают декларативные знания; существует эффективная возможность отражения причинно-следственных связей и логического вывода; легко реализуется матем. модель предметной области в логике предикатов первого порядка; гибко и экономично представляется информация, что позволяет организовать разнообразное использование одних и тех же аксиом; наглядно описывается ПО. Однако базисный механизм П. довольно беден и позволяет строить только набор фактов или правил. Очевидно, что для реализации *знаний* необходимы удобные средства отражения модульности знаний, конструирования иерархических понятий и оперирования с неполными знаниями. Реализация в П. более сложных структур знаний приводит, как правило, к сложным синтаксическим конструкциям языка. Кроме того, отсутствие отрицания в П. требует переформулирования предложений языка таким образом, чтобы необходимость в отрицании отпала. Как правило, эти модификации аксиом находятся за пределами средств, поддерживаемых П. Отрицательным качеством для реализации БЗ является также отсутствие концепции вычисления значения функции. Встраивание равенства в П. позволило бы избавиться от этого недостатка. Кроме того, язык имеет ряд недостатков, не позволяющих ему конкурировать с универсальными (с точки зрения создания БЗ) языками программирования, требующими наличия мощных инструментальных средств: недостаточная гибкость механизма поиска при выполнении программы; слабое развитие нелогических средств; отсутствие отрицания и равенства.

Для П. разработаны французская версия (в Марсельском университете), ряд венгерских версий, напр., *МПролог*, известные реализации в Чехо-Словакии, Канаде. В нашей стране язык П. (базовый входной язык системы ПРОЛОГ-ЕС) впервые был реализован в Институте кибернетики им. В.М.Глушкова АН Украины

совместно с Рижским политехническим институтом. Существует несколько версий этого языка. В П.-II не различаются функциональные и предикатные символы — одни и те же символы могут использоваться для построения как объектов, так и отношений между этими объектами. Авторы языка IC-Пролог стремились пользоваться только чистыми логическими конструкциями, не разрешается добавление и удаление аксиом (дизъюнктов) в процессе обработки запроса. В качестве примитивов в язык введены отрицание и выражения для множеств. П. может использоваться во многих конкретных областях.

Системы автоматизированного проектирования. Сюда можно отнести разработанные в Венгрии системы поиска средств борьбы с загрязнением воздуха, проектирования планов квартир в панельном доме, проектирования рабочих помещений, синтеза новых лекарств, проектирования сложного программного обеспечения, разработанные в нашей стране системы подбора кадров по различным критериям, выбора режима ликвидации аварий в шахтах, организации управления учебным процессом, а также разработанную в Швеции систему регистрации кадров.

Системы, использующие логический вывод. К ним относятся системы синтеза программ, верификации программ, генерации программ, в частности на Коболе и ассемблера языке.

Системы обработки естественных языковых текстов. Примерами таких систем является диалоговая естественная языковая система ДАСЕЯ-1, предназначенная для исследования систем автоматического управления, лингвистический процессор для информационных систем ЛИПС.

Экспертные системы (ЭС). Примером ЭС, использующей П., может служить система технической диагностики DIGN для компьютера Правец-82. Известны реализации ЭС для оценки биофизических ресурсов Земли (ORBI), по страхованию, где знания об условиях страхования представлены на П., ЭС для логического проектирования аппаратного обеспечения компьютеров, экспертные обучающие системы (ЭОС). Напр., в ЭОС САКИО БЗ, решатель, естественный языковой интерфейс были реализованы на языке Турбопролог.

Диалоговые системы. В Марсельском университете разработана вопросно-ответная система, позволяющая из утверждений, сформулированных на французском языке, выводить следствия и таким образом отвечать на вопросы, поставленные на французском языке. Исследовалась возможность организации на П. генерирующих автоматизированных обучающих программ, позволяющих конструировать учебную информацию, контролировать правильность выполнения заданий, реализовать смешанный диалог компьютера и пользователя. В Институте кибернетики им. В.М.Глушкова АН Украины была разработана модель тренажера для операторов силовой установки супертраулера. Эффективность автоматизированного обу-

чения по к.-л. предмету будет высокой, если в интеллектуальную обучающую систему (ИОС) заложена достаточно развитая модель ПО. Для построения такой модели можно использовать П. Кроме того, в ИОС удобно использовать П. для представления изучаемых объектов со сложной логической структурой или процессов, требующих большого количества логических выводов.

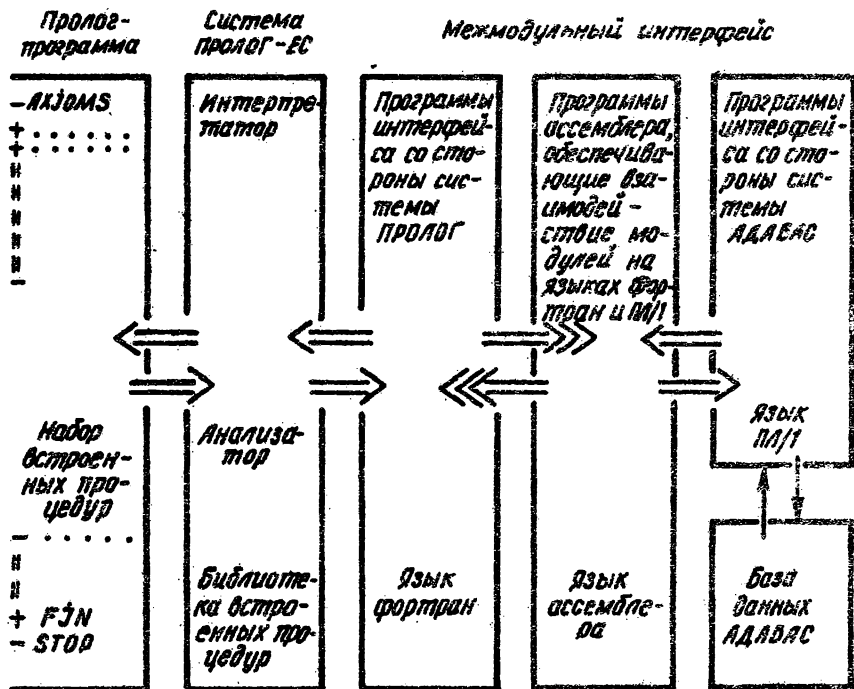
Моделирование и тестирование сложных систем. Применение средств логического программирования в решении подобных задач позволяет изменить порядок выполнения осн. этапов программирования сложных систем. Это в первую очередь относится к системам, реализующим при своем функционировании сложные логические формулы.

Инструментальные системы. Это реализованная на П. система представления знаний DLOG, интегрированные пакеты СПОК-ПРОЛОГ, АОС-ВУЗ/ПРОЛОГ, в которых П. реализован как внешняя динамическая функция и может быть использован для расширения возможностей контроля объектов и выдачи информации, лингвистический процессор ЗАПСИБ-ПРОЛОГ.

Л.Н.Гецко, В.В.Колос, С.П.Кудрявцева.

ПРОЛОГ-АДАБАС — интегрированный пакет прикладных программ, расширяющий функциональные возможности системы логического программирования ПРОЛОГ добавлением девяти встроенных процедур, связывающих Пролог-программу с базой данных (БД) АДАБАС. Объединяет существующую технологию БД и средств искусственного интеллекта. В П.-А. на систему ПРОЛОГ возлагается выполнение функций, обеспечивающих удовлетворение требований к БД по гибкости использования, созданию и обработке незапланированных запросов, упрощению процесса внесения изменений, предоставлению конечному пользователю удобного языка взаимодействия, т.е. функций по обеспечению дружелюбного партнерства со стороны компьютера. Главное достоинство такого интегрального подхода с позиций непосредственного манипулирования с данными из БД — возможность использования для этого формализма, заложенного в основы языка Пролог. Этот формализм представляет собой язык исчисления предикатов первого порядка, который, как правило, превосходит по мощности формализмы, используемые в промышленной БД. Важными характеристиками являются также легкость модификации Пролог-программы и прозрачность конструкций языка.

Создание системы П.-А. подразумевает создание межмодульного интерфейса, представляющего собой совокупность специальных программных модулей взаимодействия, ориентированных на БД АДАБАС (рис.). Необходимым условием взаимодействия между модулями в межмодульном интерфейсе является достижение требу-



емого соответствия типов данных языка программирования и параметров процедур, а также удовлетворение соглашений о межпрограммных связях при вызове процедур. Процедуры, связывающие систему ПРОЛОГ и БД АДАБАС, обеспечивают возможность поиска, чтения, обновления и добавления данных в БД, функционирующих в среде системы управления базами данных (СУБД) АДАБАС.

Ныне разработана первая версия интерфейса П.-А. Она включает набор процедур, являющихся расширением библиотеки встроенных процедур системы ПРОЛОГ, а также комплекс программ, обеспечивающих трансляцию процедур во внутренний (базовый) язык СУБД АДАБАС, управление операциями манипулирования и обмена данными между программой на языке Пролог и БД. К процедурам межмодульного интерфейса системы П.-А. относятся:

- ОТКРЫТЬ: открытие файлов БД для поиска, чтения и обновления данных;
- ЗАКРЫТЬ: закрытие файлов БД;
- СПРАВКА: получение информации о структуре записей файла БД;
- НАЙТИ: поиск записей файла БД по заданному логическому условию;

- ЧИТАТЬ: получение данных из записей файла;
- ЗАМЕНИТЬ: обновление значений полей записей файла БД;
- ДОБАВИТЬ: добавление новых записей в файл;
- УДАЛИТЬ: удаление записей из файла;
- ОСВОБОДИТЬ: завершение работы с записями, найденными процедурой НАЙТИ.

При выборе состава процедур, выполняемых ими функций и формата их задания, учитывались требования макс. обеспечения полноты операций манипулирования данными, а также ограничения СУБД АДАБАС (связанные с возможностями команд базового языка этой системы, на основе которого реализуются операции манипулирования данными) и языка Пролог (вызванные синтаксическими особенностями описания процедур на этом языке). Это связано с тем, что процедуры разработанного языка манипулирования данными являются расширением библиотеки процедур системы ПРОЛОГ и задаются в предикатной форме, как и осн. процедуры.

А.П.Ильяшенко, М.Е.Козлов.

ПРОЛОГ-ПРОГРАММА — программа, написанная на языке *Пролог*. Можно построить тремя способами: используя описание предметной области (ПО) на естественном языке; используя модель предметной области в логике предикатов первого порядка; используя процедурную интерпретацию языка Пролог.

Первый способ построения П.-п. рассматривает Пролог как средство для декларативного описания (см. *Декларативное представление знаний*) ПО. Этот подход предполагает следующую последовательность действий: описание ПО и условий задачи совокупностью утверждений на естественном языке; построение множества формул определенного вида по этой совокупности утверждений; перевод этого множества формул в совокупность дизъюнктов.

Сформулируем некоторые рекомендации к построению модели ПО в языке ПРОЛОГ. 1. Выделяем совокупность простых отношений (т.е. таких, которые не представлены в виде комбинации других), функций и бинарных операций. 2. Определяем простые утверждения, соответствующие данной ПО. Простым наз. такое утверждение, которое получается подстановкой вместо аргументов отношения переменных, констант, функциональных выражений (т.е. выражений, составленных из переменных, констант, символов операций). 3. Сформулируем утверждения, описывающие требуемую ПО. Потребуем, чтобы каждое утверждение имело вид:

“если $A_1 \dots A_n$, то Z_1 или $\dots Z_m$ ”,

где $m > 0$, $n \geq 0$, A_i и Z_i — простые утверждения. 4. При возникновении затруднений с наличием утверждений вида:

“для $\forall x_1, \dots, x_l \exists y_1, \dots, y_t$,

такие, что верно

$$P(x_1, \dots, x_n, y_1, \dots, y_m)''$$

автор должен найти такие функции f_1, \dots, f_m , которые по заданным x_1, \dots, x_n находят те y_1, \dots, y_m , для которых истинно

$$P(x_1, \dots, x_n, f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n));$$

это и будет утверждение, эквивалентное приведенному выше выражению. 5. Утверждения могут быть хорновскими и нехорновскими. Хорновским наз. утверждение, в заключении которого имеется лишь одно простое утверждение. Если все предложения хорновские, необходимы следующие действия: 1) каждое простое утверждение переводится в атомарную формулу, впереди записывается символ отношения, затем в скобках аргументы; 2) каждое хорновское предложение вида:

$$''\text{если } A_1 \text{ и } \dots \text{ и } A_n, \text{ то } B''$$

заменяется записью вида: $+B - A_1 - \dots - A_n$ (кюз). Если множество утверждений нехорновское, рекомендуется поступать следующим образом. В дизъюнкте, стоящем справа, выделяется любое одно простое утверждение. Пусть "если A_1 и \dots и A_n , то B или C_1 или \dots , или C_k " — условное утверждение, B — выбранное простое утверждение заключения; тогда для каждого C_i формулируем простое утверждение D_i , эквивалентное утверждению "неверно, что C_i ", т.е. его отрицанию. Получаем следующее утверждение, эквивалентное предыдущему условному утверждению:

$$''\text{если } A_1 \text{ и } \dots \text{ и } A_n \text{ и } D_1 \text{ и } \dots \text{ и } D_k, \text{ то } B''.$$

При втором способе построения П.-п. имеем описание задачи множеством формул логики предикатов первого порядка. Смысл логики предикатов как языка программирования связан с интерпретацией логического выражения

$$A \leftarrow B_1 \& B_2 \& \dots \& B_n.$$

Пусть $X_1, \dots, X_n \in M$, где M — множество допустимых термов атомарной формулы A , а Y_1, \dots, Y_m — множество допустимых термов атомарных формул B_1, \dots, B_n . Тогда приведенное выше выражение читается так: для любых X_1, \dots, X_n верно A , если существуют такие Y_1, \dots, Y_m , что верно B_1, \dots, B_n . Любую формулу исчисления предикатов первого порядка можно представить в виде предложения языка Пролог (в случае нехорновского предложения можно воспользоваться приемом, описанным выше). Процесс преобразования состоит из следующих этапов: 1) исключение знаков импликации; 2) уменьшение области действия знаков отрицания; необходимо, чтобы знак отрицания применялся не более чем к одной предикатной букве; 3) в области действия любого квантора переменная, связываемая им, является такой, что ее можно заменить переменной с другим именем, и это не приведет к изменению значения истинности логической формулы; следует переименовать переменные так, чтобы каждый квантор имел свою собственную переменную; 4) исключение кванторов существования; 5) вынесение вперед кванторов всеобщ-

ности; 6) приведение формулы к конъюнктивной нормальной форме; 7) исключение операции конъюнкции заменой записи вида $A \wedge B$ двумя формулами A и B .

Третий способ построения П.-п. удобен, когда необходимо использовать язык Пролог для целенаправленного вычисления. Рассмотрим процедурную интерпретацию Пролога. В соответствии с этой интерпретацией предложение языка логического программирования $+A - B_1 - \dots - B_m, m \geq 0$ рассматривается как определение процедуры. Положительный литерал A интерпретируется как имя процедуры. Отрицательные литералы B_1, \dots, B_m интерпретируются как вызовы процедур, составляющие процедурное тело определения. Целевое предложение (теорема), т.е. предложение, не содержащее положительных литералов, интерпретируется как множество вызовов процедур, которые должны быть выполнены. Чтобы построить П.-п., используя процедурную интерпретацию Пролога, следует представить имеющуюся задачу в виде последовательности подзадач. При разработке обучающих систем в виде П.-п. удобно представлять объекты и процессы со сложной логической структурой, требующие большого количества логических выводов для определения значений их структурных и функциональных характеристик.

В.В. Колос.

ПРОФИОРИЕНТАЦИЯ С ПОМОЩЬЮ КОМПЬЮТЕРА — использование компьютера для автоматизированного составления и применения профессиограмм, компьютерной психодиагностики, статистической обработки результатов, профконсультационной работы, осуществляемой в режиме диалога пользователя с компьютером. Распространены автоматизированные диалоговые профориентационные системы, являющиеся основой автоматизированных профцентров, в методическое обеспечение которых входят: справочный и профинформационный материал об учебных заведениях, на которые ориентирован данный профцентр, и соответствующие профессиограммы; диагностический инструментарий, алгоритмы, инструкции и интерпретации к нему; методическое обеспечение по использованию диагностического комплекса; методическое обеспечение индивидуальной профконсультации, методы статистического анализа результатов. Перспективным направлением является создание моделей профессий с перечислением типичных личностных свойств с последующим автоматизированным подбором характеристик обследуемой личности (оптанта) к одному из типов деятельности. Такой подбор необходимо проводить под контролем психолога, имеющего возможность вмешиваться в последовательность предъявления отдельных диагностических методик и вводить методики либо задачи, которые могли бы уточнить характеристики оптанта. В каждом конкретном случае психолог также должен решать вопрос о необходимости и форме предъявления

оптанту результатов диагностики. Современные технические и программные средства позволяют использовать для целей П.с п.к. графические дисплеи, игровые диагностические методики, повысить эффективность деятельности психолога-диагноста, уменьшить дефицит психологов, работающих в сфере профориентации. Важное значение в П.с п.к. имеет применение автоматизированных систем *информационного обеспечения*. Ныне можно выделить такие осн. направления использования компьютеров в профессиональной психодиагностике.

Автоматизация процесса психологического и психофизиологического тестирования профессионально значимых качеств и обработки его результатов. В автоматический режим переводятся многофакторные личностные опросники, тесты общего интеллекта, опросники интересов, тесты, диагностирующие профессионально значимые качества отдельных познавательных функций, методики для диагностики свойств нервной системы, психомоторики и т.д. Для автоматического обобщения результатов используются различные принципы. При сопоставлении набора психологических характеристик оптанта с наборами требований профессий можно, напр., применять метод матем. теории распознавания образов — алгоритм вычисления оценок. При переводе многих психодиагностических методик на компьютер следует учитывать необходимость их ревалидации и разработки новых оценочных шкал. *Компьютеризация* профориентационной работы не снижает требований к качеству проведения обследования, обработки результатов и их интерпретации. *Принятие решения* о результатах диагностики качеств личности требует специальных знаний и практического опыта работы с психодиагностическими методиками. Результаты автоматизированной психодиагностики помогают психологу глубже разобраться в специфике проблемы, стоящей перед оптантом.

Использование основанных на компьютерах тренажеров и игр, моделирующих определенный вид профессиональной деятельности. Это направление характерно для профотбора (напр., на операторские профессии). Такие методы позволяют весьма эффективно судить о пригодности оптанта к данному виду труда по специфике его обучаемости, характеру ошибок и трудностей и т.д.

Реконструкция систем субъективных представлений оптанта о профессиях. Это направление целесообразно рассматривать как отдельное, отличное от традиционной психометрики и тестирования, т.к. оно ориентировано на анализ качественной неповторимости внутреннего мира личности. В основе подхода лежит техника репертуарных решеток. В ходе консультации определяется круг знакомых оптанту специальностей и индивидуальный набор субъективных измерений, смыслов, "конструктов", сквозь

которые оптант видит мир профессий. С помощью программ факторного и кластерного анализа осуществляется обработка индивидуальной решетки-матрицы, в которой оптант оценил все специальности по всем "конструкциям". Полученные структуры дают возможность судить об иерархии и конфликтности глубинных мотивов человека, определяющих его предпочтения в области профессионального труда.

В.М.Бондаровская, А.Д.Винокур, Н.Н.Нечаев.

ПРОЦЕДУРА (франц. *procedure*, от лат. *procede* — продвигаюсь, прохожу) — система последовательно осуществляемых *операций*. После любой операции, входящей в состав П., либо больше не выполняется никаких операций, либо выполняется некоторая определенная операция, либо имеет место разветвление П., т.е. выполняется одна из некоторого конечного набора операций. Одиночную операцию можно рассматривать как частный вид П.

В *программировании* термин "П." часто используется для обозначения некоторой вспомогательной программы. П. состоит из заголовка (спецификации П.) и тела. В заголовке указываются имя П., информация о параметрах и типе (напр., о рекурсивности, реентерабельности и пр.), достаточная для корректного вызова П. Тело П. состоит из операторов языка программирования, на котором написана П. Различают П.-функции и П.-подпрограммы. П.-функции в результате, как правило, выдают единственное значение, что сближает их с матем. интерпретацией функции. П.-подпрограммы в качестве результата могут изменять значения нескольких переменных.

Г.А.Балл, В.А.Третьяк.

ПРОЦЕДУРНОЕ ПРЕДСТАВЛЕНИЕ ЗНАНИЙ — *метод представления знаний*, реализуемый при помощи процедур преобразований для локального участка или всей рассматриваемой предметной области. Делится на универсальные языки программирования и продукционные методы представления знаний. Различие между декларативным представлением знаний и П.п.з. может быть выражено как различие между "знать что" и "знать как". Выбор представления знаний обуславливается спецификой задачи. Напр., действия робота в окружающей его среде целесообразно описывать в виде процедур. П.п.з. характерно для экспертных систем. Использование процедур удобно для представления знаний второго порядка, т.е. для выражения фактов о том, как использовать факты. П.п.з. применяется в идеальных эвристических средствах, позволяя легко учитывать специфику конкретных предметных областей. Использование П.п.з. целесообразно тогда, когда число процедур невелико, и они применяются при работе с надежными, полными и

непротиворечивыми *данными*. Иначе возникают непредсказуемые побочные эффекты от изменения или расширения *базы знаний*. В этом случае множество процедур приходится структурировать: напр., на множестве правил (продукций) вводить *фреймы, семантические сети* или их различные сочетания.

В *автоматизированных обучающих системах* процедуры применяются при моделировании отдельных этапов процесса обучения, напр., при формировании *моделей обучаемых* на основе *анализа ответов обучаемых*. С помощью процедур в этих системах подается учебный текстовой и графический материал, а также организуется звуковое сопровождение. Во многих системах П.п.з. реализуется с помощью специальных языков типа PLANNER, QA-4, CONNIVER и др.

О.Н.Золотонуп.

ПРОЦЕССОР — 1) Часть *электронной вычислительной машины*, реализующая процесс переработки *информации*. В состав П. входят устройство управления, арифметико-логическое устройство и в ряде случаев *запоминающее устройство*. Есть П. центральный (для выполнения вычислений, задаваемых абонентом), периферийный (для управления вводом—выводом *данных*), управляющий (для переработки информации), специализированный (для решения задач определенного типа) и т.д. В многопроцессорных системах П. работает с одним главным запоминающим устройством. 2) Сложная логическая программа, входящая в состав системы автоматизации *программирования*, напр., П. сборки рабочей программы.

PROUST — экспериментальная интеллектуальная *обучающая система* для обучения программированию на языке *Паскаль*, выполняющая *семантический анализ* правильности программ. Анализ программ основан на знании программируемой задачи. Знания о задаче состоят из описаний множества объектов *данных* и целей *программирования*. Знания о программировании хранятся в виде *фреймов*, каждый из которых соответствует одному *понятию*. Осн. видами знаний являются знания о целях и знания о планах. Цели — это требования, изложенные в описании задач. Фрейм каждой цели имеет слот “Примеры”, в котором перечисляются различные планы — стереотипные методы достижения данной цели.

В процессе анализа программы PROUST выдвигает гипотезы о том, какие планы использовал *обучаемый* для достижения определенной цели, а затем проверяет эти гипотезы. Если ни оди. из известных системе планов не укладывается в программу *обучаемого*, система обращается к *базе данных* (БД) ошибок с целью найти объяснение обнаруженного расхождения. Объяснение достигается

путем применения правил ошибок, каждое из которых состоит из двух частей: проверочной, позволяющей по различиям в планах установить, применимо ли данное правило, и результативной, дающей объяснение этим различиям. Сообщая обнаруженные факты, система придает каждому описанию ошибки вид *текста* на естественном (английском) языке и, если необходимо, вырабатывает данные, демонстрирующие присутствие этой ошибки. Напр.: “Ваши пропущены текст на сигнальное значение. Если сигнальное значение подается на вход сразу же за неположительной величиной, то Ваша программа примет его за допустимое значение. Чтобы в этом убедиться, подайте на вход вашей программы такую последовательность данных: 5, —5,99999.” Имеется в виду, что 99999 (сигнальное значение) — это признак конца ввода данных.

Планы с программой обучаемого сопоставляются поэтапно. Еще до того, как производится анализ целей и планов, делается синтаксический разбор программы, в результате чего строится дерево разбора. Последующий анализ производится на дереве разбора. Система по очереди выбирает цели из описания задачи, затем в это целевое выражение подставляет все объекты, значения которых уже известны. Далее из БД извлекаются те планы, которые можно использовать для достижения данной цели. Если ни один из планов не сопоставляется с программой, система должна искать ошибки, приводящие к несоответствию. Система PROUST испытана на большом количестве программ, написанных начинающими программистами. В ходе проведенных испытаний рассмотрено 206 программ. В 72% случаях система достигла полного понимания студенческих программ, обнаружив 94% всех ошибок. 6% ошибок не были выявлены, а 55% оказались “ложной тревогой”.

Р.Ковалюнас.

ПСИХОЛОГИЧЕСКАЯ ГОТОВНОСТЬ — устойчивое образование личности, включающее мотивационный, интеллектуальный и эмоционально-волевой компоненты, а также комплекс *знаний*, умений, навыков и личностных качеств, адекватных требованиям, содержанию и условиям деятельности. Помимо личностного подхода, имеется и функциональный подход, объясняющий готовность к деятельности как определенное состояние психических функций, обеспечивающее высокий уровень достижений в осуществляемой деятельности. Между двумя подходами существует тесная связь: временная или функциональная готовность, т.е. состояние готовности, — каждый раз создаваемое функциональное состояние долговременной или личностной готовности, т.е. подготовленности личности; П.г. — это не только результат подготовки, но и ее цель, причем готовность личности к трудовой деятельности и психологическая подготовка к ней рассматриваются с точки зрения единства сознания и деятель-

ности как одного из осн. принципов развития личности. В целом П.г. к труду рассматривается как сложное многоуровневое образование, включающее операциональные и личностные компоненты. К первым относятся знания, общетрудовые умения и навыки, выступающие в то же время и как предпосылки положительного отношения к труду. Ко вторым — установка на труд, система привычек и отношений к труду. Установка на труд, возникающая на почве единства социальной ситуации развития личности и ее потребности в самоопределении, становится основой активности личности, выработки ее жизненной позиции (это относится как к учебе и общественной деятельности, так и к профессиональной деятельности).

Психологические особенности и структура готовности к труду в настоящее время во многом обусловлены тенденцией к возрастающей интеллектуализации труда. Психологические и социологические исследования свидетельствуют о непосредственной связи между уровнем готовности к труду и возможностями так наз. интеллектуального насыщения трудовых процессов. Одно из новых направлений здесь связано с изучением компьютера как средства формирования П.г. к труду, а также П.г. к использованию компьютера в трудовой деятельности. Предлагается, в частности, трехкомпонентная структура такой готовности, состоящая из мотивационного, операционно-технического и интеллектуально-творческого компонентов. При этом под мотивационным компонентом понимается осознание общественной и личной значимости данного вида труда, потребность в труде, положительное отношение и интерес к труду, привычка своевременно обращаться к компьютеру при решении задачи из любой области, целенаправленность в трудовой деятельности. В операционно-технический компонент входят трудовые умения и навыки работы с компьютером и его периферийным оборудованием. Интеллектуально-творческий компонент включает умения ставить и решать задачи в понятной для компьютера форме, организовывать поиск информации, необходимой для решения поставленной задачи, рефлексию на способы трудовой деятельности, определенный уровень развития эмоционально-волевой регуляции.

Важным условием формирования П.г. к труду, опосредованному применением компьютеров, является преодоление “психологических барьеров”, связанных с: а) непониманием преимуществ применения компьютера и вследствие этого отказом от приобретения новых машин или неиспользованием уже имеющихся; б) необходимостью овладения целым рядом новых знаний, а также изменением привычных методов деятельности; в) сложностями переобучения и перестройки навыков, происходящих в связи с постоянным изменением типов компьютеров и их программного обеспечения, г) непривычностью компьютера как технического устройства; д) трудностями в освоении символического языка общения с компьютерами, отличающегося от языка человеческого общения; е) недоверием к

результатам работы с компьютером; ж) нежеланием преодолевать организационные трудности установления связи с компьютером, а также сложности во взаимоотношениях с обслуживающим его персоналом; з) защитной мотивацией, связанной с боязнью потерять место работы из-за сокращения штатов, обусловленного внедрением компьютеров, или перестать быть источником уникальных знаний, поддерживающих высокую самооценку и профессиональный статус. Осн. путь формирования П.г. к деятельности, опосредованной применением компьютеров, связан с овладением *компьютерной грамотностью* на уровне, необходимом для успешного выполнения осн. профессиональных обязанностей.

Структура П.г. к профессионально-педагогической деятельности включает адекватную *мотивацию*, обуславливающую положительное отношение к педагогической деятельности и ее успешность, необходимый объем специальных и психолого-педагогических знаний, определенный уровень сформированности профессиональных умений и навыков; ряд авторов выделяют также в качестве подсистемы комплекс профессионально значимых свойств личности, включающий способности, характер, темперамент, особенности эмоционально-волевой регуляции и пр. См. также *Психология компьютеризации*.

Е. Д. Маргулис.

ПСИХОЛОГИЯ КОМПЬЮТЕРИЗАЦИИ — область психологии, предметом которой являются порождение, функционирование и структура психического отражения в процессе деятельности, связанной с созданием и использованием компьютера и его *программного обеспечения*. Разрабатывается психологом О.К.Тихомировым и его сотрудниками. Осн. проблемы П.к.: закономерности и принципы организации различных видов человеческой деятельности, опосредованной компьютерами. Предмет П.к. и ее проблематика определяют и необходимость развития принципов психологии (в частности, принципов детерминизма, развития, системности), уточнения ее категориального аппарата (в т.ч. категорий психического отражения, деятельности, общения, личности). Они порождают новые психологические проблемы (см. *Психологическая готовность*), связанные с анализом природы психики, предмета, структуры и методов психологии, актуализируют новые области и прикладные задачи разработки психологических основ автоматизации систем *обучения, проектирования, управления, разработки экспертных систем*, а также систем, используемых в профориентации, профобучении, профдиагностике, с анализом воздействия вычислительной техники на формирование новых общностей, социальных групп, особенностей процесса общения в них и т.д. Проекция П.к. в сферу образования выдвигает немало специальных проблем, поскольку *компьютеризация* в сфере образования по своей сути — проблема междис-

циплинарная, комплексная. Ее оптимальное решение невозможно без анализа особенностей функционирования используемых компьютеров, определяющих, с одной стороны, собственно технические аспекты, а с другой — принципиальные возможности применения компьютеров в качестве эффективных средств учебно-воспитательной, научно-исследовательской и управленческой деятельности.

Е.Д. Маргулис.

ПСИХОЛОГИЯ ПРОГРАММИРОВАНИЯ — раздел психологии, изучающий аспекты *взаимодействия человека-машинного*, связанные со способами работы человека с программными структурами и системами, с теми воздействиями, которые различные структуры и системы оказывают на его деятельность и личность. Включает знания о возможностях и способностях человека как создателя вычислительных систем; о влиянии конструкций конкретного языка программирования (ЯП) на эффективность процесса написания и отладки программ пользователем; о влиянии концептуальной модели базы данных на формулирование пользовательских запросов о ней; о влиянии опыта работы пользователя с различными системами обработки текста при освоении новой системы; об учете интеллектуальных способностей пользователей при построении различных пользовательских интерфейсов и т.д. Осн. цель П.п. — создание наиболее благоприятных условий для работы человека в процессе программирования, отладки и эксплуатации программ. Предмет П.п.: различные способы использования компьютера человеком; разработка программного обеспечения компьютера; применение средств информационного обслуживания; ЯП, языки управления информационными системами, средства обращения к базам данных; средства автоматизированного обучения; персональные вычислительные средства; программы редактирования; обработка слов и использование терминала неопытными пользователями. П.п. способствует разработке методов создания систем программирования, требующих меньших затрат на отладку, модернизацию и поддержание в рабочем состоянии; проблем правильной организации взаимодействия в коллективах программистов в процессе создания большой системы и т.д. В П.п. используются методы экспериментальной психологии, анализа процессов мышления и восприятия, методы социальной психологии, психологии труда.

Осн. результаты применения знаний о свойствах мышления человека достигаются в практике программирования путем улучшения использования программных средств, усовершенствования обучения, разработки методов оценки качества программного обеспечения и методов оценки способностей и возможностей программиста. Одна из главных проблем в исследованиях по П.п. заключается в больших индивидуальных различиях программистов.

Это связано с различиями в диапазоне знаний, приобретенных на опыте, различиями в профессионально важных качествах личности программистов. Вторая важная проблема — качественные отличия процесса решения задач программирования от процессов решения задач, традиционно рассматриваемых в психологии мышления. В программировании компьютеров не существует ни вариантов с ограниченным выбором, ни четко определенного состояния, соответствующего решению. Особый интерес представляет изучение трудностей, возникающих при решении задач, связанных с временными соотношениями по сравнению с решением задач с пространственными соотношениями. Использование ЯП — третья существенная проблема П.п. Одной из характеристик языков, предназначенных для организации взаимодействия с компьютером, является степень принуждения пользователя к описанию его задач на языке машинных процедур, далеком от естественного языка. Эта характеристика относится ко всем языковым уровням — от машинных языков и языков ассемблеров, через языки высокого уровня к языкам запросов и, наконец, к естественному языку. При составлении программ люди затрачивают значительно больше времени на описание процедур манипулирования данными, чем на описание процедур управления. Из этого следует, что естественный язык не подходит для составления спецификаций процедур, значительно эффективнее применение его ограниченного и структурированного подмножества. Весьма распространено «движение» структурного программирования. Так называемая группа исследователей показала: превосходство структурированной основной конструкции IF THEN ELSE над простой конструкцией GOTO; потенциальные преимущества использования маркеров, ограничивающих программные блоки, управляемые конкретным условным выражением; переменный характер преимуществ различных методов программирования в зависимости от природы программируемой задачи; возможность повышения скорости и безошибочности программирования за счет применения стандартной процедуры генерирования синтаксической структуры условных операторов. Возможности структурирования данных и соответствующего документального представления получаемых структур могут оказаться весьма существенными факторами увеличения производительности труда при программировании.

Представление знаний программиста существенно влияет на уровень его квалификации, формирование профессиональных умений. Совершенствование профессионального мастерства определяется не просто процессом заучивания длинного списка фактов, а творческими усилиями, направленными на познание фундаментальной структуры соответствующей области знаний. Структуры знаний у высококвалифицированных программистов гораздо более схожи между собой, чем у малообученных программистов или новичков. В П.п. значительную роль играет проблема изучения особенностей пони-

мания текстов программ. Эта проблема решается в контексте рассмотрения логических выводов как неотъемлемых факторов понимания схем представления знаний, лежащих в основе этих выводов, и схем индексирования памяти при хранении усвоенных понятий. Для повышения эффективности взаимодействия пользователя с компьютером в П.п. ведется поиск наиболее подходящей терминологии для выбора названий команд. В П.п. широко обсуждаются методологические принципы психологии программных средств. Существует два подхода к проведению эмпирических исследований — планируемый, жестко управляемый, строгий, систематический эксперимент в узкой области и менее строгий исследовательский (обнаруживающий) эксперимент, позволяющий получить информативные данные в широкой области, предоставляющий расширенные возможности интерпретации результатов.

Решение задач, стоящих перед П.п. позволит не только повысить эффективность разработки программ, эффективность и надежность их использования, но и улучшить успешность усвоения основ программирования и использования компьютеров широкими кругами пользователей. При овладении компьютерной грамотностью, при усвоении навыков работы с компьютером данные П.п. могут способствовать повышению эффективности обучения и должны широко использоваться в компьютеризации образования (см. *Психолого-эргономические основы компьютерной технологии обучения*).

В.М.Бондаровская, Н.И.Повякель.

РАБОЧАЯ ПОЗА — положение тела человека в процессе деятельности, наиболее частое и предпочтительное взаиморасположение частей тела при выполнении трудовых операций. Изменение Р.п. связано с рабочими движениями, причем поза рассматривается как пространственная граница фазы движения (начальная, граничная, конечная). Положения тела и наиболее частые позы, необходимые для выполнения трудовых заданий, являются одним из факторов, определяющих пространственную организацию рабочего места. Существуют определения рациональной Р.п., соответствующие критериям функционального комфорта: 1) выпрямленное положение позвоночного столба с сохранением его естественных изгибов; 2) миним. нагрузка на мышечную систему тела человека, связанную с поддержанием Р. п.; 3) отсутствие болезненных ощущений в результате воздействия элементов кресла на тело сидящего человека; 4) наличие субъективной оценки Р.п. как удобной; 5) обеспечение угла сгибания рук в локтевых суставах в пределах 70—90°; 6) обеспечение угла сгибания ног в коленном и голеностопном суставах в пределах 95—135°. Р.п. в положении сидя тесно связано с формой рабочего сиденья. В этом контексте различают три осн. Р.п.: 1) передняя (центр тяжести в передней части поверхности сиденья);

2) средняя (центр тяжести в средней части поверхности сиденья);
3) задняя (центр тяжести сзади). Для работы с компьютером наиболее подходящей считается средняя поза.

Р.п. — важнейшая характеристика организации работы *обучаемого—пользователя* компьютера. Работа с компьютером может приводить к физическому дискомфорту и утомлению. Существует несколько неудобных поз, которые принимает пользователь: туловище отклонено назад, а шея — вперед; высокое положение рук; руки вытянуты вперед; высокое положение предплечий и кистей рук относительно уровня плеч и кистей. Чтобы избежать их, необходимо выполнять определенные рекомендации по Р.п.: 1) шея пользователя не должна быть наклонена более чем на 20° (угол между осевой линией голова—шея и осевой линией туловища); 2) плечи не должны быть напряжены; 3) угол между плечом и предплечьем (локтевой сустав) должен составлять 90° и больше; 4) предплечье и кисти должны находиться в одной горизонтальной плоскости.

Особенно большое внимание должно уделяться вопросам Р. п. при эргономическом проектировании *автоматизированных рабочих мест* (АРМ) для учебных заведений. При этом следует учитывать, что ребенок с прямой спиной не может высидеть в этой позе и нескольких минут, так как его тазобедренный сустав может сгибаться лишь на 60° ; изгиб позвоночника характерен для положения стоя. В положении сидя круглая спина является единственной естественной позой. Проблемы Р.п. обучаемого при работе на АРМ можно решить на основе использования специальной компьютерной мебели, эргономичных технических и программных средств деятельности обучаемых, а также эргономически обоснованных решений по расположению элементов рабочего места.

В.М.Бондаровская.

РАБОЧЕЕ МЕСТО — пространство, оснащенное некоторыми средствами для выполнения определенной деятельности. В обучении Р.м. должно способствовать повышению эффективности учебного процесса путем снижения физиологического и психологического дискомфорта благодаря созданию условий для быстрой и безошибочной работы и удобств использования технических средств, обеспечения правильной *рабочей позы* и оптимальных рабочих зон.

Организация Р.м. должна удовлетворять эргономическим требованиям: габаритные размеры Р.м., его элементов и их пространственное расположение должны соответствовать антропометрическим данным соответствующего контингента *пользователей*; конструкция Р.м. должна обеспечивать оптимальные (с точки зрения психофизиологических критериев) угловые параметры рабочей позы для нормального функционирования внутренних систем человека; параметры *освещенности* Р.м. должны соответствовать психофизиоло-

гическим требованиям для визуального наблюдения; художественно-композиционное решение Р. м. нужно выбирать на основе учета психофизиологических и социально-культурных требований. Элементы *автоматизированного рабочего места* (АРМ): рабочая мебель (стол, стул, поверхность для записи, поверхность для печатных материалов, подставка для ног); компьютерная конфигурация (монитор, клавиатура, периферийные устройства); учебные средства для индивидуальной учебной деятельности (тетрадь, учебник и др.).

При выборе рабочей мебели необходимо учитывать, что: функциональные характеристики мебели должны соответствовать специфике учебного труда; необходимо обеспечить регулируемость параметров Р.м., т.к. оно используется многими *обучаемыми* и для разных целей; при кабинетной системе обучения нужно предвидеть пространство для расположения личных вещей обучаемого; для обеспечения хорошего обзора со всех Р.м., для облегчения личного контакта *преподавателя* и обучаемых и для надежной работы компьютера необходимо предусмотреть пространство для системы электропитания; нужно учитывать количество обучаемых, одновременно использующих одно Р.м. Размеры учебной мебели должны обеспечивать рабочую поверхность для размещения технических средств и для индивидуальной работы каждого пользователя Р.м., работающего с тетрадью и учебным пособием. Определяя компьютерную конфигурацию для оснащения АРМ, учитывают предметную ориентацию процесса обучения. Тип предмета связан с характером учебной *информации*, с преобладающим типом *учебных задач*, решаемых с помощью компьютера, функциональной структурой деятельности обучаемого. При определенных размерах рабочей поверхности и размещении компьютерной конфигурации компоновка Р.м. ограничена антропометрическими требованиями (зона оптимальной досягаемости и наилучшего обзора) и функциональной структурой деятельности человека (важность, частота, продолжительность и последовательность в использовании элементов Р.м. при выполнении рабочих операций).

Пространственную ориентацию Р.м. в помещении определяют в соответствии с источниками естественного и искусственного освещения (окна, осветительное оборудование), учебными техническими средствами для коллективного пользования (черная доска, видеоэкран и др.). Последовательность организации пространственной ориентации должна учитывать: ориентацию относительно окон; проектирование центрального искусственного освещения; определение местоположения и мощности локального искусственного освещения, если центральное не отвечает требованиям эргономики; взаимное расположение индивидуальных Р.м.; определение положения учебно-технических средств коллективного пользования. Художественно-композиционные решения должны приниматься с учетом требований

промышленной эстетики, связанных с выбором материалов, формой и цветовым кодированием устройств.

Т. Дамянова.

РАЗВЕТВЛЁННАЯ ОБУЧАЮЩАЯ ПРОГРАММА — обучающая программа, в которой предъявление обучаемому очередного кадра (порции) учебной информации зависит от степени усвоения материала предыдущего кадра, от количества и характера ошибок. См. *Программированное обучение.*

Н. М. Розенберг.

РАЗРАБОТЧИК — специалист по проектированию и реализации программного обеспечения. В области компьютерного обучения Р. — специалист по проектированию и реализации инструментальных систем для создания программ учебного назначения.

В. А. Третьяк.

РАКУРС (Разработчик учебных КУРСов) — универсальная авторская система, предназначенная для поддержки этапов проектирования и реализации программ учебного назначения (ПУН) линейной и разветвленной структуры демонстрационного, информационно-справочного, обучающе-контролирующего и других типов. Разработан в Московском институте инженеров гражданской авиации для микро-ЭВМ типа ЕС-1840, ЕС-1841, Искра -1030, Электроника-85, ДВК-3, ДВК-4, а также локальной информационной сети радикальной архитектуры, созданной на основе ПЭВМ. В состав Р. входят следующие компоненты: средства реализации режима обучения; библиотека курсов и средства работы с ней; текстовый редактор; графический редактор; средства анализа и обработки ответов обучаемого; средства сбора статистики об обучаемом; конвертор курсов из формата АСТРА/МИКРО (предыдущая версия системы) в Р.; конвертор курсов из формата АОС-ВУЗ/микро в Р.; средства выхода из Р. в операционную систему и возврата в текущую точку системы и др. Работа в Р. основана на принципах меню и интервью. ПУН, генерируемая системой, наз. курсом, который состоит из кадров.

Р. может работать по следующим режимам: резервирование курса; форматирование и корректировка курса; копирование и удаление курса; работа с оглавлением библиотеки курсов, вывод состояния библиотеки курсов; режим обучения/контроля. Выбор режима осуществляется из меню. В режиме формирования и корректировки Р. позволяет просматривать, редактировать, копировать и удалять кадры, корректировать параметры курса, выводить информацию о кадрах и переходить в режим обучения/контроля. Кадр в Р. может быть двух типов: информационный и контролирующий. Параметрами кадра

являются характеристика кадра (текущий или последний), наличие в нем *текста* и/или графики, область пер кодов; контролирующий кадр может также содержать эталонный ответ. Используя макроинструкции, *автор* может вставлять в текст кадра в режиме обучения/контроля значения системных переменных (фамилия обучаемого, текущая дата и время, предыдущий ответ обучаемого, данные статистики хода обучения и т.д.), а также управлять работой системы (вызвать калькулятор, выполнить командный *файл*, включить/отключить управление *терминалом*, поместить кадры, к которым можно вернуться из любого кадра и т.д.). Графический редактор Р. позволяет формировать изображение по точкам, рисовать линии, рамки, эллипсы, окружности, прямоугольники, копировать и перемещать изображения, сохранять изображения на диске и считывать его с диска, работать с генератором шрифтов, строить графики стандартных функций и т.д. Р. имеет те же возможности обработки *ответов*, что и язык *описания курсов* в АОС-ВУЗ.

Т.А.Лисюк.

РЕЖИМ ДИАЛОГОВОЙ ТРЕНИРОВКИ (РДТ) — форма организации автоматизированного учебного процесса, с помощью которой у *обучаемого* вырабатываются и закрепляются практические навыки работы в изучаемой *предметной области* (ПО). Наиболее эффективен при изучении различных программных систем и средств, непосредственно связанных с функционированием компьютера, напр., при освоении новых *операционных систем* или *языков программирования*. РДТ осуществляется средствами тренирующих пакетов диалоговых программ (ПДП), которые обеспечивают *диалог*, выполнение необходимых функциональных действий, а также помощь и объяснение ошибок. Большинство из существующих программ реализуют РДТ именно в области *программного обеспечения*. Для организации диалоговой тренировки в других ПО наибольшее распространение получили игровые методы и соответствующие программы. Важнейшая особенность РДТ — активное применение *обучающих воздействий*. *Пользователь*, как правило, имеет возможность получить развернутое объяснение своей ошибки, помощь или подсказку, соответствующий ситуации учебный *текст* или пример, если в этом есть необходимость.

Процесс диалога в РДТ может осуществляться под управлением ПДП либо с помощью ПДП. В первом случае компьютер осуществляет пошаговое управление действиями пользователя. Это позволяет обнаруживать и объяснять ошибку практически сразу же после того, как она допущена. Во втором варианте организации РДТ пользователю предоставляется значительно большая самостоятельность. ПДП выдает пользователю необходимые указания, пояснения и диагностику лишь в начале и конце отдельных этапов тренировки.

Возможность выполнять достаточно сложные действия, направленные на достижение некоторой цели, создает условия тренировки, максимально приближенные к самостоятельной деятельности. В то же время контроль со стороны ПДП и инструкции, касающиеся дальнейшей тренировочной работы, обеспечивают поэтапную диагностику ошибок и позволяют пользователю выработать определенный порядок действий, который необходимо (или желательно) соблюдать в реальной работе. Использование к.-л. других ПДП в ходе диалоговой тренировки возможно как в первом, так и во втором случае организации РДТ и, как правило, имеет место, если допускаются грубые или повторяющиеся ошибки, т.е. очевидна необходимость доучивания пользователя. Можно также обращаться к другим ПДП по запросу самого пользователя.

Поскольку диалог в РДТ является одной из форм обучающего диалога, то и применение средств универсальных *автоматизированных обучающих систем* (таких, как АОС-ВУЗ или АОС-М) наиболее удобно при создании тренирующих ПДП. Существует два осн. подхода к реализации РДТ в АОС. В первом случае элементы системы или языка, которые изучает пользователь, моделируются с помощью ПДП. Разработка таких программ трудоемка, но позволяет реализовать наиболее точную и подробную диагностику и объяснение ошибок. ПДП данного типа обычно осуществляет пошаговый контроль и управление работой пользователя. Ограничение возможных действий пользователя на каждом отдельном шаге тренировки обеспечивает возможность быстрого анализа его действий (сообщений). Для РДТ в такой реализации характерен обмен сообщениями, не требующими длительной обработки. Поэтому, если несколько пользователей одновременно работают в РДТ с одной и той же АОС, то "циклический" алгоритм, предусматривающий строго последовательное и полное обслуживание каждого очередного запроса (что реализовано, в частности, в АОС-ВУЗ) вполне приемлем и не вызывает неудобств в процессе диалога. При втором подходе пользователь тренируется, используя реальные программные средства. ПДП, обеспечивающий РДТ, осуществляет двустороннюю связь с изучаемой системой. Моделирующих функций от ПДП в этом случае не требуется. Однако при такой организации диалоговой тренировки возможны ситуации, в которых для выполнения запроса пользователя необходим достаточно большой отрезок времени работы процессора (порядка десятков секунд и более). Причинами таких затрат могут быть либо сложность алгоритмов, реализуемых изучаемым ПДП, либо большие объемы перерабатываемых данных, либо ошибки пользователя, вызывающие "зацикливание" и другие непроезводительные действия компьютера. "Циклический" алгоритм обслуживания не ориентирован на выполнение в ходе диалога больших программ. Такой алгоритм не позволяет обеспечить в случае сложных запросов необходимый темп обмена сообщениями для всех пользо-

вателей, работающих с АОС. Передача на выполнение программы или запроса, требующих значительного времени работы процессора, вызывает замедление работы (на это время) всех одновременно работающих пользователей. Избежать указанных неудобств позволяет алгоритм обслуживания, осуществляющий разделение времени процессора (что реализовано в системе АОС-РРВ). При этом запросы, требующие больших затрат времени, не вызывают задержки в работе других пользователей, поскольку время процессора распределяется равномерно, независимо от того, закончилась обработка предшествующего запроса или нет. РДТ — одна из наиболее распространенных форм диалогового взаимодействия, реализуемых средствами АОС.

М.В.Легкий.

РЕЖИМ МУЛЬТИПРОГРАММНЫЙ — режим работы *электронной вычислительной машины*, при котором рабочее время ее процессора распределяется между несколькими программами. В многопроцессорной ЭВМ несколько процессоров могут одновременно выполнять несколько программ. Но если программ больше, чем процессоров, то и в многопроцессорной ЭВМ происходит распределение времени процессоров между программами. Р.м. позволяет одной ЭВМ обслуживать несколько терминалов, в частности *рабочих мест обучаемых* (см. *Режим разделения времени*), и экономить время работы ЭВМ (пока одна программа загружается в оперативное запоминающее устройство или ожидает данные из периферийных устройств, выполняется другая).

П.Горват, В.И.Леонтьев.

РЕЖИМ РАЗДЕЛЕНИЯ ВРЕМЕНИ — режим мультипрограммный, при котором многие одновременно работающие пользователи общаются с ЭВМ с помощью терминалов. Программа каждого пользователя выполняется по частям. После выполнения части первой программы выполняется часть второй программы, далее третьей и т.д. Возвращение к каждой программе происходит после обслуживания всех остальных. Промежуток времени, выделенный на выполнение части программы, выбирается такой длительности, чтобы у каждого пользователя была иллюзия единичного использования ЭВМ, т.е. время между двумя обращениями к одной программе должно быть меньше промежутка времени, вызывающего у пользователя ощущение ожидания (обычно менее 0,25 с).

В.И.Леонтьев.

РЕЖИМ РАБОТЫ С ВИДЕОТЕРМИНАЛОМ — установленный порядок труда и отдыха при пользовании видеотерминалом (см. *Дисплей*). Связан с характером деятельности обучаемого — пользователя компьютера, типом выполняемых им заданий, качеством

технических и программных средств, организацией *рабочего места*, оснащенного видеотерминалом. В частности, типы работы с видеотерминалом различаются по многим аспектам, среди которых — визуальный акцент экрана (время и характер наблюдения за изображением в течение выполнения задания).

Оптимальный Р.р. с в. зависит от конкретных психолого-педагогических задач *компьютеризации* обучения: 1) изучения основ информатики как специального предмета (компьютер — объект обучения); 2) использования компьютера в качестве средства обучения различным учебным предметам. Ныне наиболее разработаны рекомендации по применению компьютеров для изучения основ информатики. Поскольку наиболее распространены компьютеры с экранами на электронно-лучевых трубках, с низким *качеством изображения*, с рядом ограничений по физическим факторам, не оптимальным решением вопросов организации рабочих мест и рабочих помещений, нормирование длительности работы обучаемого с компьютером имеет большое значение. Продолжительность работы с видеотерминалом является одним из осн. факторов возникновения общего утомления. С целью его предотвращения в работе пользователя рекомендуются перерывы различной длительности.

В результате изучения психофизиологического состояния обучаемых — пользователей компьютеров в возрасте 15—18 лет установлено, что можно рекомендовать проведение занятий с использованием видеотерминала три раза по 45 мин с перерывами по 15 мин без риска нарушения их здоровья. Вместе с тем целесообразно резко ограничить количество и продолжительность учебных занятий с использованием видеотерминала. В частности, рекомендуется режим учебных занятий и отдыха в кабинетах информатики и вычислительной техники, который предусматривает: соблюдение регламентированной длительности непрерывной работы за дисплеем; соблюдение регламентированных перерывов и активное их проведение; рациональное распределение уроков информатики в расписании учебных занятий; организацию правильного контроля за состоянием здоровья обучаемых в средних школах. На практические занятия непосредственно с компьютером учащиеся старших классов должны затрачивать 50% учебного времени. Рекомендуется непрерывная продолжительность работы за экраном дисплея для учащихся старших классов не более 20 мин. Для сохранения высокой учебной активности и работоспособности обучаемых рекомендуется следующая структура практических занятий по основам информатики: 20 мин работы за дисплеем; 5 мин активного отдыха; 20 мин работы за дисплеем. Работу за экраном рекомендуется проводить в свободном индивидуальном ритме. Во время активного отдыха целесообразны специальные упражнения, снимающие *утомление зрительное*. В перерывы после урока информатики в течение 10 мин рекомендуется выполнять физические упражнения для снятия позостатического

утомления. Рекомендуется уроки информатики и вычислительной техники включать в расписание учебного дня не далее 4-го урока. После урока информатики и вычислительной техники целесообразно проводить уроки физического воспитания. Рекомендуется также систематически проводить медицинские осмотры обучаемых.

В.М.Бондаровская, А.С.Коваленко.

РЕФРАКЦИЯ (лат. *refractio* — преломление) — преломление света в оптической системе. Понятие *R.* глаза включает физическую *R.*, характеризующую преломляющую силу оптической системы глаза, и клиническую *R.*, характеризующую положение главного фокуса оптической системы глаза по отношению к сетчатой оболочке. Если главный фокус совпадает с сетчатой оболочкой, то образуется соразмерная *R.* (эмметропия), если не совпадает — аметропия (гиперметропия — дальнозоркость, *миопия* — близорукость). В зависимости от формы оптических структур глаза различают сферическую *R.* глаза, когда преломление лучей в глазу одинаково во всех меридианах, и асферическую *R.* глаза — *астигматизм*.

А.С.Коваленко, В.Г.Мартырозова.

РЕЧЕВАЯ КОММУНИКАЦИЯ — способ *взаимодействия* *человеко-машинного* с использованием человеческой речи в качестве основного средства обмена *информацией*. Обеспечивается на основе добавления к базовым техническим средствам компьютера речевого ввода и речевого вывода. Главным преимуществами использования речевого ввода являются независимость от уровня зрения, возможность использования при одновременной другой деятельности обслуживания, скорость и простота обслуживания. Недостаток использования речевого ввода — высокая стоимость. Изготавливаемые ныне системы автоматического опознавания с распознаванием речи работают с ограниченным *словарем* объемом до 1000 слов с уровнем точности опознавания до 95%. В промышленности они используются в качестве опознавателей *команд*.

П.Горват.

РЕЧЕВОЙ ДИАЛОГ — *устное взаимодействие* *человеко-машинное*. Происходит с помощью систем речевого диалога (СРД), осуществляющих автоматическое распознавание речи (речевой ввод *информации*) и автоматический синтез речи (речевой вывод *информации*). Речевое взаимодействие наиболее удобно и привычно для человека и поэтому наиболее эффективно. Так, речевой ввод *данных* в 2—3 раза быстрее клавиатурного и значительно надежнее его. Речевой ввод — вывод информации высвобождает глаза и руки, создает благоприятные возможности для *обучения* (напр., уменьшает

утомляемость *обучаемых*) и тем самым расширяет возможности *компьютерной технологии обучения* (КТО).

Теор. основой создания СРД является речевая *информатика* — наука об автоматическом распознавании, понимании и синтезе речевых сигналов. Автоматическое распознавание речи — это процесс обработки речевого сигнала с целью указания последовательности слов, а понимание — с целью указания переданного смысла. Очевидно, что при автоматическом понимании (смысловой интерпретации) выполняется более сложный процесс обобщения речевой информации, поскольку один и тот же смысл можно передать различными последовательностями слов. При автоматическом распознавании и понимании речи осуществляется одновременно как преобразование сигнала с микрофона в *текст*, так и его осмысление. Это преобразование сопровождается сжатием речевой информации, которая всегда избыточна, поскольку несет сведения не только о том, что сказано, но и кто говорит, каково функциональное и эмоциональное состояние говорящего, каковы темп и громкость речи и т.п. Не следует полагать, что нужно отказаться от автоматического понимания речи, ограничившись лишь распознаванием: сначала необходимо преобразовать речь в текст — последовательность слов, а затем осмысливать этот текст приемами автоматической интерпретации текстов. Нельзя надежно преобразовать речь в текст без привлечения синтаксиса, семантики и прагматики языка *диалога*; следовательно, автоматическое распознавание и понимание речи должны выполняться в едином взаимосвязанном процессе. Проблему автоматического распознавания и понимания речи можно решить, если будет создана автоматическая машинка, печатающая и редактирующая тексты под диктовку. Ныне реальны СРД, оперирующие со *словарем* до нескольких тысяч слов и выполняющие распознавание и смысловую интерпретацию речи в рамках конкретных *предметных областей* (ПО). Такие возможности СРД достаточны для организации Р.д. в КТО.

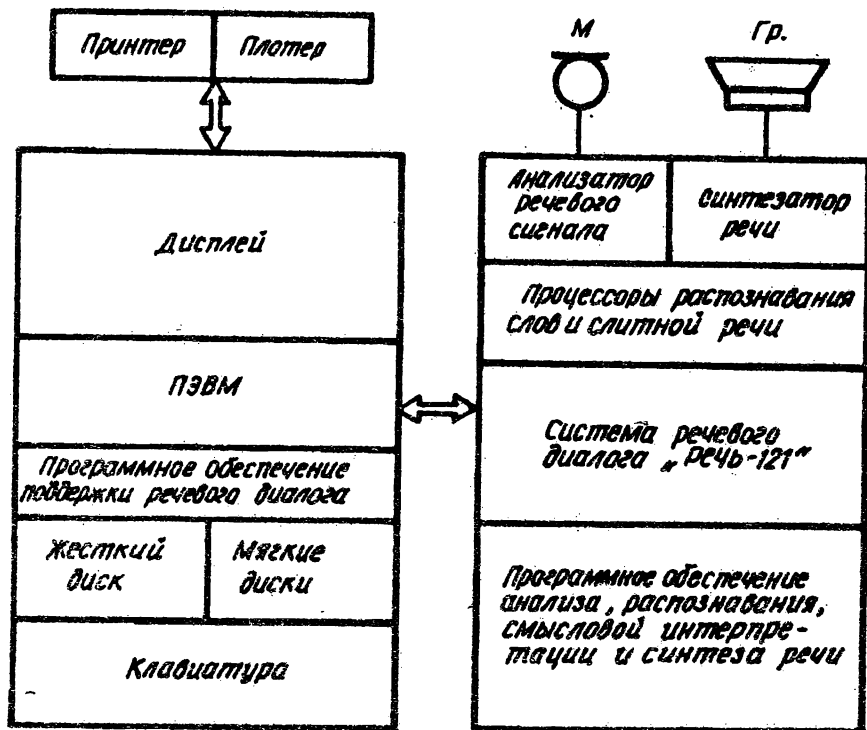
При создании СРД для КТО в качестве теор. основы можно взять методы распознавания отдельно произносимых слов и методы распознавания и смысловой интерпретации дискретной (с паузами между словами) и слитной (естественной) речи, составляемой из словаря в 1000 и более слов, при условии настройки системы распознавания речи на словарь и голос диктора. Такие методы позволяют обеспечить Р.д. на формализованных языках или *естественных языках ограниченных*. При этом для диалога может выбираться любой индоевропейский язык.

Автоматический синтез речи выступает как обратная проблема по отношению к автоматическому распознаванию — не человек говорит, а компьютер распознает, а наоборот, машина воспроизводит речь, а человек ее воспринимает. Если при автоматическом распознавании речи стремятся избавиться от меняющихся факторов,

определяющих разнообразие речевых сигналов, то при синтезе речи, наоборот, управляя меняющимися факторами, придают синтезированной речи те или иные индивидуальные и эмоциональные особенности, управляют темпом и интенсивностью произнесения, моделируют коартикуляцию и редукцию звуков, влияют на разборчивость, качество и натуральность синтетической речи. Различают "магнитофонный" синтез речи — "склеивание" речи из экономных записей отдельных слов или фраз, хранящихся в памяти ЭВМ, и фонемный синтез речи, при котором озвучиваются (синтезируется речь) произвольные цифробуквенные тексты. В свою очередь, различают простой фонемный синтез речи, при котором озвучиваются слабо размеченные тексты (обыкновенные орфографические тексты с указанием ударений в словах и дополнительных, кроме точек, запятых, знаков вопроса и т.д., интонационных знаков), и синтез осмысленной речи. В последнем случае компьютер сам составляет тексты, выражающие определенный смысл, размечает их и осуществляет синтез речи. Ныне можно считать решенной в первом приближении задачу автоматического синтеза речи на русском, украинском, белорусском, английском, японском, французском и др. языках. Для многих языков предстоит еще создавать соответствующие базы знаний, чтобы был возможен автоматический синтез речи на этих языках. Автоматически синтезируемая речь имеет выраженный "машинный акцент", приемлемую словесную разборчивость — не менее 95%. Однако предстоит еще значительный объем работ по совершенствованию синтеза речи, повышению качества и натуральности звучания, управлению индивидуальными особенностями голоса, обеспечению многоязычного синтеза речи.

СРД для КТО создаются в двух вариантах: в виде модулей (плат) распознавания и синтеза речи, встраиваемых в персональную электронную вычислительную машину (ПЭВМ), и в виде автономных СРД, подключаемых к ПЭВМ. В обоих случаях для ПЭВМ создаются программные средства поддержки Р.д. Как правило, возможности Р.д. для автономных СРД более широкие. Составителю учебных программ предоставляется набор макрооператоров ПЭВМ, среди которых базовыми являются РАСПОЗНАЙ КОМАНДУ ИЗ СЛОВАРЯ (А) и СИНТЕЗИРУЙ ФРАЗУ (В), где А — имя словаря, В — синтезируемый текст.

Для объяснения технологии составления учебных программ и порядка работы с СРД при КТО рассмотрим устройство и принцип работы модели "Речь-121", разработанной в 1986 в Институте кибернетики им. В.М.Глушкова АН Украины. Автоматизированные рабочие места обучаемого (рис.), составителя учебных программ и преподавателя состоят из ПЭВМ и СРД "Речь-121". В СРД встроена микро-ЭВМ "Электроника-60" (или ее одноплатный аналог "Электроника-МС 1201.02"), которая управляет процессами, происходящими в системе. Она же обеспечивает внутренние и внешние интерфейсы,



в частности с ПЭВМ. К каналу микро-ЭВМ подключены четыре специализированных процессора: первичного анализа речевого сигнала, синтеза речи, вычислителя элементарных мер сходства и распознавателя речи. Распознаватель речи — параллельная вычислительная машина, содержащая три однотипных процессора распознавания речи, каждый из которых имеет свою локальную память и управление которыми осуществляется от одного общего контроллера. Локальная память процессоров доступна и управляющей микро-ЭВМ. Архитектура параллельной машины распознавания речи (см. *Архитектура ЭВМ*) основана на принципе: один адрес, много инструкций и много данных. Один процессор распознавания обслуживает 256 слов. Увеличение объема словаря достигается простым наращиванием процессоров. Осн. технические характеристики СРД "Речь-121": возможность распознавания отдельно произносимых слов и слитной речи, составляемой из слов выбранного словаря; объем словаря — 768 слов; задержка выдачи ответа распознавания — 0,3 с (не зависит от продолжительности слова или фразы); макс. продолжительность распознаваемого отрезка речевого сигнала — 15 с;

надежность распознавания отдельно произносимых слов (при объеме словаря 200 слов) — 99%; надежность распознавания слов в потоке слитной речи при свободном порядке следования слов — 93%; объем словаря при синтезе речи — не ограниченный; словесная разборчивость синтезированной речи — 98%; возможность распознавания и синтеза многоязычной речи.

Программное обеспечение СРД состоит из системной и проблемно-ориентированной частей. Системная часть служит для организации диалогов и связи СРД с ПЭВМ, проблемно-ориентированная обслуживает специализированные речевые процессоры, организует прием, обработку, распознавание, синтез и выдачу речевого сигнала. Для обеспечения управления работой СРД расширена операционная система ПЭВМ — созданы системные средства поддержки Р.д. со стороны ПЭВМ. Эти средства обеспечивают формирование и передачу текста для синтеза речи в СРД, формирование файлов эталонов слов, обмен этими файлами с СРД, загрузку рабочего словаря в СРД, перенастройку СРД на новую ПО и т.п. Проблемно-ориентированная часть программного обеспечения ПЭВМ организует Р.д.

Сведения о ПО, в рамках которой реализуется Р.д., готовятся заранее составителем-программистом учебной программы. Эти сведения содержатся на жестком диске ПЭВМ или загружаются в ПЭВМ с дискеты. Каждой паре (ПО плюс диктор-оператор — обучаемый или преподаватель) соответствует так наз. речевой файл, содержащий эталоны фонем и слов конкретного диктора для данной ПО. Для словаря из 600 слов речевой файл занимает всего 20 К памяти. Диктор, приступающий к работе с СРД, на клавиатуре ПЭВМ набирает имя ПО (тему урока) и свое имя, после чего его речевой файл загружается в СРД (время загрузки около 1 с). Сообщение о готовности СРД к работе высвечивается на экране дисплея и объявляется голосом. Если обнаруживается, что для данных ПО и диктора нет речевого файла, ПЭВМ предлагает диктору представиться СРД (обучение распознаванию речи) — повторять (произносить) слова, которые высвечиваются на экране и/или синтезируются СРД. Скорость обучения на словарь составляет в среднем 30 слов в минуту при однократном произношении каждого слова и уменьшается в n раз при n -кратном произнесении каждого слова (количество n выбирает пользователь, отвечая на вопрос ПЭВМ). После окончания произнесения обучающей выборки формируется речевой файл для данной ПО и диктора; этот файл одновременно запоминается на жестком диске или дискете.

Осн. режимы работы СРД: ОБУЧЕНИЕ РАСПОЗНАВАНИЮ ФОНЕМ — диктор произносит в микрофон слова, предлагаемые ПЭВМ, а СРД вычисляет элементы фонем и передает их в ПЭВМ; ОБУЧЕНИЕ РАСПОЗНАВАНИЮ СЛОВ — по указанию ПЭВМ диктор одно- или многократно произносит слово, а СРД формирует

эталон этого слова и заносит его в ПЭВМ; РАСПОЗНАВАНИЕ СЛОВА — ПЭВМ задает подсловарь, а СРД распознает произнесенное слово как принадлежащее заданному подсловарю и передает результат распознавания в ПЭВМ; РАСПОЗНАВАНИЕ ФРАЗЫ — ПЭВМ задает структуру фразы, СРД распознает произнесенную фразу и результат распознавания передает в ПЭВМ; КОРРЕКТИРОВКА или ЗАМЕНА СЛОВА — реализуется обучение распознаванию заданного ПЭВМ слова; СИНТЕЗ РЕЧИ — ПЭВМ передает фонемный текст, а СРД озвучивает его; СИНТЕЗ РЕЧИ С КЛАВИАТУРЫ — оператор набирает текст с помощью клавиатуры, а СРД озвучивает его; ПОВТОРЕНИЕ ПОСЛЕДНЕГО ПРОИЗНЕСЕНИЯ — повторно реализуется синтез последнего произнесения слова или фразы; ЗАГРУЗКА ЭТАЛОНОВ ФОНЕМ И СЛОВ — речевой файл для данной ПО и диктора загружается в СРД. Кроме перечисленных, имеются служебные режимы работы, позволяющие контролировать правильность работы процессоров анализа, распознавания и синтеза речи. Каждый из перечисленных режимов обеспечивается соответствующей макрофункцией, включенной в системные средства поддержки Р.д. на ПЭВМ. Режимы распознавания имеют ряд модификаций, зависящих от того, озвучивается ли результат распознавания и на каком дисплее находится результат распознавания. Средства ПЭВМ, поддерживающие Р.д., погружаются в программное обеспечение составителя учебных программ. Ему предоставляются все макрофункции СРД, используя которые он вызывает распознавание или синтез речи согласно сценарию урока или лекции. При этом составитель по своему усмотрению организует устные подсказки обучаемому, указывающие, что делать, какие клавиши нажимать и что говорить в микрофон, сопровождая подсказки высвечиванием текстов на экране дисплея. Используя макрофункции СРД, высвечивание и озвучивание результатов распознавания, повторение произнесения команд, составитель учебной программы организует достоверное восприятие устных команд, наиболее приемлемое сочетание всех средств диалога — клавиатурного и речевого ввода информации, вывода информации голосом и на дисплей. Выбранная структура СРД "Речь-121" и разделение функций между ПЭВМ и СРД одинаково удобны при комплексировании СРД линии "Речь" с ПЭВМ типа IBM XT, IBM AT, ЕС-1840, ЕС-1841, "Нейрон", "Электроника -85" и др.

Применение СРД радикально изменяет, обогащает и повышает эффективность КТО. Одновременно СРД представляют качественно новые возможности в КТО: обучение родному и иностранному языку, обучение слепых и глухих, инвалидов. Кроме КТО, СРД эффективно используются в других человек-машинных комплексах и технологиях сбора, обработки информации и управления.

Т.К.Винцюк.

РЕШАТЕЛЬ, решающая система — активная система, осуществляющая решение *задач*. Наряду с реальными Р. (людьми, техническими и человеко-машинными системами) в теории задач рассматриваются идеализированные Р., вводимые в рассмотрение как системы четко охарактеризованных *средств решения задач* (в частности, логических и математических *операторов*).

Г.А.Балл.

РЕШЕНИЕ ЗАДАЧ С ПОМОЩЬЮ КОМПЬЮТЕРА — одна из главных практических целей компьютерного всеобуча, с учетом которой строится все содержание обучения *компьютерной грамотности*. Диапазон решаемых с помощью компьютера задач чрезвычайно широк. Один из критериев для их классификации — деление задач на два больших класса: учебные и практические. В первом случае цель состоит в том, чтобы научиться решать соответствующие задачи, усвоить обобщенный способ действия по их решению (см. *Компьютерное обучение*), во втором — речь идет о необходимости решить определенную, конкретную практическую задачу. Среди практических задач в свою очередь выделяют *диагностические, изобретательские, прогностические* и др. задачи.

Процессу решения любой задачи человеком предшествует ее постановка. Этап постановки обязательно требует рассмотрения применительно ко всем задачам и прежде всего к возникающим в практической трудовой деятельности. Обнаружение будущей задачи — проблемной ситуации (см. *Проблемное обучение*) — одно из важнейших качеств мышления практика-профессионала. Задачная (проблемная) ситуация чаще всего не осознается полностью и существует постольку, поскольку, образно говоря, в ней “присутствует” обнаруживший ее человек. Задача отличается от задачной ситуации тем, что она осознана субъектом, объективирована и чаще всего описана словесно либо в иной знаковой (числовой, графической) форме. При постановке задачи с целью компьютерного решения можно использовать либо знаковую форму — вербальную или графическую, формульную или числовую, либо комбинированную в виде описания структуры начальной и конечной ситуаций в некоторой среде (для роботов), либо как технологическую карту или чертеж деталей, которые надо изготовить на станке с числовым программным управлением, а также в любой другой форме. Постановка задачи содержит ее анализ и понимание на некотором определенном уровне, предварительную наметку замысла решения, стратегии, плана действий. Она предусматривает учет возможных реально доступных и имеющихся в наличии *средств решения задач* (см. *Обучение*).

В компьютерной системе выделяют три группы средств решения задач. Первая из них — средства построения *модели задачи* (или, иначе, средства формализованного описания условия задачи), пред-

ставляющие собой наборы определенных форм представления данных и связей между ними — таблиц, списков, форм и др. В упорядоченную модель задачи входят описанные в знаковой форме объекты, их функции, цель. Построение модели задачи всегда следует за этапом ее постановки. При Р.э. с п.к. этот процесс осуществляется осознанно и целенаправленно и заканчивается формальным или формализованным описанием задачей структуры. Вторая группа компьютерных средств — сервисные средства, освобождающие пользователя от работы с машинной атрибутикой, контроля за правильностью синтаксических конструкций и др. Третья группа компьютерных средств (*языки программирования, ЯП*) соответствует центральному моменту процесса решения, состоящему в постепенном сведении имеющегося плана, последовательности решения, его стратегического направления путем постепенной детализации к алгоритму решения, а затем к имеющимся программным средствам. При решении практической задачи с помощью компьютера важно уметь выбрать адекватное программное обеспечение, наиболее подходящий для данной задачи ЯП (параметры выбора многообразны — тип задачи, структура данных, процедуры языка, их масштаб и содержание и т.д.). Указанные психологические принципы Р.э.с п.к. легли в основу разработки системы решения задач обработки данных РЯОД (расширенный язык обработки данных). Она имеет все три набора средств — сервисные, специальные средства формализованного описания условия задачи (уточнения условия задачи обработки данных), специализированный ЯП и соответствующее учебное обеспечение — систему рекомендаций по уточнению условия задачи и по построению алгоритмов на РЯОДе. Эти средства введены с целью подготовки пользователя к решению задач данного класса.

М.Л.Смольсон.

Р0БКО — комплексный кабинет обучения робототехнике и компьютерной технике. Существует несколько типов таких кабинетов Р0БКО-А, Р0БКО-В, Р0БКО-Е, Р0БКО-Н и др. Каждый из них ориентирован на конкретную стадию обучения — от первоначального ознакомления с компьютерной техникой до систем проектирования и управления технологическими процессами на макетных моделях. Р0БКО-А включает рабочее место преподавателя и шесть рабочих мест обучаемых. Каждое рабочее место оборудуется учебным робототехническим устройством. В ходе учебного процесса обучаемые должны научиться подключать макеты промышленных роботов к компьютеру, составлять и отлаживать управляющие программы. Р0БКО-В дополнительно включает макеты обрабатывающих станков (фрезерного, токарного) и графопостроитель для изучения средств проектирования технологических процессов. Р0БКО-Е

представляет собой кабинет для обучения механике. Он оснащается управляемыми макетными моделями фирмы ФИШЕР (Австрия). Обучаемые закрепляют приобретенные ранее знания и навыки в области *программирования*, проектирования, сборки, использования датчиков различного типа, построения системы управления. РОБКО-Н — кабинет обучения в области новых технологий, использующих последние достижения компьютерной техники. Обучаемые на базе кабинета РОБКО-Н изучают и осваивают принципиально новые способы и формы взаимодействия с вычислительной техникой (речевая, визуальная), новые формы и способы передачи данных (фибрионтика). Обеспечивается изучение систем обработки *текстов* и систем автоматизированного проектирования. Все кабинеты оснащаются видеоаппаратурой, диа- и шрайбпроекторами, специальной мебелью, учитывающей эргономические особенности *учебной деятельности* континентов обучаемых, а также изучаемых процессов и технологий.

Одна из осн. целей включения кабинетов РОБКО в учебный процесс — изучение средств и технологий контролирования текущих состояний объектов в промышленных системах. Обучение строится на основе целенаправленной связности различных стадий, этапов и фаз и преследует следующие цели: формирование представлений об истории развития робототехники; приобретение начальных знаний и навыков о механических системах в робототехнике и методиках проектирования и моделирования производств на их основе; систематизация знаний по *информатике*; формирование осн. понятий робототехники — “рабочая среда робота” и др.; изучение осн. технических устройств типа “сенсоры” и т.п., освоение осн. принципов их использования; освоение осн. принципов комплексного применения робототехнических комплексов применительно к профилю различных производств; приобретение технологических и конструкторских знаний по новым информационным технологиям в промышленности. Кабинеты со структурой, подобной РОБКО, представляют собой достаточно развитую *учебную среду* открытого типа. В их структуру можно достаточно просто включить новое оборудование и увязать его применение с целями педагогического процесса.

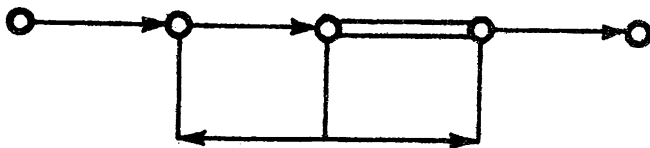
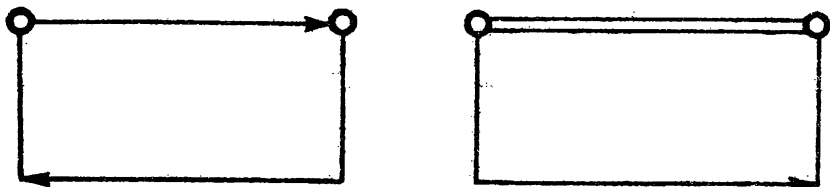
*А.Ангелов, К.Белов, В.Георчев, И.Далбоков,
В.Добринов, К.Томов, Н.Шваров, К.Янакиев.*

Р-ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ — *технология программирования*, основанная на промышленной концепции многоконтурного и многоуровневого *проектирования* программных средств методом поэтапной конкретизации любых неформальных *понятий* в алгебре графических структур, обеспечивающей “сборочный” стиль *программирования*, многократную применимость и новые принципы

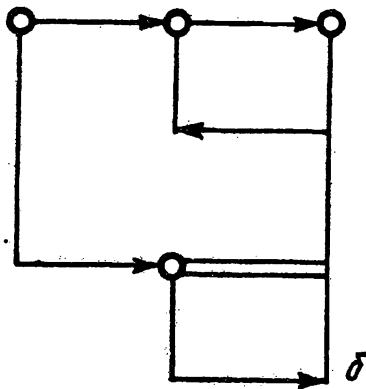
фондирования программных средств; единой, не зависящей от языков программирования (ЯП), графической форме представления алгоритмов, программ, данных и процессов на всех стадиях и этапах их разработки; системе автоматизации организационной деятельности коллектива, обеспечивающей дружественный интерфейс заказчиков, постановщиков задачи, руководителей проекта, математиков-алгоритмистов, программистов и др. специалистов в процессе разработки программного проекта и "отторжение" его от разработчиков в конце работы. Разработана в Институте кибернетики им. В.М.Глушкова АН Украины и развивается в Международном научном центре технологии программирования ТЕХНОСОFT (Киев). В основе Р-т.п. лежат результаты фундаментальных теор. исследований по теории автоматов, методу формализованных технических заданий и безбумажной информатике. Предназначена для разработки широкого класса программных средств, в т.ч. структурно и логически сложного программного обеспечения во всех областях применения вычислительной техники.

Для Р-т.п. разработаны технологические комплексы РТК, автоматизирующие графический стиль программирования практически на всех этапах программного проекта — от постановки задачи, системного анализа и модельного проектирования до его конструирования, отладки, документирования и эксплуатации пользователем. Комплексы РТК реализованы на многих вычислительных машинах широкого применения и поддерживают графический стиль программирования в штатных операционных системах со всеми распространенными ЯП: Фортран, ПЛ/1, Паскаль, Си, Модула-2 и др. В алгебре конструирования графических программ используются две графические структуры и операции последовательного (рис.а), вложенного (рис.б) и параллельного (рис.в) их соединения. Из графических структур (Р-структур) с помощью операций над ними создаются конструкции, наз. Р-схемами. На дугах Р-структур могут быть записи произвольного вида и длины (формулы, операторы ЯП, естественноречевой текст, рисунки, таблицы и т. п.), расположенные в одну или более строк. Записи над дугами Р-схем имеют смысл условия прохождения по дуге, а записи под дугами — выполняемого при этом действия.

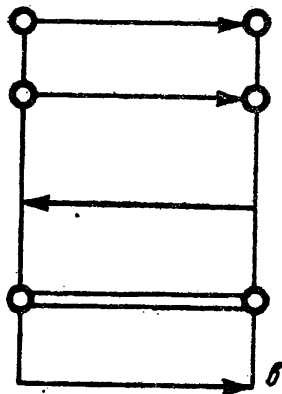
Технологический модуль Р-т.п. состоит из главной и обеспечивающей (операционное и информационное поля) частей. Пользовательский интерфейс определяется главной частью. Операционная часть определяет все допустимые действия специалиста в интерфейсной видимой части модуля, а также связь ее с информационной частью и др. модулями. Информационная часть — это локальная база знаний модуля или то, что определяет комфорт специалиста при работе с модулем: подсказки, HELP- тексты, учебные тексты, паспортные данные, словари проектирования, аксиомы вывода и



a



б



в

правила продукций (см. *Продукционные методы представления знаний*) локальной предметной области (ПО), меню и т.д.

Наиболее развитая и интегрированная форма *представления знаний* — чертеж. Он стандартизует описание любой ПО, привязывает его к схемам рациональной промышленной реализации, делает это описание одинаковым по оформлению. Чертеж состоит из четырех полей: имя чертежа, поле спецификаций, рабочее поле, поле абстракций. Чертежи связаны друг с другом иерархически через механизм раскрытия любого обозначения или понятия на чертеже. Помимо иерархии, чертежи можно объединить в контуры, связь между которыми не является жесткой, алгоритмической, а скорее информационной. Контуры проекта могут быть самыми разнообраз-

ными, напр., информационный контур заказчика (техническое задание), моделирующий, алгоритмический, учебный и т.д.

Технологический процесс в Р-т.п. формируется на этапе технологической подготовки работ из технологических модулей, соответственно запрограммированных. Этим достигается гибкость процесса в Р-т.п., обеспечивающая эффективную ее настройку на конкретные условия конкретного коллектива.

И.В.Вельбицкий, В.Д.Доронин.

САКИО — *экспертная обучающая система*, предназначенная для обучения пользователей написанию простейших логических программ. Использует две стратегии обучения: 1) от описания задачи к изучению понятий, связанных с этой задачей; 2) от изучения понятий предмета обучения к унификации этих понятий и построению описания решаемой задачи. Дает возможность: построить простейшую логическую программу; изучить язык логического программирования.

Если обучаемый правильно формализует предметную область (ПО) или рассматриваемую задачу, то он может написать логическую программу. Поэтому в основе обучения пользователя построению простейших логических программ лежит обучение формальному описанию ПО (задачи). Обучаемый должен научиться описывать свою задачу, используя утверждения об объектах и отношениях. Это первая профессиональная цель для обучаемого. Для написания конкретной логической программы, напр., на языке *МПролог*, необходимо знать осн. синтаксические конструкции языка. Изучение этих конструкций — вторая профессиональная цель для обучаемого. Исходя из этих двух целей, обучаемый должен научиться: словесно описывать заданную ПО; структурировать ПО (описывать в виде объектов, свойств, отношений); логически описывать структурированную ПО в виде аксиом логики предикатов первого порядка; писать простейшие *Пролог-программы*; уметь описывать утверждения Пролог-программы в виде синтаксических конструкций языка логического программирования.

С. обеспечивает индивидуальное обучение различных категорий пользователей на основе *модели обучаемого*, стратегий и методик обучения. Модель обучаемого содержит четыре уровня *информации*: локальный уровень, содержащий информацию о выполнении текущего задания и используемый для построения сценария обучения в рамках данной темы; текущий уровень, содержащий информацию об изучении данным пользователем текущего урока и используемый для выбора очередного урока для изучения или выбора материала для повторения; глобальный уровень, содержащий усредненные характеристики всего процесса обучения и последовательность прохождения обучаемым материала курса и используемый для построения сценария обучения;

психологический уровень, содержащий результаты априорного (доопытного) тестирования.

С. работает в режимах: априорного тестирования, справки, обучения, консультации и экзамена. Режим априорного тестирования используется для определения индивидуальных характеристик обучаемого, построения модели обучаемого, выбора наиболее целесообразной стратегии, методики и модели процесса обучения. Режим справки предназначен для получения необходимой информации о понятиях. Режим обучения используется для обучения методам построения простых логических программ и осн. понятиям языка логического программирования. Режим консультации применяется тогда, когда обучаемый приходит со своей задачей и хочет проверить предлагаемое решение или получить помощь. Режим экзамена обеспечивает контроль и оценку уровня знаний пользователя по результатам *ответов* на предложенные *вопросы* и решения ряда задач. Естественный языковой *интерфейс* в С. включает реализацию информационного *запроса*, описания потребности обучаемого, ответов на вопросы системы; описания задачи и анализ связанного текста. С. реализована на ПЭВМ типа IBM PC. Языки реализации Турбоси и Турбопролог.

Фрагмент *диалога* в режиме обучения (С — САКИО, П — пользователь—обучаемый):

С: Приведите пример описания предметной области о расположении предметов (КУБ, ШАР, ...) относительно друг друга. Сначала определите предметы (объекты).

П: КУБ, ШАР, МАЛЕНЬКИЙ КУБ.

С: Определите свойства объектов.

П: СИНИЙ КУБ, БОЛЬШОЙ ШАР.

С: Есть ли информация о свойствах МАЛЕНЬКОГО КУБА?

П: Нет.

С: Определите расположение объектов относительно друг друга.

П: КУБ справа от ШАРА.

С: Какой КУБ?

П: КРАСНЫЙ.

С: Такого объекта нет. Ответьте еще раз.

П: МАЛЕНЬКИЙ КУБ.

С: Есть ли информация об объекте: СИНИЙ КУБ?

П: Нет.

С: Определите правило перемещения объектов в виде:
ЕСЛИ ... ТО

П: ЕСЛИ СИНИЙ КУБ слева от ШАРА, ТО переместить ШАР влево.

С: Для данной предметной области нельзя установить истинность

посылки правила. Хотите ли Вы добавить сведения о расположении объектов друг относительно друга.

П: Нет.

В.В.Колос, С.П.Кудряцева.

САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА — внеаудиторная работа, осуществляемая без прямого контакта с преподавателем. Рассматривается как управляемый процесс. Современная теория управления учебно-воспитательным процессом опирается на применение кибернетики как общей теории управления процессом обучения, в частности на *программированное обучение*. Несмотря на то что программированное обучение как практический метод не было успешным, в педагогику оно внесло значительный вклад. В его теории управление рассматривается как один из главных детерминантов процесса учения. Уже сама деятельность *обучаемого* является программированной. Однако путем *программирования* эту деятельность можно сделать непосредственно управляемой (без присутствия преподавателя), т.к. посредством управления, базирующегося на *программе*, в процесс учения входят опыт и знания преподавателя. Управляющий и управляемый элементы являются составными частями единой развивающейся системы, т.е. студент, выступая в роли “управляемого элемента”, представляет собой не только объект, но и субъект учебно-воспитательного процесса. Чем чаще реализуется *обратная связь в обучении*, тем эффективнее и быстрее можно достичь поставленной цели обучения. Контроль или самоконтроль осуществляется по каждому шагу *учебной деятельности*, рекомендованной или запрограммированной преподавателем студенту для достижения конкретной цели обучения для данного шага. Приведем модель учебно-воспитательного процесса как управляемого процесса. Автор модели выделяет несколько фаз управления педагогическим процессом: определение целевых параметров и условий; выявление начального состояния субъекта управления; план учения и проект обучения; информация о состоянии и протекании управляемого учения; корректирующее воздействие; информация о конечном результате управляемого процесса. Управление С.р.с. на уровне предмета тесно связано с тенденцией обеспечения осн. активной роли студента. С проблемой активности неразрывно связано и требование индивидуализации управления процессом учения. Система управления С.р.с. на уровне предмета опирается на следующие основные принципы: а) расчленение всего учебного материала предмета на отдельные учебные единицы (УЕ); б) определение обучающих целей предмета и УЕ; в) управление С.р.с. с помощью письменных материалов; г) систематическая обратная связь, предполагающая самоконтроль, а также контроль со стороны преподавателя; д) полное достижение соответствующих обучающих целей каждой УЕ. Кроме

пяти основных, применяют еще два дополнительных принципа, которые повышают действенность системы, а в случае необходимости исключают опасность односторонних ее воздействий. Это принципы самостоятельности и творчества, а также индивидуализации и дифференциации.

При расчленении материала на УЕ нужно учитывать не только критерии и содержание предмета, но и дидактические и методические критерии. Важный дополнительный критерий при расчленении учебного материала на УЕ — управление посредством обратной связи. УЕ является как тематической, так и оперативной единицей системы. В рамках УЕ протекает цикл управляемого обучения начиная с определения учебных целей УЕ и кончая оценкой результатов обучения преподавателем, напр., при помощи теста. Использование ЭВМ дает возможность применять более сложные и адаптивные формы управления С.р.с., реагирующие на его индивидуальные характеристики, достижения и недостатки. УЕ должна охватывать 1—2 недели работы по данному предмету.

В формулировку обучающей цели включают: требования к учебной деятельности студента; условия учебной деятельности студента (ее объем, способ осуществления, применение учебника, пособий); нормы учебной деятельности (определение миним. уровня, при достижении которого цель можно считать достигнутой). Общие цели изучения предмета конкретизируются в широкие цели изучения отдельных УЕ, а они, в свою очередь, разбиваются на более узкие — специфические цели УЕ, являющиеся исходным пунктом для подготовки дальнейших элементов управления С.р.с.

Письменные материалы, используемые при С.р.с., можно условно разделить на две части: учебные *тексты* (учебники, пособия и т.п.); методические указания для каждой УЕ. Методические указания рассматриваются как основное средство непрямого управления работой студента. В них материализуются и передаются студенту опыт преподавателя, его рекомендации, указания, выдвигаемые задачи. Указания разрабатываются для каждой УЕ. Структура методической инструкции для УЕ: краткое, мотивирующее введение; набор широких и более узких целей УЕ; предварительные требования к *знаниям* студента, необходимым для работы над УЕ; программа рекомендованной учебной деятельности, представляющая собой обучающее ядро письменной инструкции (содержит ссылки на литературу по УЕ, задачи и упражнения для ее освоения, а также другие методические указания по процессу обучения); автотест, дополненный ключом к результатам; информацию об условиях контрольного теста по материалу УЕ.

Стремление к систематическому внедрению обратной связи находит реализацию в каждом достаточно продуманном способе текущего контроля. Прямой контроль над каждой УЕ, осуществляемый преподавателем, предполагает наличие коррекционных прог-

рам учебной деятельности. Студенты, не выполнившие требования контроля, одновременно с результатами проверки или теста должны получить надлежащую информацию о том, какие части материала не освоены ими, что следует заново проработать или какие упражнения выполнить. Путем самоконтроля студент проверяет, насколько учебный материал им усвоен. Самоконтроль может быть связан с инструкцией, необходимой для проведения автокоррекции. Следовательно, студент имеет возможность исправлять в ходе учения допущенные ошибки. При достижении правильного результата он получает не только команду продолжать работу, но и оценку полученного результата.

Принцип “полного” овладения нужно рассматривать не как совершенное овладение конкретной научной дисциплиной, а как овладение комплексом конкретных учебных целей конкретных тем предмета обучения на заранее определенном уровне познавательных способностей. Критерием для определения требуемого уровня овладения отдельной обучающей целью является уровень подготовки студента, необходимый для понимания и изучения нового учебного материала. Проверку уровня овладения комплексом обучающих целей УЕ проводят с помощью теста. Принцип “полного” овладения проверяется более высоким критерием доли правильных ответов (от 70 до 100%), в отличие от критерия успеваемости при классических методах обучения.

Э.Мазак.

СВЕТОВОЕ ПЕРО — устройство, используемое для указания элемента изображения на экране дисплея. Преобразует излучение экрана в точке касания в электрический сигнал, подаваемый на управляющий электрод электронно-лучевой трубки. Если фиксировать концом С.п. к.-л. точку на экране, то эта точка будет высвечиваться. Передвигая С.п. от точки к точке, можно рисовать линии на экране. Переключая на С.п. специальный тумблер или кнопку, можно стирать засвеченные точки, а при движении пера - и целые линии. С.п. — удобное средство редактирования данных и ввода графической информации.

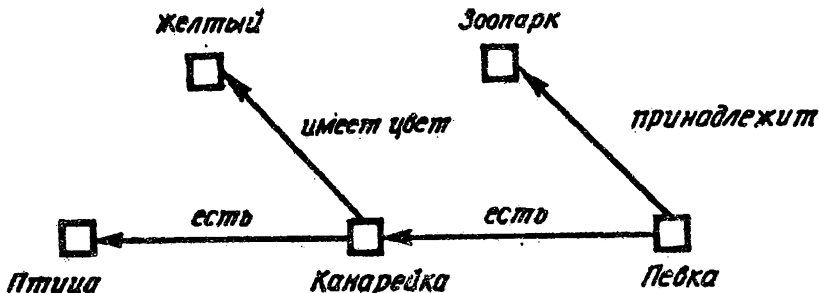
СВЕТОВОЙ ПОТОК — величина, оценивающая поток излучения, т.е. мощность оптического излучения по вызываемому им световому ощущению. Единица С.п. — люмен. 1 люмен равен С.п., создаваемому точечным источником света, расположенным в центре сферы радиусом 1 м и проходящему через отверстие указанной сферы площадью 1 м^2 .

С.п. — величина, выделенная из излучающего потока путем оценки излучения по его воздействию на селективный рецептор,

спектральная чувствительность которого определяется с помощью стандартных спектральных коэффициентов. При проектировании рабочих мест, оснащенных видеотерминалами на электронно-лучевых трубках, необходимо предусматривать устройства, ограничивающие С.п. от искусственных и естественного источников света.

С.А.Миرونченко, В.С.Пашковский.

СЕМАНТИЧЕСКАЯ СЕТЬ — модель декларативного представления знаний, состоящая из множества понятий и множества бинарных отношений между ними. Возникла и развивалась в 70-х гг. в целях моделирования ассоциативных механизмов рассуждения и вывода в системах искусственного интеллекта, поиска и доступа к данным в информационных системах. С.с. изображается графами с именованными вершинами — понятиями и именованными дугами — отношениями между понятиями. Каждое понятие предметной области сопоставляется с единственной вершиной. Дуги представляют отношения между понятиями (родо-видовые, принадлежности и др.). На рис. изображен иллюстративный фрагмент



С.с., из которого формальными методами можно извлечь информацию, т.е. получить, напр., следующие непосредственные заключения: канарейка есть птица; Певка есть канарейка; все канарейки желтые и т.п. Возможны более сложные заключения, требующие анализа окружения вершины, как, напр.: Певка есть желтая или Певка — птица. Существенно, что в С.с. знания о каждом понятии содержатся в структуре его окружения. В механизме анализа сети с целью извлечения информации фиксируется, как правило, порядок обхода соседних вершин. Именно так в С.с. пытаются разрешать противоречия.

Модели знаний в виде С.с. используются в экспериментальных автоматизированных обучающих системах. В сети системы BRAIN используется пять типов отношений: дефинитивные, факторные, атрибутивные, локативные и отношения близости. Учебный материал

в системе преобразуется в форму С.с. Структура сети позволяет генерировать *ответы на вопросы* и вопросы к *обучаемому*. Система ИТ строит С.с. *текста* учебного материала, в таком же виде представляет ответ обучаемого, сравнивает сети и сообщает о различиях. В системе MPS С.с. используется для анализа ответов сложной структуры, выявления семантических ошибок и организации содержательного обучающего диалога.

О.Н.Золотопуп, Р.Кавалюнос, В.Реклайтис.

СЕМАНТИЧЕСКИЙ АНАЛИЗ — построение семантического графа *текста*, в котором явно выражены смысловые отношения, существующие между словами и предложениями входного текста. Для этой цели выявляются слова, которые можно истолковать как *фреймы*, определяющие смысл предложения. В примере “Какие вычислительные машины имеются в московских институтах РАН?” такими словами-фреймами будет предикат “имеются” со слотами <что?> <где?> <у кого?> и слово “институты” с единственным слотом <какие?>. Часто на изображении семантического графа показывают фрагмент предметной области. Напр., во фрагменте семантической сети семантического графа выделены вершины-понятия: “ведомство”, “организация”, “учреждение”, “место”, “иметь”, “средство вычислительной техники” и “вычислительные машины”, а также показаны отношения между этими понятиями в виде дуг: “подчинение” и “расположение”. Они характеризуют *предметную область*. Отношение ISA (есть—некоторые) — родо-видовое, “агенса”, “лока” и “тема” — отношения лингвистической природы. Эти отношения характерны для глагола “имеется”. Наложение семантического графа на сеть позволяет осуществить семантическую интерпретацию *запроса* и в конечном итоге сформировать запрос на языке внутреннего представления.

Н.А.Власенко.

СИ — язык машинно-ориентированный системного программирования. Разработан в США в 1972—73. Более сложен в изучении и использовании, чем Паскаль и Бейсик, однако затраты на его освоение окупаются более высокой эффективностью получаемых программ. Ныне используется во многих областях, где традиционно применялся ассемблера язык. Все это позволяет рекомендовать Си в качестве инструментального языка для создания автоматизированных обучающих систем.

Синтаксис Си достаточно сложен, силен и, вместе с тем, краток. Он позволяет эффективно выражать самые разнообразные особенности реализации алгоритма и максимально учитывать архитектуру современных микро-ЭВМ и мини-ЭВМ. Однако неосторожное использование этой мощности часто приводит к уменьшению читабельности

программ и появлению различных “программистских трюков”. Си включает осн. базовые типы *данных* (целые и вещественные числа, *символы*) и широкий набор средств структурирования (массивы, структуры, объединения). *Пользователь* может определить свои типы данных при помощи описателя `typedef`. Оригинальна трактовка взаимосвязи между указателями и массивами (имя массива рассматривается как константа типа указателя, на указателях определены *операции* прибавления или вычитания целого числа, эквивалентные доступу к массиву по индексу, и т.д.). Учет архитектурных особенностей компьютера обусловил, напр., введение нескольких типов целых чисел (целое, короткое целое, длинное целое), которые соответствуют различным длинам машинных слов. Контроль за совместимостью типов и правильностью их использования в Си менее строг, чем в Паскале. Шире используются автоматические преобразования типов. Это дает программисту большую свободу, но может служить и источником ошибок.

Управляющие структуры позволяют писать ясные, хорошо структурированные программы. Введен ряд *операторов* (`break`, `continue`, `return`), позволяющих в большинстве случаев избежать употребления операторов безусловного перехода. Все *подпрограммы* в Си имеют вид функций. Головная программа также имеет вид функции с фиксированным именем `main`. Вложенные описания подпрограмм запрещены. Все параметры передаются только по значению. Если пользователь хочет изменить значение фактического параметра, он должен использовать указатели. Такой аппарат, с одной стороны, допускает эффективную реализацию, а с другой — обеспечивает достаточную безопасность. Особенностью Си является допущение переменного числа параметров функций. Контроль за соблюдением типов и числа параметров, как правило, не ведется, однако большинство реализаций содержат средства такого контроля.

Си допускает отдельную компиляцию функций. Единица компиляции — *файл*, содержащий определения одной или более функций. Имеются различные средства, позволяющие достичь хорошей модульности — наличие различных классов памяти (глобального, статистического, автоматического и регистрового) и так наз. файлов-заголовков. Сам язык не содержит средств ввода/вывода — они включаются в стандартные библиотеки. Такие библиотеки содержат как универсальные функции, так и функции, позволяющие максимально использовать особенности архитектуры конкретного компьютера и конкретной *операционной системы*. Характерная черта Си — наличие препроцессора, входящего в большинство реализаций. Препроцессор содержит средства для включения файлов, условной компиляции, а также определения макросов с параметрами. Использование препроцессора существенно повышает читабельность программ, а также их модульность. В настоящий момент Си реализован

практически на всех микро- и мини-ЭВМ, а также на многих больших компьютерах (напр., серии ЕС).

И.Б.Романенко, Е.М.Синица.

СИЛОД-ПК — то же, что и *MULTIS*.

СИМВОЛ (от греч. *σύμβολον* — знак, примета, признак) — вещественное, графическое или звуковое условное обозначение предмета либо *понятия*; образ, воплощающий какую-либо идею. Графический С. имеет вид буквы, цифры, геометрической фигуры, конфигурации или к.-л. иного *знака*, используемого как средство организации, управления или представления *данных*. Изображение С. состоит из смежных либо связанных штрихов или точек. С. может быть ассоциативным, т.е. своей формой напоминать форму объекта, или условным (абстрактным), т.е. обладать независимой от объекта формой, значение которой специально определяется. Осн. свойством С. является читаемость (см. *Читаемость информации*), зависящая от его воспринимаемости, идентифицируемости и различимости. Время поиска С. на экране распределяется в таком порядке: быстрее всего поиск можно осуществить по цвету, затем по форме, далее по яркости и медленнее всего — по размеру.

И.Ю.Никонова.

СИНТАКСИЧЕСКИЙ АНАЛИЗ — построение синтаксической структуры входного предложения на основе использования морфологической *информации*, полученной при *морфологическом анализе*, и сведений из *словаря*. Синтаксическая структура отражает синтаксические связи, существующие между словами в предложении. Т.к. после морфологического анализа слово в общем случае представлено в виде совокупности МИ-строк, то С.а. должен устанавливать синтаксические отношения не между словами (или словосочетаниями), а между конкретными МИ-строками (или последовательностями МИ-строк). В лингвистике наиболее употребительны два способа описания синтаксической структуры предложения: описание с помощью систем составляющих; описание с помощью деревьев зависимостей.

Н.А.Власенко.

СИНТЕЗ ПРОГРАММ — метод автоматического или автоматизированного построения *программы* по заданной спецификации (описанию постановки *задачи*, подлежащей *программированию*); раздел программирования, изучающий такие методы. Существуют различные методы С.п. — логический (дедуктивный), трансфор-

мационный, сборка и др. Логический С.п. основан на некотором исчислении логическом, служащем для формулирования спецификации программ и доказательства теорем. Одновременно с построением доказательства строится и программа, реализующая исходную спецификацию. Осн. проблема, возникающая в этом методе, — поиск доказательства (т.е. вывода) утверждений в используемом исчислении. В общем случае эта проблема алгоритмически неразрешима, что препятствует полной автоматизации логического С.п. Трансформационный С.п. состоит из двух этапов. На первом этапе по неформальному описанию задачи строится прототип программы, решающей эту задачу. Обычно прототип представляет собой программу высокого уровня абстракции (напр., набор рекурсивных управлений), которая не поддается эффективному исполнению на компьютере. Цель второго этапа — с помощью цепочки формальных преобразований (т.е. трансформаций) превратить прототип в эффективную программу на процедурном языке программирования. Каждый шаг преобразования должен сохранять семантику программы. Существующие системы трансформационного С.п. содержат богатые каталоги полезных трансформаций (до нескольких сотен правил) и помогают автоматизировать второй этап этого метода синтеза. Характерный пример трансформации — устранение рекурсии из программы путем замены ее на итерацию (т.е. циклы). Одной из существенных разновидностей трансформационного С.п. является смешанное вычисление программы. Пусть имеется программа $P(x_1, x_2, \dots, x_n)$ с n входными параметрами x_1, \dots, x_n . Если зафиксировать значения нескольких (но не всех) параметров (напр., $x_1 := a_1, \dots, x_k := a_k$), то можно построить новую программу $P'(x_{k+1}, \dots, x_n)$, которая является частичной конкретизацией программы P в том смысле, что для любых значений переменных x_{k+1}, \dots, x_n результат программы P' совпадает с результатом программы P при тех же значениях переменных x_{k+1}, \dots, x_n и ранее зафиксированных значениях x_1, \dots, x_k . Процесс построения такой программы P' наз. смешанным вычислением (в процессе построения может также вычисляться часть результатов исходной программы). Поскольку программа P' имеет меньшее количество аргументов, то ее можно сделать более эффективной, чем P . В этом осн. смысл применения смешанных вычислений для С.п. Сборка — это простейший способ С.п., когда конечная программа собирается из заранее заготовленных модулей в соответствии с заданной спецификацией. Примером автоматической системы сборочного С.п. является система ПРИЗ, разработанная под руководством академика Э.Х.Тыгу.

В.М.Антимиров.

СИСТЕМА КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ — комплекс программных и технических средств, реализующий одновременный и независимый коллективный доступ группы пользователей к информационным и вычислительным ресурсам электронной вычислительной машины с помощью терминальной сети. Режим коллективного использования ресурсов ЭВМ обеспечивается ее развитым базовым (системным) программным обеспечением, включающим пакеты прикладных программ, которые расширяют функции операционной системы по организации и поддержке множественного доступа. При этом каждый из некоторого множества пользователей имеет свою персональную ЭВМ, что достигается вследствие режима разделения времени работы ЭВМ на малые временные отрезки. Развитием системы режима разделения времени является система виртуальных машин, в которой каждому пользователю выделяется своя виртуальная ЭВМ со своими вычислительными ресурсами. КТО обычно ориентирована на ту или иную программную систему коллективного пользования.

В.Н.Никулин.

СИСТЕМА МАЛЫХ ЭВМ (СМ ЭВМ) — семейство микро-ЭВМ и мини-ЭВМ, используемых для автоматизированного обучения, автоматизации контроля, коммутации сообщений, в локальных и территориальных сетях сбора и обработки данных и др. Все модели СМ ЭВМ агрегативны, унифицированы и совместимы. СМ ЭВМ предназначена преимущественно для построения управляющих вычислительных комплексов, которые формируются на базе процессоров и периферийных устройств широкой номенклатуры.

Программное обеспечение СМ ЭВМ построено по модульному принципу, что обеспечивает возможность компоновки программных средств в соответствии с требуемыми режимами работы и выполняемыми функциями при заданной конфигурации технических средств. В состав программного обеспечения входят операционные системы различного назначения, библиотеки программ, программно-ориентированные и процедурно-ориентированные пакеты прикладных программ, сервисные и контрольно-диагностические программы.

СМ ЭВМ организована как открытая система с заложенными в ней возможностями пополнения и расширения технических средств и программного обеспечения. Агрегатный принцип построения СМ ЭВМ позволяет создавать различные варианты специализированных проблемно-ориентированных комплексов (см. *Инструментальный технический комплекс*).

Управляющие вычислительные комплексы на базе СМ ЭВМ при малых габаритах и сравнительно низкой стоимости отличаются высокой производительностью, экономичностью, проблемной ориентацией архитектуры, простотой диалога с пользователем.

И.А.Емченко.

СИСТЕМА ПОНЯТИЙ — знания о совокупности понятий, описывающих некоторую область человеческой деятельности. С.п., как правило, иерархичны. Понятие можно описать через ряд понятий более низкого уровня. Каждое понятие можно рассматривать как некоторую систему знаний о свойствах и законах изменения понятий. С.п. присуща определенная структура, включающая связи и отношения между образующими ее понятиями по некоторым правилам. Она имеет определенные правила построения и функционирования, может непрерывно обогащаться новыми элементами, изменять свою структуру, развиваться. Состав понятий меняется в результате взаимодействия их носителей с окружающей средой, а также в процессе решения задач. С.п. дискретны, ограничены и имеют конечное число понятий; следовательно, они не могут отразить многообразие и непрерывность реального мира. Объемы каждого из понятий, не введенных по определению, не имеют четких разделяющих границ. Конкретизация заключается в добавлении нового признака, что уменьшает объем понятия. Существует много переходных форм, затрудняющих проведение условных границ между объемами понятий.

Примером С.п. может служить термин “исчисление высказываний”. Исчисление представляет собой формальную систему и является объектом изучения матем. логики, которая, в свою очередь, содержит язык исчисления, формулы, аксиомы, теоремы, правила вывода и т.д. Новые понятия вводятся с помощью определений. Обобщение дает понятие с новым объемом. *Обучаемый* овладел С.п., если он способен: распознавать образующие ее понятия, формулировать их определения; применять все понятия из С.п. при решении практических задач.

А.Ф.Манако.

СИСТЕМА ПРОГРАММИРОВАНИЯ — компонент базового программного обеспечения, предназначенный для поддержки процесса программирования на компьютере. Представляет собой интегрированную среду, содержащую набор языковых процессоров и сервисных программ. Как правило, С.п. ориентирована на один входной язык программирования. В состав языковых процессоров С.п. традиционно входят: *текстовый редактор; транслятор; интерпретатор; редактор связей*, объединяющий отдельные модули в исполняемую программу; *отладчик*, предназначенный для проверки работы программы и поиска ошибок. В состав сервисных средств входят: библиотека периода трансляции, содержащая вспомогательные подпрограммы; библиотека периода исполнения, содержащая данные и программы, необходимые в процессе исполнения разработанной программы (драйверы устройств, шрифты, программы обработки ошибочных ситуаций); подсистема работы с файлами; подсистема

автоматизации процесса *тестирования программ*; информационная подсистема, содержащая сведения о режимах и средствах С.п., а также о средствах входного языка; монитор, управляющий функционированием С.п. С развитием *технологии программирования* понятие С.п. было обобщено до технологической линии производства программного продукта, поддерживающей все или осн. этапы жизненного цикла программной системы. Для разработки *программ учебного назначения* созданы специализированные С.п. с входным авторским языком программирования (AOC-M, LDS, Ракурс, Тенкор) и авторские системы (AUTOOL, WISE, ADAPT).

В.А.Петрушин.

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД) — система *программного обеспечения*, имеющая средства обработки на языке *базы данных (БД)*, позволяющая обрабатывать обращения к БД и поддерживать ее целостность. По типу структур БД различают сетевые, иерархические и реляционные СУБД; по типу взаимодействия с обрабатывающей программой — автономные СУБД и СУБД с включающим базовым языком программирования; по типу архитектуры, определяющей способ представления информационной базы, — двухуровневые и многоуровневые СУБД.

В состав осн. средств СУБД входят: средства описания структур данных; средства реорганизации и изменения структуры БД; средства обеспечения логической независимости программ от схем БД; средства телеобработки данных. Применяют СУБД по следующей схеме. Сначала вводятся знания о данных, т.е. строится схема БД. Далее, пользуясь этой схемой, СУБД решает задачи: загрузка БД; разработка ответов на запросы; манипулирование данными, т.е. преобразование формы данных, дополнение, изменение данных.

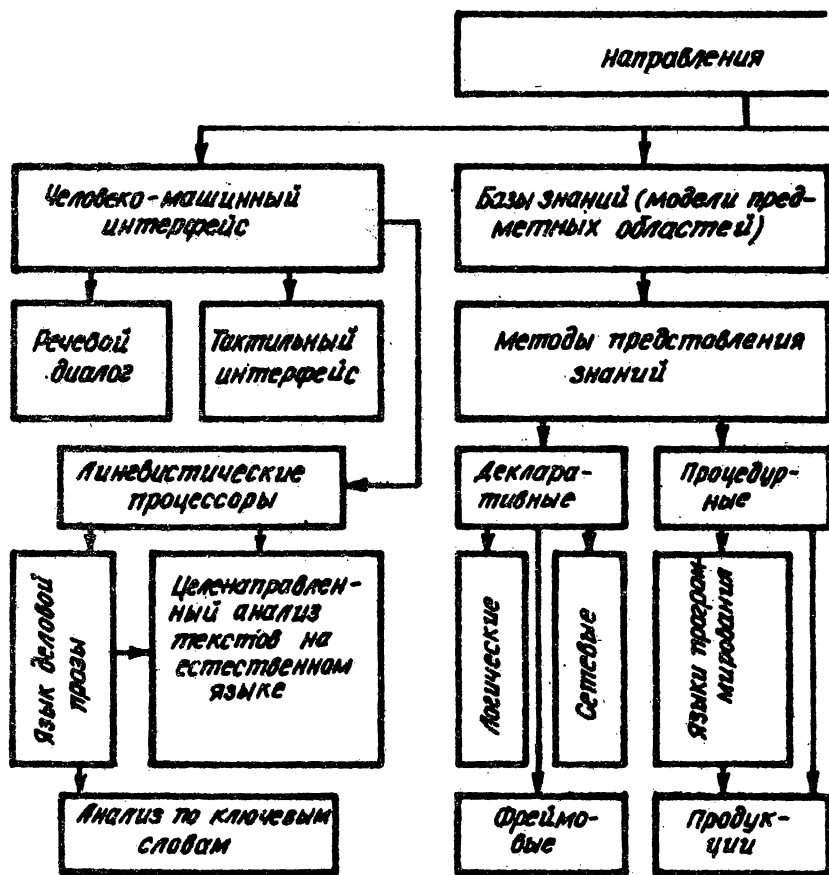
Многоуровневое представление данных предполагает соответствующее описание данных на каждом уровне и согласование одних и тех же данных на разных уровнях. Для описания данных используется язык описания данных. Для формулирования конечным пользователем произвольных запросов к БД применяется язык запросов, включаемый в состав СУБД. Язык манипулирования данными используется при написании прикладной программы. Основная его функция — выполнение функций ввода—вывода данных при работе с БД. С его помощью можно, напр., задать описание процедуры поиска данных на основе набора условий. Практически для всех СУБД характерно наличие операторов включения, исключения, изменения, извлечения и т.д. СУБД должна также выполнять функции по согласованию и управлению параллельной обработки БД, восстановлению БД, поддержанию семантической целостности БД.

Ныне СУБД поддерживают взаимодействие с пользователем в течение всего жизненного цикла БД. Наиболее многочисленной категорией пользователей, применяющих БД, должны стать пользователи-непрограммисты, для которых идеальным было бы формулировать запросы к БД на естественном языке.

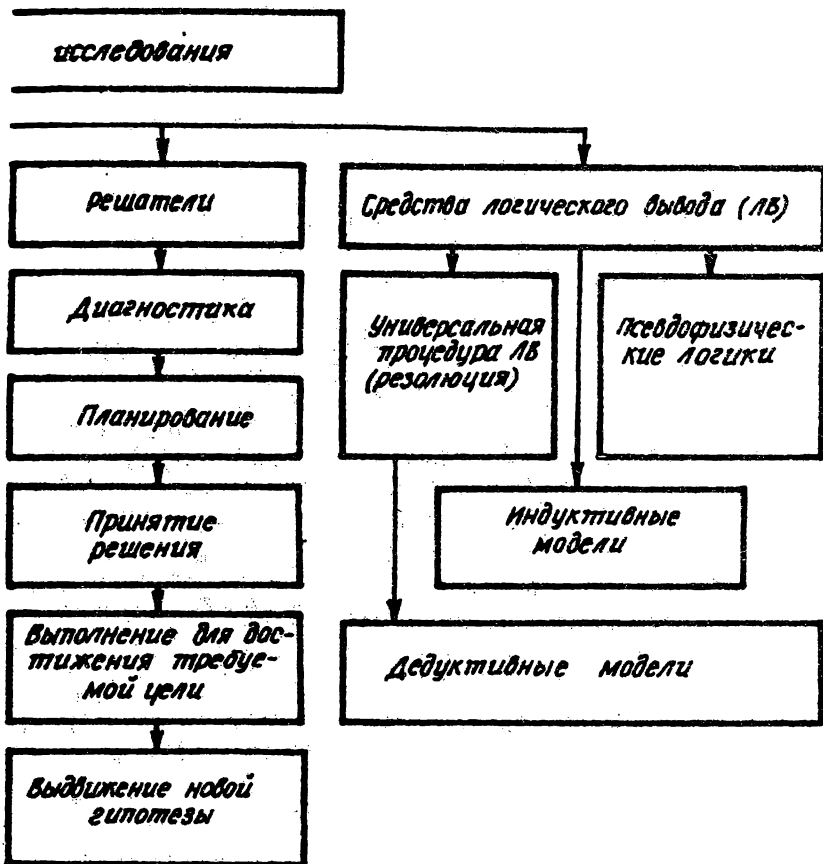
А.П.Ильяшенко, М.Е.Козлов.

СИСТЕМЫ, ОСНОВАННЫЕ НА ПРЕДСТАВЛЕНИЯХ ЗНАНИЙ — системы, в которых механизмы интеллектуальных рассуждений применяются к явным *представлениям знаний*. Имеют в своем составе *базу знаний (БЗ)*, подсистему решения задач (*решатель*) средства логического вывода, человеко-машинный интерфейс (рис.). БЗ содержит понятия, выраженные на языке *представления знаний*, и имеет соответствующие механизмы обработки. Представление знаний — главная проблема при создании С.,о. на п.з. Различные трактовки БЗ связаны с различными *методами представления знаний* — декларативными и процедурными (см. *Декларативное представление знаний, Процедурное представление знаний*). Процесс *приобретения знаний* в системе достаточно сложен как в плане *семантического анализа* вводимых знаний, так и в их структуризации в БЗ, которая может иметь весьма сложную структуру (см. *База знаний учебного назначения, База знаний инструментальной системы*) в зависимости от решения тех задач, на которые ориентирована система. С.,о. на п.з., используя знания о *предметной области (ПО)*, должна осуществлять диагностику процессов в системе, принимать решения, формулировать планы действий, выдвигать и проверять гипотезы и др. Эти функции возложены на решатель. Для приобретения новых знаний в БЗ, обобщения знаний и проверки их непротиворечивости необходимы средства логического вывода. Для интеллектуальной обработки информации требуются механизмы, позволяющие сокращать ненужный поиск, используя, напр., свою осведомленность о классе используемых задач. Можно использовать и универсальную *процедуру логического вывода* (см. *Исчисление логическое, Стратегия логического вывода*), основанную на так наз. принципе резолюции. Человеко-машинный интерфейс применяется для построения внешних уровней по отношению к С.,о. на п.з. Если при общении используются средства *речевого диалога*, то к тем функциям, которые выполняет *лингвистический процессор*, добавляются фонетический анализ и синтез. Известно также использование в качестве человеко-машинного интерфейса тактильного интерфейса. В последнее время получены существенные результаты и в области *машинного перевода*.

В КТО наибольшее значение имеют следующие классы С.,о. на п.з. (табл): интеллектуальные *информационно-справочные системы (ИИСС)*, интеллектуальные *обучающие системы (ИОС)*, *экспертные*



системы (ЭС). Для ИИСС, или систем взаимодействия с базами данных на естественном языке ограниченном (системы ПОЭТ, ЗАПСИБ, ЗАПСИБ-ПРОЛОГ и др.), характерно наличие лингвистического процессора и модели предметной области. Работа ИОС основана на анализе модели ПО, модели обучаемого и модели обучения. К ИОС относятся прежде всего генерирующие автоматизированные обучающие системы (ГАОС): СОАСН, MALT, SCHOLAR. Один из осн. блоков ГАОС — генератор сообщений, работающий на основе модели ПО (см. Генератор формул исчисления предикатов первого порядка). Оснащение компьютеров цветными графическими дисплеями сделало возможным широкое применение тренирующих систем и тренажеров (см. Тренирующие программы,



Тренажер по программированию). Эти системы требуют таких средств представления знаний, используя которые можно было бы построить модель ПО и динамически ее корректировать в зависимости от сложившейся ситуации. Среди ЭС выделяют инструментальные ЭС (BESS и др.). Наибольший интерес для КТО представляют экспертные обучающие системы (ЭОС). Для функционирования ЭОС, кроме знаний по предмету, необходимы знания об обучаемом, о процессе обучения, по методике обучения и диагностике ошибок. Разработка ЭОС требует не только развитых средств представления знаний, но и взаимодействия различных источников знаний (см. также САКИО, BETS, SOPHIE, WHY, GUIDON, BUGGY). Для реализации С., о. на п.з. можно использовать языки Пролог, МПролог,

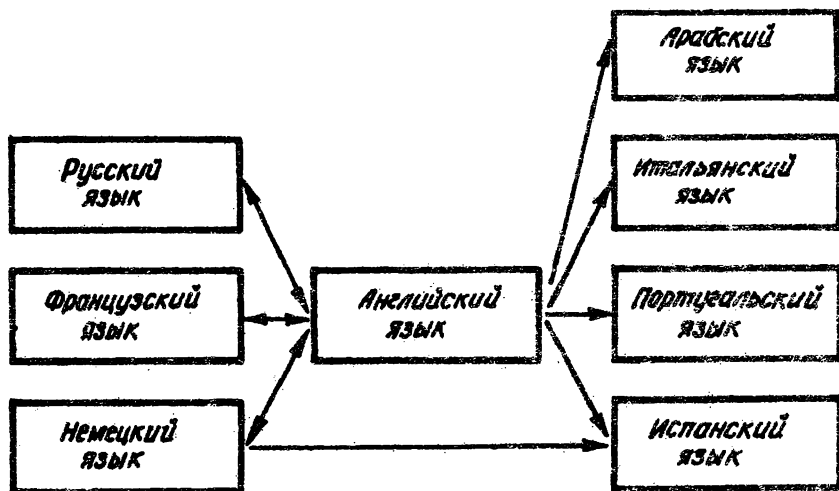
Турбопролог, ARITY/PROLOG, Лисп, интегрированные системы БЗ и БД (ПРОЛОГ-СИЗИФ, ПРОЛОГ-АДАБАС).

Основные классы систем, основанных на представлениях знаний в компьютерной технологии обучения

Класс системы	Отличительные признаки	Примеры систем
ИИСС	1. Модель предметной области 2. Модель языка (словарь, грамматика)	ЗАПСИБ, ПОЭТ, FLIGHT
Системы машинного перевода	3. Модель перевода	СИЛОД, СИСТРАН, BRAWIS, ATLAS-II ПЛАСТ, СПАС
ИОС	1. Модель предметной области 2. Модель обучаемого 3. Модель обучения	ГЕОГРАФ, СОАСН, MALT
ЭС	1. Подсистема объяснений 2. Знания эксперта.	
Инструментальные ЭС	3. Подсистема приобретения знаний 4. Подсистема общения 5. Подсистема решения	BESS, GURU, MYCIN
Прикладные ЭОС	6. Модель предметной области 7. Модель обучаемого 8. Модель обучения 9. Знания по стратегиям и методикам обучения 10. Знания об ошибках (диагностика ошибок)	САКИО, BETS, WHY, SOPHIE, GUIDON

С.П.Кудрявцева.

СИСТРАН (англ. SYSTRAN, от System of Translation — система перевода) — одна из первых в мире коммерческих систем машинного перевода (МП). Ее можно рассматривать в качестве классического примера использования стратегии прямого перевода. С. была создана в 70-х гг. и является ныне наиболее развитой системой МП. Объем словаря для каждой языковой пары составляет около 70 тыс. слов и 35 тыс. оборотов. С. ориентирована на перевод научно-технических текстов, причем располагает средствами и настройки на определенный тип документов (описание изобретений, тексты стенограмм и т.д.). Недостаток С.: высокая совместимость составления требуемых для нее словарных статей (однако сегодня это компенсируется очень высокими скоростями перевода). С. разработана фирмой SYSTRAN (США) и реализована на машинах IBM-360/370, а



позднее версии модифицированы для компьютеров Siemens-770.

К.Р.Пиотровская.

SCHOLAR — генерирующая автоматизированная обучающая система, предназначенная для обучения географии. Самостоятельно генерирует текст, вопросы и соответствующие ответы, оценивает ответы обучаемого, сравнивая его ответы с собственным сконструированным решением, классифицирует их как правильные, неправильные, частично или приблизительно правильные, оказывает необходимую помощь. Может подсказать обучаемому ответ, показать, когда обучаемый его не понимает, обнаружить орфографические ошибки, ответить на вопрос обучаемого, в результате чего между ними возможен диалог в режиме поочередного обмена информацией с вопросами и ответами с обеих сторон.

Система может придерживаться содержания текста урока или изменять его в зависимости от целесообразности и наличия времени, не требуя при этом специфических или подробных указаний, а применяя общие критерии и методы к имеющемуся материалу. Работа ее основывается на модели предметной области, представленной в виде информационной семантической сети, которая представляет собой организацию элементарных информационных единиц, выраженную через их смысл и взаимосвязи. Каждая информационная единица характеризуется определенной семантической информацией, представленной набором признаков, таких, как словесное обозначение информационной единицы, ее значение, набор меток для удобства выбора исполнительной программой и пр.

Значение признака может выражаться либо набором признаков, либо ссылкой на совокупность других информационных единиц, характеризующихся иными признаками. Благодаря этому возможны различные комбинации объектов, составляющих *модель предметной области*. Информационная сеть образует сложную сеть фактов, *понятий*, методов, напр., вывод логического умозаключения о климате местности на основе использования понятий, характеризующих местность (широта, долгота и др.). Иными словами, если климат местности не запрограммирован фактически (в информационных единицах, обозначающих температуру, уровень осадков и т.п.), о нем будет сделан логический вывод с большой степенью вероятности успеха на основе знаний о графической широте, высоте над уровнем моря и т.д.

Исполнительная программа SCHOLAR почти не зависит от содержания предмета, к которому она применяется. Так, при смене района изучения обновляется только часть информационной сети, полная же замена предмета будет обозначать почти полный пересмотр содержания семантической сети, но не исполнительной программы. При генерации вопросов была принята методика, заключающаяся в построении коротких предложений без использования придаточных. Задача генерации состоит в поиске элементов сети, соответствующих плану обучения, формировании их согласованной последовательности. SCHOLAR реализует двухсторонний диалог на *естественном языке ограниченном*. Нераспознанные в сообщении слова локализируются и предъявляются обучаемому для реформулировки.

Пример диалога обучаемого с SCHOLAR (сообщения обучаемого помечены *).

Здравствуйте, меня зовут SCHOLAR. Я желаю вам помочь в изучении географии Южной Америки. Приступим к дискуссии. Что такое Чили?

* страна *

Прекрасно. Используйте одно из следующих слов. — Сукре, французский, -испанский для ответа на вопрос: На каком языке говорят в Чили? Вы слишком долго думаете

* Испанский *

Вам следовало бы написать "Испанский". Очень хорошо. Какова приблизительно площадь Бразилии?

* 2 500 000 кв. миль *

Неверно.

Пожалуйста, ответьте, правильны или нет следующие утверждения: Площадь Парагвая приблизительно 47 432 кв. миль.

* Какова площадь Бразилии? *

Площадь Бразилии приблизительно 3 600 000 кв.миль.

Теперь ответьте на вопрос, на который вы не дали ответа ранее.

* Ошибочно *

Перефразируйте ваше утверждение, я не могу понять его.

* Неверно*

Очень хорошо.

Какова приблизительно численность населения в Бразилии?

* 60 000 000*

Приблизительно верно, но более точно — 70 000 000.

Л.Н.Гецко.

СЛОВАРЬ — упорядоченная совокупность слов с пояснениями, толкованиями или переводами на другой язык. Существуют различные типы С.: двуязычный, орфографический, толковый, терминологический словари, *тезаурус* и др. Алфавитный список лексических единиц С. наз. *словником*. Словарный компонент в том или ином виде присутствует в составе *информационной системы* и чаще всего служит для идентификации поступающих *запросов*. Наряду с этим автоматизированный словарный комплекс может выступать и как самостоятельная информационная система, способная выполнять справочные и обучающие функции (см., напр., *Автоматизированный словарный комплекс учебного назначения*).

Т.Б.Андрусенко.

СЛОЖНОСТЬ ЗАДАЧИ, уровень сложности задачи — количественная характеристика структуры процесса решения задачи, тесно связанная с уровнем *трудности задачи* (точнее обнаруживающая достаточно высокую положительную *корреляцию* с ним). Различают два вида С.з.: реальную, т.е. сложность реального (или возможного) процесса решения задачи конкретным *решателем* (или решателем определенного типа) и *нормативную*, т.е. сложность процесса решения ее нормативным способом. Если существует несколько *нормативных способов решения* некоторой задачи, то ее можно охарактеризовать несколькими значениями нормативной сложности. Напр., одна и та же матем. задача может обладать для одного и того же решателя различной сложностью в зависимости от того, как ее решать — арифм. или алгебр. способом. Реальная С.з., как правило, выше или равна нормативной, которая не зависит от особенностей отдельных решателей, но может существенно зависеть от их типа (от контингента субъектов, если речь идет о решении задач людьми). Это связано с тем, что для решателей разных типов часто планируются различные нормативные способы решения одинаковых задач.

Наиболее четко содержание *понятий* о реальной и нормативной С.з. выявляется при алгоритмическом подходе к их оценке. Согласно ему реальная С.з. оценивается по числу элементарных (эффективных или квазиэффективных) *операций* в реальном (либо в таком,

который, возможно, осуществляется) процессе ее решения, а нормативная — по числу таких операций в нормативном способе ее решения. Оценка нормативной С. з. на основе алгоритмического подхода возможна для рутинных и квазирутинных задач, а реальной С.з. — и для нерутинных, поскольку и для них часто можно указать последовательность фактически осуществленных в процессе решения элементарных операций. Напр., число произведенных слепым шахматистом при обдумывании хода “вызовов информации” (фиксаций пальцами полей доски) служит показателем реальной С.з. поиска хода. Обобщением алгоритмического подхода к оценке С.з. является операционный, при котором требование элементарности выделяемых операций снимается. Поскольку при этом отдельные виды операций могут сильно различаться по уровню трудности, каждому виду операций приписывается так наз. коэффициент сложности, пропорциональный среднему времени выполнения операции этого вида (ее средней трудности). При этом С.з. оценивается как сумма коэффициентов сложности операций, последовательно выполняемых в ходе ее решения. Для установления коэффициентов сложности можно идти по пути структурного анализа отдельных видов операций, входящих в рассматриваемый способ решения задачи. Весьма распространен энтропийный подход к оценке С.з. (прежде всего познавательных). Согласно этому подходу реальная С.з. оценивается по величине неопределенности, устраняемой в реальном успешном процессе решения задачи, а нормативная — по величине неопределенности, которая должна устраняться, если решение задачи осуществляется в соответствии с некоторой нормой. С.з., т.е. устраняемая неопределенность, определяется при этом как разность значений энтропии (величины неопределенности), характеризующих соответственно исходную ситуацию и ситуацию, получаемую в результате успешного решения задачи. Наряду с оценкой первоначальной С.з. можно выяснять и то, как она уменьшается по ходу решения. В результате открывается возможность сравнительной оценки различных стратегий решения задач. Со С.з., оцениваемой на основе энтропийного подхода, тесно связана так наз. информационная напряженность, измеряемая количеством информации, которое решатель должен переработать или фактически перерабатывает за единицу времени, решая стоящую перед ним задачу. Если в соответствии со С.з. находится, как правило, их интегральная трудность, то в качестве коррелята информационной напряженности выступает дифференциальная трудность.

Для прогнозирования трудности задач, решаемых в процессе достаточно сложной деятельности, часто приходится учитывать систему показателей сложности этих задач. Каждый из них отражает определенный аспект структуры нормативного или реального процесса решения задачи. Напр., показателями С.з. понимания учебного материала являются среднее и макс. значение количества понятий

в подлежащих усвоению суждениях, полнота воспроизведения в них ранее изложенной информации и др. Варьируя с помощью многоуровневых обучающих программ показатели этого рода, можно установить для каждого обучаемого такой уровень трудности решаемых задач, который способствует успешному протеканию учебной деятельности. Иногда на основе отдельных показателей С.э. определенного типа удается получить ее единую числовую оценку. Это достигается, в частности, при прогнозировании трудности задачи понимания печатного текста с помощью так наз. формулы читабельности (см. *Читабельность текста*).

Г.А.Балл.

СМОЛТОК-80 — объектно-ориентированный язык программирования, предназначенный для создания интеллектуальных программ и систем в режиме разделения времени с широким применением средств машинной графики. Разрабатывается с начала 70-х гг. в США. В обучении может применяться для создания моделирующих учебных программ, тренажеров, экспертных обучающих систем, а также для экспериментов в области представления учебных знаний.

Осн. типом данных в С.-80 является объект, содержащий процедурную и понятийную части (объект=данные+процедуры). Свойства объекта зависят от того, какие сообщения он может получать. Описание совокупности свойств (методов) наз. спецификацией объекта. Спецификация объекта, зарегистрированная в системе, представляет собой класс; классы также являются объектами. Существует множество объектов одного и того же класса, которые наз. экземплярами класса. Спецификация экземпляра состоит из объявления переменных и методов экземпляра. Для каждого класса может указываться надкласс и подкласс.

Передача управления осуществляется на основе посылки сообщений от объекта к объекту. В сообщении указывается объект—получатель сообщения и само сообщение. Если объект не содержит методов для обработки сообщения, то управление передается соответствующему классу, который стремится обработать сообщение своими методами. При неуспехе сообщение передается надклассу и т.д. Следовательно, объект получает возможность наследовать свойства классов более высокой иерархии. При поиске объекта с подходящими для данного сообщения методами возможно возникновение перебора с возвратом. Дедуктивный механизм явно не выражен. Некоторый вывод можно получить на основе поиска подходящего метода по иерархии классов. Интерфейс с пользователем осуществляется на основе визуализации объектов с использованием средств машинной графики.

В.А.Третьяк.

СНОБОЛ-4 (англ. SNOBOL, от StriNg Oriented symBOlic Language — символический язык, ориентированный на обработку строк) — язык для программирования задач, связанных с обработкой текстовой информации. Разработан в 1969 в США. В обучении применяется для организации естественногоязыкового интерфейса, манипулирования текстовой информацией и пр.

Осн. типом данных в С.-4 являются цепочки символов произвольной длины, из которых могут формироваться более сложные типы данных. Цепочка, служащая для ссылки на тип данных, наз. формальным идентификатором типа данных (табл.). Программа на С.-4 является последовательностью инструкций. При её выполнении цепочка символов, представляющая собой некоторую инструкцию, преобразуется в объектный код (функция CODE). Программа выполняется последовательно, начиная с первой инструкции, если явно не указывается переход на метку — в этом случае управление передаётся на инструкцию с соответствующей меткой.

Формальные идентификаторы типов данных, допустимых в Снобол-4.

Тип данных	Формальный идентификатор
Цепочка	STRING
Целое число	INTEGER
Вещественное число	REAL
Структурный образец	PATTERN
Массив	ARRAY
Таблица	TABLE
Создаваемое имя	NAME
Невычисляемое выражение	EXPRESSION
Объектный код	CODE
Определяемый программистом	Имя типа
Внешний	EXTERNAL

Осн. операцией в С.-4 является ассоциативная выборка. С её помощью переменным присваиваются значения; от результата завершения сопоставления зависит передача управления. Элементарный объект сравнения — символ. Образцы конструируются из элементарных образцов с использованием операций альтерации и конкатенации. Элементарными образцами могут быть как цепочки символов, так и идентификаторы. Передача управления на следующую инструкцию осуществляется в зависимости от успеха или неуспеха выполнения предыдущей инструкции. О мощности образцов можно судить исходя из того, что программа синтаксического разбора некоторой грамматики очень похожа на запись этой грамматики в форме Бекуса-Наура. Интерфейс с пользователем осуществляется использованием переменных INPUT и OUTPUT.

В.А.Третьяк.

СОВМЕСТИМОСТЬ И ПРЕИМУЩЕСТВЕННОСТЬ компьютерной технологии обучения — совместимость различных этапов (до вуза, в учебном процессе вуза и после его окончания) внедрения *компьютерной технологии обучения* (КТО), реализуемая в форме преимущественности накапливаемых умений *обучаемых*. Осуществляется на основе согласования содержания КТО с различными звеньями системы образования. В перспективе возможны новые формы реализации этого принципа, напр., отбор в вузы и техникумы с учетом уровня *компьютерной грамотности* абитуриентов, организации вступительных экзаменов на базе вычислительной техники.

Н.М.Когдов, А.Я.Савельев, В.А.Стаканов.

СОРЕВНОВАТЕЛЬНАЯ ИГРА — игра, в которой игрок стремится выполнить игровые действия по некоторым параметрам лучше (напр., безошибочнее или быстрее), чем соперник. Игровой компонент порождается именно за счет соревновательных мотивов. Процесс решения *игровой задачи* в С.и. необходимо оценивать по определенному критерию (временному, экономии ресурсов и т.п.). В *компьютерных играх*, как правило, в этих целях используется система бонификаций (штрафов) и некоторый счетчик баллов (очков), позволяющий игроку сравнивать текущий результат с результатом противника или результатом, полученным предыдущим игроком, либо со своим предыдущим результатом (соревнование с самим собой).

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

SOPHIA — специализированная *авторская система*, предназначенная для создания обучающих, тестирующих и тренирующих программ *учебного назначения*. Реализована на IBM PC для обучения *иностранному языку с помощью компьютера*, однако может использоваться и в других *предметных областях*. Состоит из подсистемы *автора*, с помощью которой автор создает *файлы* упражнений, и подсистемы *обучаемого*, которая выбирает упражнения из файлов и представляет их обучаемому. В подсистему автора входит *текстовый редактор и компилятор с авторского языка программирования*, предназначенный для разработки упражнений и рассчитанный на *пользователя*, не являющегося профессиональным программистом. Упражнения могут иметь несколько вариантов. Вариант упражнения может использовать глобальные (общие для нескольких вариантов или в целом по упражнению) и локальные *тексты* (относящиеся к одному варианту упражнения). Программа обучаемого позволяет автоматически получать из *обучающей программы* тренирующую и/или тестирующую.

В системе поддерживаются следующие типы обработки ответов: множественный выбор, словесный ответ, сопоставление, иденти-

фикация, заполнение бланков, числовой ответ. В ней реализован выбор упражнений и их вариантов из *меню*, а также случайный и динамический выбор на основе значений переменных, отражающих уровень их сложности. Автор может определять условия ветвления на основе ответов обучаемого и результатов выполнения предыдущих упражнений. Каждый файл упражнений можно снабдить вводными и заключительными рекомендациями, а также определением условий для оценки уровня *знаний* обучаемого. В системе может создаваться протокол ответов обучаемого. Подсистема автора, в т.ч. авторский язык, имеет две версии: на чешском и английском языках. Программа обучаемого может выполняться на чешском, словацком, английском, французском, немецком, испанском и шведском языках.

Т.А.Лисюк.

SOPHIE — интеллектуальная система для обучения поиску неисправностей в электрических цепях. Работа с SOPHIE построена следующим образом. *Обучаемому* предъявляется схема фрагмента цепи, содержащего неисправность, которую необходимо обнаружить. Поиск осуществляется в *диалоге* с системой, которая информирует обучаемого о значениях параметров цепи, выполняет замену отдельных элементов, а также отвечает на его вопросы. Инициатива в диалоге принадлежит обучаемому: SOPHIE не навязывает ему к.-л. *стратегии* поиска неисправностей, а также не выясняет причины допущенных ошибок. Задача системы заключается в создании "дружественной" среды для самообучения. Обучающий характер взаимодействия проявляется в том, что система оценивает целесообразность очередного действия обучаемого (измерения параметра или замены элемента) с точки зрения возможности локализации ошибки и информирует его в том случае, если это действие не является необходимым.

Основу системы составляют блок выдвижения и оценки гипотез о состоянии цепи, *база знаний* (БЗ) по электрическим цепям, представленная *семантической сетью*, имитатор цепи, естественно-языковая система и *модель обучаемого*, отражающая текущие знания обучаемого о задаче. Имитатор электрической цепи служит для моделирования произвольных фрагментов цепи и измерения параметров, БЗ используется преимущественно для анализа естественно-языковых сообщений. Блок выдвижения и оценки гипотез используется для оценки действий обучаемого и *ответа* на некоторые *вопросы*, напр., какие элементы могут быть неисправны (на основании проведенных измерений). С этой целью порождаются все гипотезы, не противоречащие наблюдаемому поведению цепи, а затем имитируются соответствующие фрагменты цепи и выполняются необходимые измерения. Естественно-языковая система SOPHIE основана на так наз. выполнимой или семантической *грамматике*,

в которой синтаксические категории (существительное, глагол и т.п.) заменены семантически значимыми категориями (измерение, элемент цепи и т.п.). Это позволило выполнять анализ анафорических ссылок, контекстно-зависимых удалений и эллиптических конструкций. Реализация естественно-языковой системы с помощью АТН-машины позволила в десятки раз повысить быстродействие. В последних версиях в систему были включены: учебный материал для подготовки к лабораторным занятиям, организованный по фреймовому принципу; подсистема отладки и конструирования объяснений, предназначенная для экспертного решения задачи поиска неисправностей и объяснения осн. этапов этого решения; блок поиска ошибок в режиме *деловой игры* между двумя группами обучаемых.

Е.М.Синица.

СПАС (Система Перевода с помощью Автоматического Словаря) — система человеко-машинного перевода (см. *Машинный перевод*), использующая двуязычный словарь и ряд сервисных средств по подготовке исходного текста. Осн. таким средством является многофункциональный оконный редактор для исправления орфографических и грамматических ошибок, изменения порядка слов, изменения структуры текста или его формата. Переводчик имеет возможность отредактировать текст, распечатать его, передать в локальную сеть, канал связи и т.п. Ненайденные слова или словосочетания по желанию переводчика можно ввести в осн. словарь системы.

Объем автоматического словаря — 20 тыс. словарных статей. Словарь содержит термины по вычислительной технике, микроэлектронике, программированию. Система разработана ВЦП РАН и СО "ДЗУ" (Болгария) для персональных компьютеров, совместимых с IBM PC с жестким диском; предусмотрено объединение нескольких систем СПАС в локальную сеть и обмен словарной информацией между отдельными местами для организации современного переводческого бюро.

К.Р.Пиотровская.

СПОК-ВУЗ (Система Программирования Обучающих Курсов для ВУЗов) — специализированный пакет прикладных программ учебного назначения, предназначенный для автоматизации разработки учебных курсов по различным предметным областям и организации автоматизированного обучения на базе ЕС ЭВМ под управлением операционной системы ДОС ЕС. Дает возможность: применять компьютеры для обучения, оказания помощи и контроля знаний обучаемых, работающих на машине одновременно по индивидуальным диалоговым программам (курсам); существенно ускорить и облегчить процесс создания прикладных обслуживающих и обуча-

ющих программ для различных применений; обеспечить административную учебное заведение или вычислительного центра необходимыми данными о ходе и результатах автоматизированного обучения и обслуживания *пользователей*. С.-В. обеспечивает эффективное *программирование* и использование осн. режимов взаимодействия с *обучаемыми*: получение консультации и справки; *программированное обучение*; диагностика потребностей и/или знаний обучаемого с последующей отсылкой его к соответствующим печатным материалам; контроль знаний, умений и навыков с выставлением оценки и распечаткой протокола контроля; обучающе-справочный режим; управляемое машиной программирование на незнакомом пользователю *языке программирования*.

В состав технических средств, необходимых для функционирования С.-В., входят: компьютеры серии ЕС ЭВМ (ЕС-1020, ЕС-1022, ЕС-1033 или старших моделей) с оперативной *памятью ЭВМ* объемом не менее 128 К; дисководы ЕС-5052 на сменных *магнитных дисках* (не менее двух); штатный набор *периферийных устройств* ЕС ЭВМ; серийные алфавитно-цифровые дисплейные комплексы ЕС-7906 с индикаторами на электронно-лучевой трубке ЕС-7066 и устройствами группового управления ЕС-7566 (до четырех комплектов).

Специализированное программное обеспечение состоит из базовой *системы программирования* обучающих курсов, расширения базовой системы и расширения *языка описания курсов* (ЯОК). Базовая система включает: конфигуратор, обеспечивающий *адаптацию* системы к конкретной совокупности технических средств; инициализатор, проверяющий при запуске системы реальную техническую и программную обстановку, *файлы, данные*, защиту от ошибок и т.п.; управляющую программу, которая обеспечивает обслуживание *терминалов* пользователей, организует связь между файлами системы, находящимися на различных устройствах, исправляет возникающие ошибки и следит за прохождением в системе работ, вызванных сообщениями пользователей; *интерпретатор* — обеспечивает интерпретацию и выполнение *операторов ЯОК* и языков директив пользователей; библиотеку *автоматизированных учебных курсов* (АУК), обеспечивающую одновременное хранение в системе до 400 прикладных диалоговых программ (курсов); архив, предназначенный для хранения данных о работе пользователей, а также информации о функционировании системы. Расширение базовой системы включает: а) программу управления экраном *дисплея*, обеспечивающую ввод и вывод сообщений в заданную позицию экрана, очистку и заполнение любым *символом* определенной части экрана, сдвиг кадра на заданное число строк; б) программу диалоговых вычислений, обеспечивающую вычисление значений алгебраических, тригонометрических и логических выражений; вычислитель можно использовать также для проверки правильности

числовых *ответов*; входной язык вычислителя в основном соответствует стандартным правилам записи математических выражений; в) программу генерации случайных чисел, позволяющую получить равномерно распределенные случайные целые числа в интервале от 1 до 63 и случайные последовательности чисел в интервале от 1 до 49; эту программу можно использовать для случайной выборки последовательности заданий; г) программу анализа и обработки сообщений, позволяющую определять длину ответов обучаемых, позиции первого несовпадающего символа строки и первого вхождения подстроки в строку; д) программу *синтаксического анализа*, анализирующего конструкцию языка *Фортран* и ответы на формальных языках, описываемых контекстно-свободными *грамматиками*; е) программу обмена с периферийными устройствами, позволяющую выводить *информацию* на пишущую машинку, алфавитно-цифровое *печатающее устройство* (АЦПУ), перфокарты, магнитную ленту. Средства расширения языка описания курсов дают возможность пользователям включать в С.-В. дополнительные подпрограммы, реализующие функции, отсутствующие в ЯОК. Реализован аппарат внешних динамических функций.

С.-В. поддерживает работу *диспетчера*, автора АУК, преподавателя и обучаемого. Диспетчер выполняет *процедуры*: 1) связанные с работой обучаемых; 2) относящиеся к работе авторов и обслуживанию курсов; 3) управления работой системы. Первая группа процедур предназначена для регистрации обучаемых в системе и удаления их после изучения курса, просмотра и изменения данных обучаемых; ко второй группе относятся регистрация, копирование, переконпоновка и удаление курсов, получение распечатки курсов на АЦПУ и др.; третья группа связана с ежедневной инициализацией С.-В., планированием работы пользователей, получением статистических данных о работе обучаемых с курсами. В задачи автора входит разработка сценария АУК, подбор и структуризация учебного материала и заданий, написание программы курса на ЯОК, ввод, редактирование и *отладка программы* курса. Задача преподавателя — помочь обучаемым во время занятий в дисплейном классе. Обучаемые могут управлять последовательностью изучения курса, посылать при необходимости сообщения автору курса (эти сообщения помещаются в архив, откуда диспетчер может вывести их на АЦПУ и передать автору).

О.П.Платонова.

СПОСОБ РЕШЕНИЯ задачи — *процедура*, которая при осуществлении ее *решателем* может обеспечить решение задачи. Предписания по С.р., в т.ч. *алгоритмы*, *квазиалгоритмы*, эвристические предписания (см. *Эвристические средства*), относятся к *средствам решения* задач. Различают С.р., реально осуществляемые тем или

иным решателем, и нормативные (эталонные). Существуют разные аспекты рассмотрения С.р. Если, напр., С.р. геом. задачи рассматривается в матем. аспекте, то речь идет о ее решении идеализированным решателем, охватывающем, в частности, средства евклидовой геометрии или к.-л. ее раздела, а также некоторые средства логики и, возможно, арифметики, алгебры и др. дисциплин. Если же С.р. той же задачи рассматривается в психологическом аспекте, то в качестве решателя выступает человек, в большей или меньшей степени владеющий перечисленными средствами. Различен и состав *операций*, из которых строятся обсуждаемые в матем. и психологических исследованиях С.р. задачи. Если человек овладел или должен овладеть средствами решения задач, предоставляемыми, напр., некоторым разделом евклидовой геометрии, то в С.р. задачи этим человеком могут входить геом. операции (вообще — операции, соответствующие *операторам*, имеющимся в рассматриваемом идеализированном решателе). Но характер таких операций при этом, как правило, изменяется. В частности, эффективными операциями, реализуемым идеализированным решателем, соответствуют (в лучшем случае) квазиэффективные операции, реализуемые человеком. Вместе с тем в состав С.р. задачи человеком входят также операции, обеспечивающие ориентировку в ситуации (анализ предмета задачи), планирование последующих операций и т.п.

Г.А.Балл

СПРАВОЧНО-ОБУЧАЮЩИЙ РЕЖИМ (СОР) — форма организации *диалога*, целью которого является оперативное предоставление *пользователю* справочной или дополнительной *информации* в *ответ* на его *запрос*. Осн. отличие СОР от справочного обслуживания, характерного для многих *информационных систем*, состоит в том, что в ходе справочно-обучающего диалога активно используются индивидуальные характеристики пользователя и *данные* о предшествующих сеансах его работы, т.е. модель пользователя. Это позволяет предъявлять запрошенные данные в форме, наиболее приемлемой для конкретного пользователя. Напр., один пользователь получит определение интересующего его понятия в краткой формулировке, другой — в развернутой, а третий — в краткой, но с примерами, причем выбор представления осуществляется либо в результате уточняющего диалога (если в модели нет соответствующей инфо. ладии), либо автоматически на основании данных модели. Применение СОР в процессе диалога с *обучающей системой* целесообразно как в качестве самостоятельного режима работы, при котором человек может получать сведения, напр., о возможностях системы, о наличии учебного материала и т.д., так и в качестве составной части таких диалоговых режимов, как *обучение*, тренировка, совместное решение задачи. В этих случаях применение СОР

направлено обычно на получение краткого информационного сообщения или учебного примера, которые содержат сведения, необходимые пользователю для преодоления возникших затруднений.

Реализация СОР имеет ряд существенных особенностей. Во-первых, инициатива осуществления диалога полностью принадлежит человеку, который своим запросом инициирует процесс поиска необходимой информации. Потребность в справке может возникнуть в любой момент работы; при этом правила формирования запроса не должны требовать от пользователя специальной подготовки. Следовательно, программная система, обеспечивающая СОР, должна быть постоянно готова к приему сообщения пользователя. Во-вторых, само функциональное назначение СОР предполагает фрагментарность процесса диалога и возможность произвольного доступа к информации, которой располагает обучающая система. В-третьих, следует учитывать, что пользователь не обладает полными знаниями о возможностях и правилах работы системы, обеспечивающей выдачу справки. Следствием этого является достаточно высокая вероятность неполных, избыточных или даже противоречивых запросов, для обработки которых может потребоваться уточняющий диалог.

Возможность *адаптации* к пользователю в СОР позволяет ускорить анализ и выполнение запросов, упростить выбор данных и, следовательно, поддерживать необходимый уровень естественности и лаконичности диалога при сохранении его информативной емкости.

М.В.Легкий.

СРЕДСТВА РЕШЕНИЯ задач — материальные и идеальные объекты, не входящие в задачу, но привлекаемые для ее решения. К С.р. относятся *операторы*, которыми располагает *решатель*, а также используемые им *операнды*, дополнительные к имеющимся в предмете задачи. С.р. подразделяются на внутренние (входящие в состав решателя) и внешние (не входящие в его состав, но применяемые им). Осн. С.р. рутинных и квазирутинных задач являются соответственно *алгоритмы* и *квазиалгоритмы* их решения. Для решения *проблемных задач* привлекаются так наз. *эвристические средства*.

При *решении задач с помощью компьютера* у пользователя появляются дополнительные С.р. задач, в т.ч. вычислительные и прочие реализуемые на компьютере алгоритмы, а также разнообразные *модели*, форма представления которых повышает их эвристический потенциал. См. также *Способ решения задачи*.

Г.А.Балл.

СТАТИЧЕСКАЯ РАБОТА (от греч. *στατός* — стоящий, неподвижный) — работа, при которой мышцы сокращаются в

фиксированной позиции. Характеризуется непрерывностью напряжения мышц (напр., неподвижная поза, удерживание на весу к.-л. предмета). При выполнении С.р. обычно не нужны к.-л. энергичные движения, напр. перемещение предметов. Вместе с тем С.р. сопровождается значительным расходом энергии и быстро вызывает утомление. Во время статического напряжения потребление кислорода организмом уменьшается, а сразу же после его прекращения резко возрастает, усиливается кровоток. С.р. регулируется с помощью рефлексов положения отдельных конечностей и рефлексов позы, т.е. рефлексов сохранения равновесия тела человека при различном положении конечностей и туловища (стоя, сидя, на корточках и т.д.).

При использовании компьютера возникает взаимосвязь между *рабочей позой* пользователя и неподвижностью экрана, а также визуальными эргономическими факторами, такими, как *блескость* экрана, которая может послужить причиной значительных ограничений в изменчивости рабочей позы. Использование *рабочих мест*, позволяющих регулировать высоту рабочей поверхности, высоту сидения или др. параметры, ведет к снижению мышечного дискомфорта. При изучении утомляемости *пользователей* компьютеров была обнаружена зависимость между жалобами на боли и тугоподвижность в плечевых суставах и жалобами на *утомление зрительное*. Проблема дискомфорта скелетно-мышечного аппарата пользователей видеотерминалов связана с тремя осн. факторами: *повторяющимися движениями при работе с клавиатурой*; *неправильной рабочей позой*; *неподвижностью*. С.р. у пользователей связана и с отсутствием подлокотников или *неправильной конструкцией спинки рабочего сидения*.

Особенно важно учитывать опасности С.р. при использовании компьютеров в сфере образования. Желательно оснащать рабочие места *обучаемых* специализированной компьютерной мебелью, соответствующим образом проектировать деятельность *обучаемого* с компьютером.

В.М.Бондаровская, И.Ю.Никонова.

STEAMER (англ. steamer — паролод) — имитационно-моделирующая тренирующая *программа учебного назначения*, предназначенная для *обучения* морских офицеров задачам управления корабельной паровой установкой. Осн. педагогическая цель системы по замыслу авторов состояла в создании у *обучаемого* полного представления о поведении паровой установки в различных ситуациях и развитии у него “чувства” состояния паровой установки. Обучение в первую очередь направлено на формирование *понятий*, а затем на выработку навыков *принятия решений*. Моделирование работы паровой установки основано на разработанной ранее *матем. модели* и ее визуализации с помощью средств *машинной графики*.

Для взаимодействия с обучаемым STEAMER использует два терминала: на одном представлена графическая модель паровой установки, а другой выводит текущие характеристики модели и учебную информацию. Вначале паровая установка представлена визуальной моделью самого высокого уровня, состоящей из 20 компонентов и потоков между ними. Каждый компонент представляет собой целую подсистему. Обучаемый может с помощью мыши (см. *Манипулятор ручной*) выбрать любую подсистему и, выдав команду "РАСКРЫТЬ", получить диаграмму данной подсистемы. Действуя таким образом, обучаемый может выбрать и увидеть работу любого, даже самого мелкого, компонента. Кроме того, манипулируя вентилями, задвижками, изменением температуры и т.п., он может изменить условия функционирования паровой установки. Все показатели работы установки выводятся на второй дисплей в виде показателей манометров, которые доступны морскому офицеру при управлении реальной установкой. Система может отвечать на вопросы обучаемого и хранить протоколы диалога с системой.

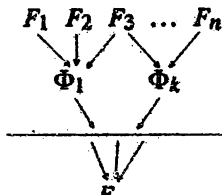
В.А.Петрушин.

СТРАТЕГИЯ — доминирующая тенденция в интеллектуальном поведении субъекта, решающего задачу (определение В.А.Моляко). Предполагает умение поставить и проанализировать новую задачу, осуществить поиск наиболее подходящей гипотезы решения и само решение. Термин "С." соотносится с широко распространенными терминами "метод решения", "способ", "прием" и др., означающими совокупность (последовательность, систему) действий (*операций*, шагов и т.п.), обеспечивающих решение задачи. Иначе говоря, "метод", "способ" и другие подобные термины относятся к процессу решения задачи в его реализующей, исполнительской части. Очевидно, эта последовательность действий также должна иметь основание. Механизмом, управляющим процессом решения, и является С. Описать ее можно в виде системы средств и рекомендаций по их преобразованию.

С. в наиболее развитом виде формируются у профессионалов, во многом определяя индивидуальный стиль их деятельности, общие подходы к постановке и решению новых задач. Практического работника обучают так наз. начальным (рабочим) С., а затем профессионал в ходе приобретения практического опыта преобразует их в более мощные и эффективные *процедуры*, обогащенные индивидуальной подсистемой средств профессиональной деятельности, индивидуализированными приемами работы с ними. Этот психологический принцип распространяется и на процесс внедрения компьютера в профессиональную деятельность.

М.Л.Смульсон.

СТРАТЕГИЯ ЛОГИЧЕСКОГО ВЫВОДА — алгоритм построения вывода в исчислении логическом. Вывод формулы F из множества посылок $A = \{A_1, \dots, A_n\}$ в исчислении J можно представить в виде дерева



, в котором каждый лист $F_i, i=1, \dots, n$, — это либо аксиома исчисления J , либо посылка из A , а все остальные узлы дерева получаются применением к.-л. правила вывода исчисления J к непосредственно предшествующим узлам (т.е. к “сыновьям” данного узла). Построение вывода является важнейшей задачей, т.к. это основное средство автоматического доказательства теорем. Однако само исчисление не определяет способ построения вывода; для этой цели формулируется С.л.в.

В рамках одного исчисления могут применяться различные С.л.в. Самое общее разделение их связано с двумя способами построения дерева вывода: от посылки к цели или от цели к посылкам. В обоих случаях возможны дальнейшие уточнения С.л. в. Это обусловлено тем, что на каждом шаге построения вывода в общем случае можно применить различные правила вывода, и даже одно и то же правило может применяться различными способами. Разные С.л.в. могут приводить к разным выводам одной и той же теоремы исчисления. Важнейшая характеристика С.л.в. — ее полнота: С.л.в. в исчислении J наз. полной, если с ее помощью можно построить вывод любой теоремы исчисления J .

Наиболее полно исследованы С.л.в. в так наз. резолюционном исчислении: здесь найдено большое количество полных стратегий. Однако программная реализация полной С.л.в. может оказаться весьма неэффективной (по времени и по требуемой памяти), поэтому на практике иногда используют и неполные стратегии. Характерный пример — стандартная реализация языка *Пролог*, фиксирующая такой способ вывода целевого утверждения (*запроса*), который в определенных случаях не завершается, несмотря на то, что утверждение является логическим следствием программы (т.е. *ответ* на запрос существует).

В.М. Антимиров.

СТРАТЕГИЯ УПРАВЛЕНИЯ УЧЕБНЫМИ ДЕЙСТВИЯМИ — общие правила выбора последовательности учебных целей и действий.

По степени самостоятельности обучаемых выделяют три группы С.у.д. Стратегии первой группы предусматривают “жесткое” программное управление действиями обучаемых. Второй группе стратегий соответствует более активное поведение обучаемых, которые отвечают на текущие *вопросы* и имеют возможность сами задавать вопросы, уточняя неясные выражения и *операции*. При этом обеспечивается текущий контроль степени усвоения пройденного материала и корректировка плана изучения предмета. В стратегиях третьей группы осн. работа преподавателя заключается в организации условий для активной самостоятельной деятельности обучаемых. Контроль усвоения осуществляется путем наблюдения за действиями обучаемых. В соответствии с характером изменения промежуточных целей выделяют четыре вида управления: программное, следящее, адаптивное и экстремальное. При программном управлении цель определяется функцией времени. При следящем управлении последовательность действий определяется некоторой эталонной моделью (напр., алгоритмом решения типовой задачи). При адаптивном управлении текущая цель выбирается на основе анализа состояния обучаемого в предшествующие моменты времени. При экстремальном управлении выбор текущих целей определяется последовательностью состояний, на которой показатель качества процесса обучения достигает экстремального значения.

Ю.И.Лобанов.

СТРЕСС (от англ. stress — давление, напряжение) — состояние общего напряжения организма в ответ на воздействие внешнего неспецифического фактора (стрессора). Часто термин “С.” используется для обозначения обширного круга состояний человека, возникающих в ответ на разнообразные экстремальные воздействия, для которых существуют общие закономерности в проявлении ответных реакций организма человека на различные стрессоры. Первоначально термин “С” возник в физиологии. Впервые был введен в 1936 канадским физиологом Г.Селье для обозначения неспецифического (физиологического) ответа организма на любое предъявляемое к нему требование. С. характеризуется (по Г.Селье) общим адаптационным синдромом. Патогенез С. можно разделить на три фазы: фаза быстрой активизации (реакция тревоги), фаза устойчиво-длительной активации (фаза резистентности), фаза истощения функции. Обычно выделяют простые физико-химические стрессоры (температура, шум, вибрация, физико-химический состав атмосферы, токсические вещества и др.) и сложные психологические и социально-психологические стрессоры (дефицит времени, риск, опасность, новизна ситуации, значимость решения определенной задачи, высокая степень сложности задачи и др.). Стрессовые ситуации делят на кратковременные и длительные, требующие адаптационной перест-

ройки функциональных систем человека. Длительное пребывание в стрессовых условиях часто вызывает сложные изменения физиологических, психологических и социально-психологических характеристик человека. С. может оказывать как положительное, так и отрицательное влияние на деятельность, вплоть до ее полной дезорганизации, что ставит перед исследователями задачу изучения возможностей *адаптации* человека к сложным условиям, а также прогнозирования его поведения в стрессовых ситуациях. Работа в стрессогенных условиях предполагает дополнительную мобилизацию психологических и физиологических резервов человека, что может иметь неблагоприятные последствия. Физиологические признаки С.: изменения в поведении (нарушения сна, аппетита), желудочно-кишечные расстройства, мышечные напряжения, функциональные нарушения сердечно-сосудистой системы.

Компьютер является сложным (социальным, социально-психологическим, непрямым физико-химическим и психофизиологическим) стрессором для человека. Расположение компьютера на *рабочем месте*, не соответствующее современным требованиям эргономики, изменение температуры и превышение шума в помещении часто выступают достаточно простыми, но важнейшими стрессорами при работе с ним. Неправильное распределение заданий (напр., в группах программистов) также может явиться существенным и оказывающим длительное влияние стрессором. Многие виды нарушений здоровья при работе с компьютером (головные боли, нарушения зрения, невроты, заболевания кожи, гинекологические заболевания, в т.ч. срыв беременности и др.) связаны со С. Головные боли выступают как вторичный синдром, напр., *астенопии*, и могут быть связаны со скелетно-мышечным дискомфортом при работе за компьютером. Невроты и психосоматические расстройства — наиболее распространенные виды заболеваний, обусловленные устойчивым влиянием стрессогенных факторов, связанных с работой за компьютером — неправильной организацией рабочего места, несоответствующим гигиеническим и эргономическим требованиям микроклиматом в рабочем помещении, неправильной организацией труда, дискомфортным программным обеспечением компьютера. Если стрессогенный фактор не будет вовремя устранен, такие временные расстройства здоровья могут переходить в устойчивые патологические состояния. Пути устранения С. и стрессогенных ситуаций: правильная организация рабочего места, соответствующие гигиеническим и эргономическим нормам микроклимат и шум в рабочем помещении, оснащенных компьютерами, психологическая и эргономическая обособанность программного обеспечения компьютеров.

А.С.Коваленко, Н.И.Повакель.

СТРУКТУРА ДИАЛОГА — систематически изменяемые аспекты *диалога*. Формируется на основании множества целей, свойственных определенному типу диалога, и фактически разделяет весь процесс взаимодействия на ряд *задач*: а) *задача учения* (ЗУ), состоящая в переводе некоторой части *знаний о предметной области* (ПО) из разряда неизвестных в известные для *обучаемого*; б) *задача контроля* (ЗК) *знаний*, устанавливающая соотношение между *знаниями обучаемого* и некоторым эталонным множеством знаний; в *обучающей системе* ЗК может решаться в двух типичных ситуациях: для проверки усвоения изученного материала и для диагностики ошибок при частично или полностью самостоятельной работе обучаемого с объектами ПО; в) *адаптации задачи* (ЗА), в результате решения которой выбираются последовательность, объем и форма изложения учебного материала и контрольных упражнений, наиболее приемлемые для каждого конкретного *пользователя*; г) *задача выполнения* (ЗВ), заключающаяся в исполнении системой действий, указанных пользователем; д) *задача обоснования* (ЗО), в результате решения которой пользователь получает объяснение действий, примеров или результатов, выработанных компьютером.

Осн. этапы диалога “преподаватель—обучаемый”: а) тестирование и определение уровня знаний обучаемого; б) передача обучаемому нового учебного материала; в) контроль усвоения; г) решение задач либо под контролем компьютера, либо в режиме *кооперативного взаимодействия* обучаемого и системы; д) получение помощи или справки; е) обоснование. Все эти этапы не обязательно присутствуют в каждом конкретном диалоге, а их последовательность определяется методикой *обучения* и инициативой пользователя. Напр., диалог в ЗВ может быть либо кратким, если пользователь вводит *команду* или *запрос*, которые однозначно интерпретируются и обрабатываются системой, либо развернутым, когда необходимые параметры и *данные* выясняются в процессе уточняющего диалога, если сообщение пользователя, инициирующее ЗВ, неполно или неоднозначно воспринимается системой.

Возможность различными способами организовать диалог, касающийся определенной темы, делает необходимым дальнейшую конкретизацию его структуры с учетом конкретных форм реализации. Такое описание наз. тематической С.д. Примеры тематической С.д.: а) учебно-информационный диалог (используется в ЗУ), в ходе которого обучаемому предъявляются порции материала и/или примеры, поясняющие данные определения или описания; обучаемый сообщает о возможности продолжения диалога или запрашивает более подробное изложение материала и дополнительные примеры; б) тренировочный диалог (применяется при решении ЗК и ЗА), направленный на выработку определенных навыков, использование и закрепление полученных знаний и определения уровня их усвоения; в) уточняющий диалог (используется в ЗВ), в результате которого

однозначно определяются параметры и данные для некоторого действия, которое либо выполняется, либо описывается в формальной нотации и сообщается пользователю; г) объясняющий диалог (используется в ЗО), возникающий, когда пользователь требует объяснить, почему система приходит к тем или иным выводам относительно объектов ПО.

С.П.Кудрявцева, М.В.Лёгкий, Н.А.Петрова.

СТРУКТУРА УЧЁБНОГО ДИАЛОГА — структура обмена сообщениями между *обучаемым* и *обучающей системой*. При обучении обучаемый и система поочередно становятся инициаторами обмена. Стадии *диалога*: установление контакта; распознавание (согласование) *кода* общения; уточнение цели диалога; согласование плана общения; обмен осн. *информацией*; оценка результата общения; завершение контакта. Структура конкретного диалога определяется последовательностью целей, соответствующих текущим намерениям партнеров.

Выделяют три этапа образования цели: формирование множества возможных целей; выбор предпочтительного (с точки зрения имеющихся критериев) множества целей; определение отношения к текущей цели (принятие или непринятие). При *учебном диалоге* цель обучаемого — получение максимума *знаний* за отведенное время или достижение требуемого уровня знаний за миним. время, цель системы — макс. удовлетворение познавательных потребностей обучаемого и обеспечение его учебных действий. На выбор текущих учебных целей существенно влияют результаты решений предыдущих проблем или *задач*. Отрицательные результаты могут снизить уровень *мотивации*. Динамика выбора целей зависит от соотношения уровней притязаний и успехов, а также ряда других индивидуальных особенностей.

Ю.И.Лобанов.

СХЕМА ДЕЙСТВИЙ ЛОГИЧЕСКАЯ — способ описания логической последовательности действий (*операций*). Является обобщением понятия логической схемы *алгоритма* и используется для описания законов функционирования целенаправленных систем. Под действием понимается процесс, управляемый рассогласованием описаний цели и текущего состояния управляемого объекта. Все разнообразие возможных действий в *информационной системе* можно получить, используя лишь четыре исходных типа обобщенных действий: D1: $M \rightarrow M$ — преобразование данных; D2: $M \rightarrow D$ — распознавание данных (дешифрация); D3: $D \rightarrow M$ — кодирование действий; D4: $D \rightarrow D$ — коммутация (связь) действий; здесь M — множество данных, обрабатываемых в системе, D — множество действий (*операций*) системы. На множествах указанных типов можно задать

операцию отождествления их полюсов (входов, выходов). В результате применения этой операции получаются составные схемы, относящиеся к одному из исходных типов.

Ю.И.Лобанов.

ТВОРЧЕСКАЯ ДЕЯТЕЛЬНОСТЬ — деятельность человека или коллектива людей по созданию новых общественно значимых материальных и духовных ценностей. Выделяют четыре стадии процессов Т.д. — подготовка, созревание, озарение и проверка. Центральный, специфически творческий момент — это озарение, т.е. интуитивное “схватывание” искомого результата.

Т.д. в работе с компьютером — культурно-историческое явление, связанное с компьютеризацией труда, предполагающее наличие у пользователя способностей, мотивов, знаний и умений, благодаря которым создается, напр., программный продукт, отличающийся новизной, оригинальностью, уникальностью. Передача компьютеру доступных формализации умственных операций резко повысила интерес к процессам Т.д., которые нельзя формализовать. В связи с этим особое значение приобретает разработка методов диагностики (в т.ч. и компьютерной психодиагностики), творческих способностей и стимуляции Т.д. При использовании компьютеров в обучении важную роль играет разработка методов и средств стимулирования Т.д. обучаемых.

Н.И.Повакель.

ТЕЗАУРУС (от греч. *θησαυρός* — сокровище, клад) — 1) Совокупность знаний, накопленных человеком или некоторым коллективом. 2) Словарь семантических отношений между основными понятиями некоторой области знаний (*предметной области*). Создание *информационной системы* связано с систематизацией терминологии соответствующей предметной области. Перечень семантических отношений устанавливается на основе анализа *текстов* данной тематики, имеющихся словарей и справочников. При отборе терминологии исходят из определённой практической задачи. В информационных системах Т. выступает как способ описания терминологии и как инструмент поиска. Т. с развитым аппаратом семантических связей может быть источником *ответов* на достаточно разнообразные типы *запросов*, т.е. служить своего рода базой данных (см. *Автоматизированный словарный комплекс учебного назначения*).

Т.Б.Андрусенко.

ТЕЗАУРУСНО-СЕТЕВАЯ МОДЕЛЬ предметной области — модель, построенная на основе описания объектов предметной области

(ПО) и *тезауруса* этой области. Включает: терминологическую лексико-семантическую систему, программные средства реализации различных режимов работы, сервисные возможности, предоставляемые *пользователю*. *Информация* об объекте ПО представляется по схеме: *понятие* — определение — семантические отношения данного понятия с другими (табл.).

Пример перечня семантических отношений

Отношение	Характеристика
ФУНКЦИЯ	Функциональное назначение
РОД	Концептуально более широкое понятие
ВИД	Концептуально более узкое понятие
ДЕЙСТВ НА	Иметь объект физического действия
ПОД ДЕЙСТВ	Обратное отношение
ВХОДИТ В	Быть частью физич. устройства
СОСТОИТ ИЗ	Обратное отношение
СИНОНИМ	Условный синоним
АССОЦИАЦИЯ	Ассоциативное отношение
ЗНАЧЕНИЕ	Значение имени или признака

Пример словарной статьи тезауруса:

МИКРОПРОЦЕССОР

ФУНКЦИЯ: обработка, управление

РОД: БИС

ВИД:

ДЕЙСТВ НА: данные, команды

ПОД ДЕЙСТВ:

ВХОДИТ В: микроЭВМ

СОСТОИТ ИЗ: АЛУ, дешифратор команд, регистр, схемы управления, шина

СИНОНИМ: интегральная схема

АССОЦИАЦИЯ: структурная схема микропроцессора

ЗНАЧЕНИЕ:

Вся совокупность объектов ПО, связанных с понятием МИКРОПРОЦЕССОР по указанным отношениям, должна быть отнесена к учебным целям планируемого обучающего диалога. Отношение РОД ориентирует деятельность обучаемого на изучение основ классификации объектов ПО. Отношение ФУНКЦИЯ определяет функциональные свойства (назначение) конкретного объекта. Отношение ВХОДИТ В определяет цели, связанные с изучением данного объекта в составе объекта более сложной структуры. Отношение СОСТОИТ ИЗ определяет составные части объекта, позволяет описывать его архитектуру на композиционной основе. Отношение ДЕЙСТВ НА

задает цели, связанные с изучением функциональных свойств объекта. В примере с МИКРОПРОЦЕССОРОМ его функциональные свойства ОБРАБОТКА, УПРАВЛЕНИЕ применимы к ДАННЫМ, записанным в ОЗУ, и к набору реализуемых КОМАНД. Отношение СИНОНИМ позволяет увязать объекты, одинаково интерпретируемые в рамках данной ПО. Отношение АССОЦИАЦИЯ реализует цели по закреплению умения связывать объекты ПО, напр., за счет определения формы представления и/или описания типа модели объекта. Отношение ВИД устанавливает учебные цели по определению данного объекта как объекта более узкого класса. Отношение ПОД ДЕЙСТВ определяет цели, связанные с изучением функциональных свойств объекта, находящегося под воздействием другого объекта. Отношение ЗНАЧЕНИЕ ориентировано на изучение правил означивания — именованного объекта в соответствии с его классификацией, функциональными свойствами, количественными и качественными параметрами.

Все описанные типы отношений определяют множество целей, которые необходимо реализовать в учебной деятельности обучаемых. Напр., отношение СОСТОИТ ИЗ определяет список объектов, составляющих архитектуру микропроцессора. Для решения задач по проектированию микропроцессора необходимо включить в деятельность обучаемых манипулирование объектами следующего списка: СОСТОИТ ИЗ: АЛУ, дешифратор команд, регистр, схемы управления, шина, причем для обеспечения наглядности обучаемому должны быть доступны определения понятий и их графические модели. Пример порождения цепочек понятий тезауруса. Преподавателю необходимо спланировать деятельность обучаемого при изучении объекта МИКРОПРОЦЕССОР по отношению СОСТОИТ ИЗ. Каждое понятие, стоящее в позиции данного отношения, "раскручивается" вглубь по данному отношению. Тем самым линейное представление преобразуется в иерархическое:

Ø МИКРОПРОЦЕССОР

- 1 АЛУ
- 2 ВХОДНОЙ ПОРТ
- 2 ВХОДНОЙ ПОРТ
- 2 ВЫХОДНОЙ ПОРТ
- 1 ДЕШИФРАТОР КОМАНД
- 1 РЕГИСТР
- 1 СХЕМЫ УПРАВЛЕНИЯ
- 1 ШИНА

При переходе от тезаурусного представления связи понятий к сетевому цепочки рекурсивной связи понятий выступают исходными элементами для построения семантической сети.

Тезаурус является эффективным средством организации знаний о ПО. Правила манипулирования объектами в зависимости от приложений могут быть различными (*задачи* диагностики, проектирования и пр.). Тип диалога выбирается в соответствии с типом решаемой задачи. Этапы конструирования объектов, правил и композиционной схемы диалога в значительной степени независимы друг от друга.

Т.Б.Андрусенко, А.Е.Стрижак.

ТЕКСТ (от лат. *textum* — ткань, связь, построение) — объект и результат обработки в *информационных системах* (входной и выходной Т.). В компьютерной среде Т. можно разделить на три обширные группы: Т. программ, Т. данных и Т. диалога. Т. диалога (протоколы) представляют несколько осн. разновидностей диалога в человеко-машинных системах, напр., диалог с *операционной системой*, диалог при первичной загрузке базы данных (БД), диалог при пополнении БД, диалог при проектировании конкретного варианта информационной системы. Указанным разновидностям диалога соответствуют, как правило, смешанные Т., содержащие, наряду с элементами естественного языка, специальные команды, а также элементы Т. программ и данные. Т. человеко-машинного диалога на естественном языке выступает как фиксированная или прогнозируемая в композиционном плане последовательность коммуникативных ситуаций (передаваемых соответствующими языковыми средствами в пределах главной темы диалога), выполняющая определенную прагматическую функцию.

Т.Б.Андрусенко.

ТЕКСТОВЫЙ РЕДАКТОР — семейство программ, предназначенных для ввода текста в файл и обработки этого текста. Как правило, позволяют использовать вычислительную машину как первоклассную пишущую машинку. Различают Т.р. строчные и экранные. Строчные редакторы (редакторы) трактуют файл как совокупность текстовых строк и позволяют производить различного рода манипуляции только с конкретной строкой. Функциональные возможности их весьма ограничены; в большинстве случаев позволяют вводить текст в файл и осуществлять несложное редактирование его. Экранные редакторы (текстовые процессоры) являются наиболее распространенными программами в данном семействе; общее их свойство — возможность работы со всем текстом, фрагмент которого виден на экране монитора. Текстовые процессоры обладают очень широким спектром функциональных возможностей, реализуемых нажатием одной клавиши или последовательности клавиш. Общие функции текстовых процессоров: ввод текста в файл и чтение

текста из файла для последующего редактирования; перемещение по тексту вверх/вниз по строкам/страницам и вправо/влево по символам/словам, в начало/конец строки; удаление/вставка символов/строк; выделение фрагмента текста с целью удаления/копирования/переноса в другое место текста; контекстный поиск букв/слов и контекстная замена букв/слов во всем тексте или в определенном фрагменте; чтение текста из другого файла с целью вставки его в редактируемый текст и запись выделенного фрагмента текста в файл и т.п. Текстовые процессоры вводят текст в файл, как правило, в формате *кодов ASCII* (Американский стандарт кодов передачи информации) и в этом виде являются полностью совместимыми, т.е. текстовый файл, подготовленный при помощи одного из текстовых процессоров, можно прочитать и изменить другим текстовым процессором, обработать *компилятором* (если текст—исходный текст программы) или *операционной системой* (если текст—команды операционной системы). Исключением из этого правила является текстовый процессор ChiWriter, формирующий файл с текстом в несколько ином, “своем” формате. Последующее редактирование данного текста возможно только при помощи ChiWriter’a. Еще одно существенное различие текстовых процессоров — отсутствие или наличие многооконности (см. *Многооконный интерфейс*). Текстовые процессоры отличаются также дополнительным набором тех или иных сервисных функций, и выбор конкретного из них зависит полностью от потребностей *пользователя* и возможностей его вычислительной системы.

Для пользователей из стран с русским алфавитом к таким возможностям в первую очередь относится наличие *символов кириллицы* на его компьютере. Для компьютеров с CGA-контроллером и монитором — наличие кириллицы в постоянном *запоминающем устройстве* (ПЗУ) знакогенератора монитора, для компьютеров с EGA-контроллером и монитором — наличие программы-драйвера, загружающего символы кириллицы в ОЗУ контроллера. Эту проблему решает текстовый процессор ЛЕКСИКОН (разработан на вычислительном центре РАН), позволяющий вводить тексты с кириллицей на компьютерах, не имеющих ПЗУ с кириллицей или драйвера загрузки кириллицы. ЛЕКСИКОН предоставляет дополнительные возможности: ввод/редактирование до десяти различных файлов в десяти окнах; выделение строчных и блочных фрагментов в тексте и перенос/копирование их из окна (файла) в окно (файл); форматирование текста; расстановка страниц в тексте; подготовка оглавления текста; использование различных шрифтов при написании и печати текста; три типа печати (грубая, средняя и качественная) и т.п.

Текстовые процессоры, входящие в состав *систем программирования TurboСi, TurboПаскаль и TurboПролог* фирмы Borland

International Inc., хотя и позволяют работать только с одним окном, но ориентированы в большей степени на ввод исходных текстов программ на языках *Си*, *Паскаль* и *Пролог*. При компиляции введенного текста и обнаружении в нем синтаксических ошибок эти текстовые процессоры автоматически устанавливают указатель на строку с ошибкой, текст-сообщение компилятора относительно которой выделен в окне сообщений под окном редактора. Это несомненно облегчает *отладку программ*. Текстовый процессор, входящий в состав системы SideKick той же фирмы, при запуске на выполнение становится резидентным в памяти компьютера и может существенно ускорить работу по написанию и отладке программ вне среды Турбо-продуктов. Напр., при работе с Microsoft C редактировать файлы можно с помощью резидентного SideKick, компилировать — запуская утилиту MAKE Microsoft C, сообщения компилятора можно наблюдать на экране монитора, уменьшив окно текстового процессора и расположив его в удобной части экрана. Кроме того, в состав системы SideKick входит калькулятор (двоичный/десятичный/шестнадцатеричный), который можно активизировать из текстового процессора и использовать для расчетов, таблица “активных” (реализованных в ПЗУ или загруженных в ОЗУ) кодов ASCII, электронная записная книжка и телефонный вызов. Текстовый процессор ChiWriter предназначен для подготовки научных статей, документаций и позволяет использовать вычислительную машину как первоклассную пишущую машинку с двадцатью различными шрифтами. Он предоставляет возможность написания матем. символов, формул и текстов на различных языках, включения в текст графических изображений, подготовленных при помощи *графических редакторов* GraphicsPartner или GED. Существенным достоинством ChiWriter является то, что вводимый и редактируемый текст виден на экране монитора в том же виде, в каком он будет выведен на печать. В ChiWriter реализовано все множество необходимых действий над текстом, которое может понадобиться при подготовке статьи или документации: перемещение по тексту; форматирование текста; переключение шрифтов; добавление и уничтожение подстрок (над- и подуровней при написании формул, таблиц, блок-схем); синхронизация и асинхронизация подстрок; динамическое изменение интервала между строками; разделение текста на страницы; поиск и замена символов/слов в тексте; выделение фрагмента, уничтожение, перенос или копирование его; возможность создания и использования ключевых записей, напр., написания различных знаков суммирования только по имени ключевой записи:

Σ Σ Σ Σ

возможность создания и печати заголовков страниц и сносок и т.п. Как указывалось ранее, текстовый процессор ChiWriter формирует

файл с текстом в "своем" формате; однако он может форматировать файл также на основе кодов ASCII. Эта некоторая "отчужденность" его от других текстовых процессоров является односторонней: ChiWriter позволяет осуществлять перевод любого ASCII файла, подготовленного другим текстовым процессором, в файл "своей" структуры. При выводе на печать можно использовать различное качество печати, при высшем — печать в шесть проходов на строку с практически неразличимыми точками при формировании символов на матричном *печатающем устройстве*. В состав процессора входят редактор собственных шрифтов, файлы со шрифтами, драйверы мониторов и печатающих устройств. Как правило, все текстовые процессоры имеют встроенную помощь по своим осн. функциям и самодокументированы, т.е. содержат файл "Руководство пользователя", подготовленный данным текстовым процессором.

А.А.Сахно.

ТЕЛЕКОММУНИКАЦИЯ — передача данных между *электронной вычислительной машиной* (системой) и удаленным устройством с помощью аппаратуры, обеспечивающей необходимые преобразования формата данных и управление скоростью передачи. Осуществляется по сетям связи, сетям переноса данных, сетям ЭВМ, где конечный *пользователь* удален от источника хранения и преобразования данных, а его аппаратные средства связаны с источником физическими каналами связи. Т. используется при создании глобальных и региональных сетей ЭВМ, территориально распределенных *систем коллективного пользования* ресурсами ЭВМ. Т. дает возможность контактировать территориально разбросанным группам *обучаемых*, проводить телеконференции, непосредственно участвовать в процессах формирования *банков знаний* по различным *предметным областям*, фондов *программ учебного назначения*, справочной и др. информации для региональных и национальных интерактивных *информационно-справочных систем* учебного назначения.

В.Н.Никулин.

ТЕОРИЯ РЕШЕНИЯ ИЗОБРЕТАТЕЛЬСКИХ ЗАДАЧ (ТРИЗ) — теория, основанная на выявлении закономерностей развития технических систем (автор теории — Г.С.Альтшуллер). В соответствии с ТРИЗ существуют объективные законы развития технических систем, которые можно познать и использовать для решения изобретательских задач. Выявлены две группы закономерностей функционирования и развития технических систем — статические и кинематические. К первой группе относятся критерии жизнеспособности технической системы: наличие системы или хотя бы миним. работоспособность ее осн. частей; сквозной проход энергии через систему к ее рабочему органу и согласование собственных частот колебаний (либо перис-

дичности действия) всех частей системы. Вторая группа характеризует направление развития независимо от конкретных технических и физических механизмов этого развития. Все технические системы развиваются: 1) в направлении увеличения степени идеальности (система идеальна, если ее нет, а функция осуществляется); 2) увеличения степени динамичности; 3) неравномерно — через возникновение и преодоление технических противоречий; 4) до определенного предела, за которым система включается в надсистему в качестве одной из ее частей. В рамках ТРИЗ разработан алгоритм решения изобретательских задач (АРИЗ).

Развитие технических систем в соответствии с ТРИЗ осуществляется путем разрешения технических противоречий. Типовые технические противоречия преодолеваются типовыми приемами. Осн. приемы устранения технических противоречий (с учетом их физического содержания) заложены в компьютерный банк приемов ТРИЗ. Среди них имеются, в частности, следующие принципы: дробления (разделить объект на независимые части, выполнить объект разборным, увеличить степень дробления объекта); асимметрии (перейти от симметрии формы объекта к асимметрии; если объект уже асимметричен, увеличить степень асимметрии); объединения (соединить однородные или предназначенные для смежных операций объекты, объединить со временем однородные или смежные операции); универсальности (объект выполняет несколько разных функций, благодаря чему отпадает необходимость в других объектах); “матрешки” (один объект размещен внутри другого, который в свою очередь находится внутри третьего и т.д.; один объект проходит сквозь полость в другом); “заранее положенной подушки” (компенсировать относительно невысокую надежность объекта заранее подготовленными аварийными средствами); непрерывности полезного действия (вести работу непрерывно, устранять холостые и промежуточные ходы); самообслуживания (объект должен сам себя обслуживать, выполняя вспомогательные и ремонтные операции, использовать отходы — энергии, вещества); замены механической схемы; использования гибких оболочек и тонких пленок; однородности (объекты, взаимодействующие с данным объектом, должны быть сделаны из того же материала или близкого ему по свойствам) и др. приемы (всего около 40). Широкое распространение получают различные компьютерные реализации ТРИЗ, которые следует рассматривать как одно из важных направлений применения и развития компьютерных технологий обучения, связанные с тренингом компьютерным.

М.Л. Смульсон.

ТЕРМИНАЛ. — устройство дистанционного ввода — вывода информации для оперативного взаимодействия пользователя с электронной вычислительной машиной или системой. Малые ЭВМ обычно связаны с несколькими Т., а большие могут быть подключены к сети, объединяющей сотни или тысячи их. Т. представляет собой два относительно независимых устройства: ввода (*клавиатура*) и вывода (*экран и/или печатающее устройство*). Различают Т. пассивные и активные. Пассивные Т. не перерабатывают информацию, их функции не изменяются в процессе эксплуатации. К ним относятся *дисплеи*, телетайпы и другие устройства отображения информации. Активные (интеллектуальные) Т. имеют средства переработки информации. Они строятся на базе *мини-ЭВМ* и *микро-ЭВМ* (в частности, персональных), берут на себя функции организации диалога, позволяют накапливать, редактировать данные и производить их первичную обработку.

И.А.Емченко, В.Л.Леонтьев.

ТЕРМИНАЛЬНАЯ СЕТЬ учебного назначения — сосредоточенная (в одном помещении) или распределенная (в нескольких помещениях учебного заведения) система терминалов; элемент системы коллективного пользования ресурсами высокопроизводительной ЭВМ.

В зависимости от удаления терминалов от центральной ЭВМ или вычислительного центра различают Т.с. локальную и удаленную. Локальная Т.с. единой системы ЭВМ состоит из подсистемы отображения информации типа ЕС-7920.01, ЕС-7920М, ЕС-7970, представляющей собой устройство группового управления ЕС-7922.01, позволяющего подключить до 32 терминалов типа ЕС-7927 (дисплей) и его модификации, а также ЕС-7934 (*печатающее устройство*). Информация от устройства группового управления передается к каждому из терминалов по кабелю. При этом терминалы от устройств управления, располагаемого в нескольких метрах от ЭВМ, могут быть удалены на несколько сот метров.

В удаленной Т.с. единой системы ЭВМ конечные пользователи расположены на значительном (километры и более) расстоянии от ЭВМ, и связь с сетью осуществляется по городским междугородным и др. линиям связи. Для удаленной обработки информации применяют средства и методы телеобработки. В вычислительном центре размещают аппаратуру передачи данных — мультиплексор передачи данных (МПД) или процессор телеобработки данных (ПТД), а у потребителя — устройство группового управления для удаленной работы ЕС-7920.11 с комплектом терминалов, схема подключения такая же, как и в локальной Т.с. В случае применения ПТД возможно подключение удаленных одиночных терминалов. В удаленной Т.с. информация от ЭВМ к потребителю проходит по

цепи: ЭВМ—МПД (ПТД) — модем—канал связи—модем—устройство группового управления —терминал. Существенный недостаток такой сети — низкая скорость передачи данных по каналу связи и, как следствие, значительное ограничение числа одновременно работающих пользователей (до нескольких человек) с целью сохранения малого времени реакции системы на *запросы* к ней конечного пользователя (*обучаемого*).

В Т.с. системы малых ЭВМ используется аппаратура передачи данных типа СМ-8514, позволяющая подключить до 15 терминалов типа СМ-7209, ВТА-2000 и др., технические характеристики которых отвечают требованиям внешнего *интерфейса* мультипроцессора. Т.с. учебного назначения могут быть реализованы и на других типах ЭВМ, обладающих необходимыми для этого программными средствами. Однако принципы их построения аналогичны приведенным выше. Вычислительная Т.с. в зависимости от производительности, характера решаемых *задач* и организации вычислительного процесса может иметь несколько устройств группового управления и соответственно несколько десятков терминалов, размещаемых, как правило, в дисплейных классах для организации групповых занятий обучаемых.

В.Н.Никулин.

ТЕСТИРОВАНИЕ ПРОГРАММЫ (от англ. test — проверка) — выполнение *программы* на специально выбранных входных данных (тесте) с целью обнаружения ошибки. О наличии ошибки свидетельствует расхождение между ожидаемыми входными данными программы для данного теста и результатами ее выполнения на тесте. Т.п. является одним из способов проверки правильности программы, но в отличие от *верификации программы* позволяет гарантировать ее корректность только при исчерпывающем тестировании, т.е. после проверки на всех возможных входных данных. Т.п. считается успешным, если позволяет установить факт наличия ошибки. Локализуются и исправляются ошибки при *отладке программы*. Поскольку исчерпывающее Т.п. невозможно, представляют интерес методы построения набора тестов, имеющего наибольшую вероятность обнаружения большинства ошибок при фиксированных затратах. Эти методы можно условно разделить на два класса в соответствии с тем, учитывается ли при построении тестов структура программы (так наз. методы “белого ящика”), либо тесты строятся только по ее спецификации (методы “черного ящика”).

Среди методов “белого ящика” наиболее известны: метод покрытия операторов, требующий, чтобы на множестве тестов каждый *оператор* программы выполнялся хотя бы один раз; метод покрытия решений (переходов), требующий, чтобы каждое направление передачи управления было проверено хотя бы один раз;

метод покрытия условий, требующий, чтобы на множестве тестов каждое элементарное логическое условие приняло значение "истина" и "ложь"; метод комбинаторного покрытия условий, требующий проверки всех возможных сочетаний значений элементарных условий в решении. Методы "черного ящика" носят в значительной степени эвристический характер и основаны на профессиональном опыте поиска ошибок. Наиболее распространенные из них — метод эквивалентной разбивки и метод анализа граничных условий. Метод эквивалентной разбивки заключается в разбивке пространства входных данных программы на конечное число возможно пересекающихся областей так, чтобы каждый тест из области был эквивалентен всем остальным из данной области в том смысле, что если тест обнаруживает ошибку, все другие тесты из данной области будут ее обнаруживать. Если же тест не обнаруживает ошибки, то ни один из тестов данной области, не принадлежащий одновременно к.-л. иной области, ее не обнаружит. Метод анализа граничных условий основан на предположении о том, что тесты, принадлежащие границам областей эквивалентности тестов, построенных по предыдущему методу, а также лежащие вблизи границ, наиболее продуктивны. При построении областей эквивалентности тестов по этому методу учитывается как пространство входных, так и пространство выходных данных.

Для локализации ошибки в процессе отладки используются тесты, построенные с целью более точной характеристики поведения программы. Тестирование пакетов прикладных программ также имеет свои особенности. Существуют специальные стратегии (напр., восходящего и нисходящего тестирования), определяющие порядок тестирования отдельных модулей и методы построения тестов, применяемые на каждом шаге.

И.Б.Романенко, Е.М.Синица

ТЕХНИЧЕСКИЕ СРЕДСТВА ОБУЧЕНИЯ (ТСО) — устройства, аппараты, системы, используемые в учебном процессе для совершенствования научной организации труда преподавателей и активизации познавательной деятельности обучаемых. Создаются, как правило, для группового обучения, но не исключено использование их и для индивидуального обучения. С учетом дидактической направленности выделяют шесть классов ТСО: звуковые, визуальные статические, визуальные динамические, *программированного обучения*, на базе ЭВМ и средства оргтехники.

Звуковые ТСО подразделяют на звукотехнические и лингафонные. Звукотехнические средства включают электрофоны, проигрыватели, магнитофоны, радиоприемники и др. средства, используемые для записи и воспроизведения звука. Лингафонная техника

обеспечивает разные виды коммуникативной деятельности обучаемых и преподавателя.

Визуальные статические ТСО делят на экранные и экранно-звуковые. Экранные ТСО состоят из диапроекторов различных типов, фильмоскопов, кодоскопов (графопроекторов) и т.п. В таких ТСО обязательно предусматриваются системы автоматического зашторивания окон и опускания экрана, дистанционного управления. К экранно-звуковым ТСО относятся диафонопроекторы и др. радиотехнические устройства.

Визуальные динамические ТСО включают кинопроекционные аппараты и киноустановки для демонстрации учебных кинофильмов, систему автоматического зашторивания окон и опускания экрана, телевизионную аппаратуру, аппаратуру замкнутых телевизионных систем, диапроекторы, магнитофоны, видеомагнитофоны, устройства для создания соответствующей освещенности аудиторий, "подсвета" телеэкранов и др. Учебное телевидение обладает специфическими особенностями. Намечается тенденция создания такого учебного телевидения, в котором все виды учебно-наглядных пособий можно было бы получить при помощи передвижных камер и специальных приставок с экрана телевизионной приемной аппаратуры. Использование видеомагнитофонов открывает широкие возможности для оперативного воспроизведения на экране фрагментов учебного материала.

Технические средства программированного обучения (ТСПО) подразделяют на контролирующие (позволяют в процессе обучения проверять знания обучаемых, усвоение ими программного материала и готовность к выполнению лабораторной работы, а также проводить зачеты, экзамены и т.д.) и обучающие. К контролирующим ТСПО относятся машины различных конструкций и марок с выборочным, результативным, конструируемым и другим вводом информации, а также ЭВМ. Чаще всего при машинном контроле знаний обучаемым выдаются карточки с вопросами или вопросы высвечиваются на экране. Иногда контрольные вопросы целесообразно высвечивать с помощью визуальных (напр., диапроекторов, кодоскопов) и соответствующих вспомогательных средств. Контролирующими ТСПО могут быть средства учета регистрации на ленте результатов проверки знаний обучаемых. Обучающие ТСПО содержат средства программированного обучения, которые не только контролируют, но и обеспечивают выдачу на экран обучающей информации. Это персональные обучающие устройства в виде тренажеров различного назначения, адаптивные и другие технические устройства с выдачей обучающей информации. При дидактическом обосновании необходимости в них могут входить визуальные технические средства, а также средства регистрации и документирования текущей учебной и исследовательской работы.

ТСО на базе ЭВМ подразделяют на учебные микропроцессорные стенды, учебные роботы с микропроцессорным управлением, специальные автоматизированные рабочие места, видеокomпьютерные обучающие системы; средства отображения данных ЭВМ для коллективного пользования (большие экраны). ТСО (чаще комплексы) на базе ЭВМ могут быть контролирующими и обучающими.

Средства оргтехники включают чертежные установки, копировально-множительную технику, аппаратуру передачи цифровых данных по каналам связи, процессоры передачи данных, средства внешней связи.

При выборе ТСО следует придерживаться принципов применения средств обучения.

И.И.Мархель.

ТЕХНОЛОГИЯ ОБУЧЕНИЯ — система средств обучения и способов их применения. По сути, Т.о. — это применение теории обучения в деятельности преподавателя и обучаемых, связующее звено между теорией и практикой обучения.

Термин “Т.о.” стал наиболее интенсивно употребляться с середины 70-х гг., причем в него вкладывался различный смысл. Напр., одна из наиболее узких трактовок закрепляла за этим термином применение различных машин в учебном процессе. Иное понимание означало применение научных знаний о процессе учения при построении учебного процесса. Еще одна трактовка объединяет две первые, существенно дополняя их системным подходом к построению учебного процесса. При этом под Т.о. понимают не отдельные элементы учебного процесса или технические средства, а их место и функции в учебном процессе, в системе обучения. Наконец, еще один смысл в понимании Т.о. близок к предыдущему, но не предполагает обязательной опоры на научный базис: функцию науки при таком подходе может выполнить человеческий опыт.

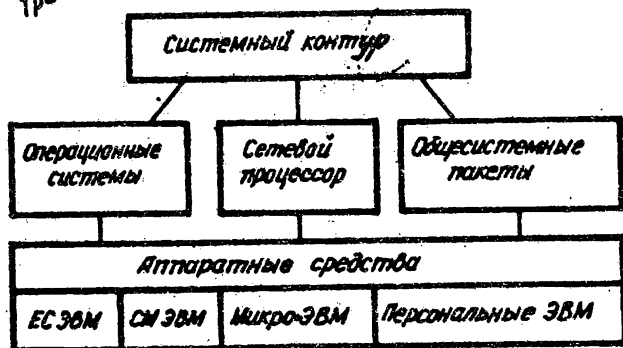
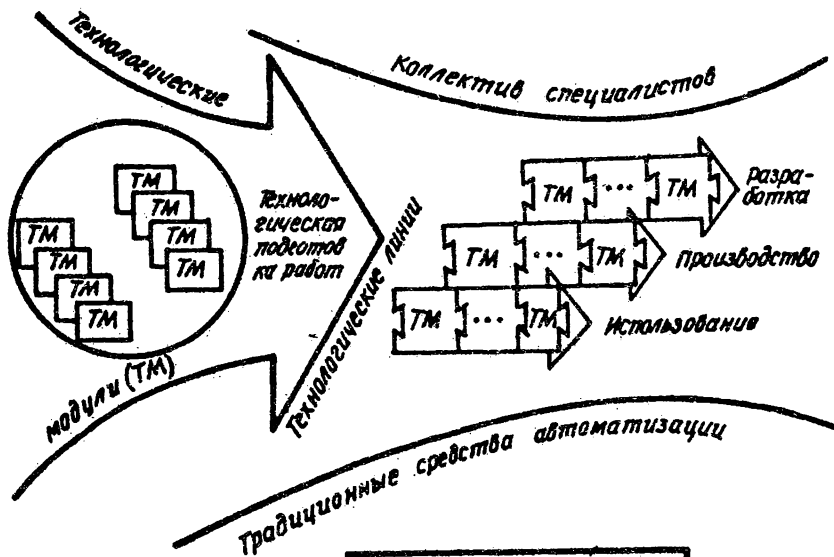
В наиболее распространенном смысле Т.о. — это совокупность технических, программных, учебных и методических средств, используемых при обучении. К техническим средствам обычно относят компьютер, сеть, плэйер, кодоскоп, фильмопроектор и т.д. К программным — программы учебного назначения, авторские системы, экспертные обучающие системы, к учебным — книги, видеокассеты, слайды, фильмы. Методические средства — частные методики обучения, наборы учебных задач, различные способы представления учебного материала (с помощью компьютера и без него). В связи с компьютеризацией процесса обучения резко возросли требования к технологизации научных знаний, в т.ч. знаний о процессе обучения. Возникла необходимость пересмотра и уточнения многих традиционных психолого-педагогических и дидактических понятий для использования их на технологическом уровне. Примером

может служить понятие метода обучения, или способа управления учебной деятельностью. Доказано, что традиционные трактовки этого понятия недостаточно технологичны. Уточнение его в свете требований технологии обуславливает, по мнению укр. психолога Е.И.Машбица, рассмотрение этого понятия как существенной детерминанты учебной деятельности, анализ реализации метода обучения в различных аспектах: в системе *обучающих воздействий*, в способе включения обучаемых в воспроизведение учебной деятельности, в "поле самостоятельности" обучаемых, в организационных формах обучения и в модальности обмена информацией между преподавателем и обучаемым.

Д.Ковач, Е.Д.Маргулис.

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ — 1) Организованная и оформленная в документах совокупность теоретических знаний, методов и приемов деятельности, программных средств ее автоматизации и регламентированного порядка применения знаний и средств, направленных на разработку, производство, внедрение (настройку, обучение пользователей) и использование программной продукции в заданных условиях и с заданными показателями качества. 2) Научно-техническая дисциплина, изучающая совокупность методов и средств, ориентированных на промышленную организацию и проведение работ по созданию и использованию *программного обеспечения*. 3) Учебная дисциплина, в которой изучаются знания, умения и навыки профессионального программирования коллективами специалистов разных категорий (не только программистов). Пример Т.п. — *Р-технология программирования*, разработанная в Институте кибернетики им. В.М.Глушкова АН Украины.

В структуре осн. понятий Т.п. (рис.53) базовым понятием является технологический модуль (ТМ) — функционально самостоятельная часть технологии разработки, производства и использования. *Интерфейсы* Т.м. соответствуют унифицированному соглашению о связях между ними. Примерами ТМ являются *текстовые редакторы, трансляторы, электронная почта, графические редакторы, методические и нормативные документы, отладчик, экспертная система* и т.д. Технологическая подготовка работ (разработки, производства и т.д.) — это совокупность регламентированных мероприятий, направленных на обеспечение организационных, технических, методических и других условий для проведения процесса программирования в соответствии с документацией. На этапе технологической подготовки работ формируются (программируются и komponуются из соответствующих ТМ) технологические линии, определяющие содержание технологического процесса и регламентирующие его выполнение. Т.п. основана не на искусстве



отдельных программистов, а на коллективных формах работы. Она составляет основу индустрии программных средств, которая отличается от любой другой индустрии прежде всего тем, что относится в большей степени не к сфере заводского тиражирования (производства) ранее сконструированных изделий, а к сфере их конструирования (разработки). Вместе с тем производство программных средств (процесс их создания конвейерным способом) также имеет место в программной индустрии. Этот процесс наз. еще автоматическим синтезом программных средств, или генерацией экземпляров. Сгенерированный экземпляр программного средства затем тиражируется (т.е. многократно копируется), поставляется пользователю и эксплуатируется.

Разработка и функционирование *программного обеспечения* базируются в Т.п. на жизненном цикле программных средств, который включает следующие этапы: 1) формирование потребности; 2) исследование аналогов; 3) анализ осуществимости разработки; 4) разработка технического задания; 5) реализация и исследование прототипа (макетирование); 6) определение спецификаций; 7) внешнее проектирование; 8) внутреннее проектирование; 9) конструирование (реализация); 10) тестирование и отладка; 11) документирование и разработка учебного обеспечения; 12) внедрение (генерация экземпляра, поставка, установка, обучение); 13) эксплуатация (использование и сопровождение). Этапы 4—11 наз. технологическим циклом разработки программных средств.

Анализ известных и широко распространенных технологий и методов программирования позволяет сформулировать осн. требования к Т.п.: обеспечение отторжимости программных средств от их разработчиков; обеспечение целенаправленной деятельности как профессиональных программистов и их коллективов с четко определенной промышленной организационной структурой, так и специалистов различной профессиональной ориентации (непрограммистов); гибкость, мобильность, комплексированность на основе программируемых технологических маршрутов, обеспечение планируемого повышения производительности труда программистов и разработки (производства и т.д.) программных средств с заданными показателями качества, стоимости и сроков выполнения; возможность реализации технологических процессов на существующих и перспективных вычислительных системах; наличие отдельных фаз, этапов, технологических операций и т.д., выделенных так, чтобы обеспечивать автоматизированное планирование и нормирование работ, контроль за ходом технологического процесса и качеством выпускаемой продукции; наличие комплекса нормативно-методических и правовых документов, определяющих как способ описания самой технологии, так и порядок проведения технологического процесса и форму документов, получаемых на каждом этапе; простота в освоении, наличие автоматически включаемых средств помощи и обучения, а также рекомендации по порядку внедрения Т.п., включая все уровни образования — школы, ПТУ, техникумы, вузы, ИПК. Современный взгляд на Т.п. базируется на трех принципах: возможность комплексирования набора ТМ в технологические линии различной проблемной ориентации, функционирующие в определенной технологической среде; развитие графического стиля программирования с использованием таблиц, рисунков, схем, диаграмм и т.п., визуализирующих программный проект и процесс его разработки (производства и т.п.); глубокая и сквозная автоматизация организационной деятельности коллектива специалистов на безбумажной

основе с формализацией процесса поручений работ и автоматическим контролем за их выполнением.

И.В.Вельбицкий, В.Б.Доронин.

ТЕХНОЛОГИЯ СОЗДАНИЯ И ПРОИЗВОДСТВА ВИДЕОКОМПЬЮТЕРНЫХ ПРОГРАММ — алгоритм, ставящий целью охватить необходимые виды деятельности, связанные с созданием видеокomпьютерных программ. Для его осуществления необходимо выполнить следующие действия.

1. Анализ учебного материала, определение отдельных тем, лекций, логических, понятийных и прочих связей между ними, т.е. структурирование учебного материала и выбор подходящей лекции, в которой применение *видеокomпьютерного рабочего места* было бы самым эффективным. Обычно этот вид деятельности осуществляется группой ведущих преподавателей по выбранному предмету, с консультантами-педагогами и методистами.

2. Разработку алгоритма и видеокomпьютерного сценария, специфические особенности и различия которого состоят в научно обоснованном разграничении элементов учебного материала, которые необходимо демонстрировать с помощью компьютерной графики, мультипликации и прочих абстракций, либо показать в натуре с помощью видеозаписи, диапозитивов или фотопластинок, либо в комбинации перечисленных вариантов. Алгоритм лекций можно наглядно представить с помощью граф-схемы, обеспечивающей возможность адаптации к интеллектуальным особенностям обучаемых. Адаптация реализуется полуавтоматически, преподавателем, оценивающим уровень знаний, интересы и профессиональную направленность аудиторий. В этот коллектив необходимо включить редактора, научных консультантов и др.

3. Составление режиссерского и съемочного плана, организацию видеооператоров, художников, мультипликаторов, программистов. С этого момента сценарий переходит в руки режиссера, и он играет ведущую роль. В режиссерском рабочем сценарии нужно определить, в каком плане будет снят каждый отдельный кадр, чтобы найти самый подходящий для данного момента и данной ситуации план. От правильного решения этой задачи в значительной степени зависит удачный монтаж и, в конечном счете, облик фрагментов и программы. Сопровождающий текст не должен быть абсолютно самостоятельным, пригодным для перепечатки. Когда текст потеряет связь с изображением, он перестает быть действенным. Он должен аргументировать то, что в данный момент происходит на экране. За весь период подготовки и разработки рабочего варианта режиссерского сценария необходимо активное содействие консультанта. Режиссер и консультант распределяют задачи для программиста, мультиплика-

тора, оператора, осветителя, звукооператора. Они помогают друг другу при монтаже до последнего кадра, до последнего звука.

4. Составление авторской программы, создание компьютерной графики и мультипликации, снятие видеофрагментов и подбор видеокадров, снятие диапозитивов, фотопластинок и др. После составления рабочего сценария и распределения задач начинается действительная работа по отдельным частям видеокомпьютерной программы. Авторскую программу составляют с помощью *авторской системы*, поддерживаемой *графическими редакторами*, *текстовыми редакторами*, мультипликационными и шрифтовыми редакторами, которые дают возможность наглядно представить учебный материал так, чтобы он был эстетичен и понятен. Под контролем режиссера создаются предварительные уточнения компонентов видеофрагментов, компьютерная графика, мультипликация, снимаются необходимые диапозитивы, подбираются соответствующие видеосцены, если их нельзя снять с натуры. Эта часть учебного материала связана с научным объяснением явлений — формулами, графиками и другими абстрактными *моделями*.

5. Монтаж и озвучивание видеокомпьютерных программ. После выполнения всех заданных в сценарии *задач* материал готов для монтажа, и начинает создаваться законченное произведение. Длина кадра определяется в зависимости от степени его сложности. Здесь нужно учитывать время, за которое обучаемый освоит обстановку. Предложенная картина должна давать ему возможность быстро и легко понять смысл и значение явления. Одновременно с этим аудитория должна быть удовлетворена эстетичностью картины. При объединении комментариев с оптическим и акустическими образами встречаются два в равной степени сильных фактора. Нельзя допускать, чтобы они мешали друг другу и чтобы один из них уменьшал воздействие другого. Они должны взаимно дополнять друг друга. Здесь снова в центре внимания авторская программа, а точнее — видеокомпьютерный редактор, связывающий компьютерные изображения с видеографическими согласно составленному и актуализированному алгоритму. Если учтены эти требования и соблюден сценарий, видеокомпьютерная программа готова, работа режиссера и консультанта окончена.

6. Экспериментальное внедрение программы, редактирование, актуализация и тиражирование дидактических материалов. Прежде чем видеокомпьютерная программа войдет в учебный процесс, необходимо провести экспериментальные лекции, в которых оценивалось бы ее действие в реальных условиях. Добавляются либо устраняются компьютерные или видеофрагменты, редактируется алгоритм, проводится последняя актуализация. После рекомендационного приема экспертным учебно-методическим и художественным советом программа тиражируется и, сопровождаемая соответст-

вующей методической и программной документацией, распространяется.

Г. Ж. Желев.

ТЕХНОЛОГИЯ ЭКСПЛИКАЦИИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ — совокупность способов и приемов уточнения описания знаний при создании программ учебного назначения. Осн. этапы экспликации знаний: 1) отбор фрагментов учебных текстов на естественном языке; ввод, редактирование и организация хранения в виде файловых систем (см. *Файл*) с помощью текстовых редакторов и графических редакторов; 2) классификация текстов (иерархическая, дискретная и др.) по формальным (текст, рисунок, таблица и т.п.) или по содержательным (определения, примеры, задачи, ответы и др.) признакам; организация информационно-справочных систем на основе баз данных (БД); 3) выделение отдельных объектов и установление связей между ними (локация и предикация объектов); определение отношений на совокупности объектов и процессов; организация фактографических систем на основе реляционных БД; 4) концептуализация данных — определение объектов и процессов как отношений на множестве признаков, правил интерпретации и семантического вывода; правила интерпретации задают смысл используемых предикатов (отношений); правила семантического вывода служат для определения новых семантических категорий в пределах имеющихся БД; организация дедуктивных БД — интеллектуальных банков данных; 5) прагматизация знаний — определение пространства возможных действий и правил принятия решений на основе результатов логического и семантического выводов; организация системы управления базами дедуктивных данных — организация интеллектуальных систем принятия решения.

Ю. И. Лобанов.

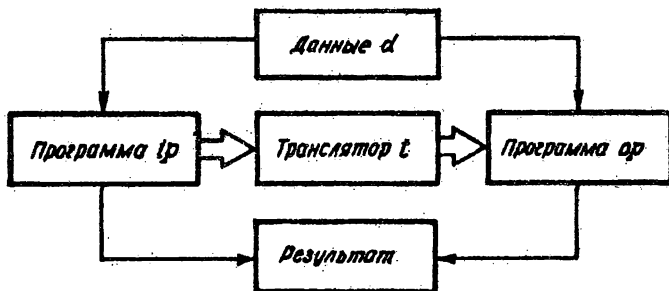
ТИРАЖИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ — получение копии программных средств, т.е. копии машинного носителя информации, на который записано программное средство, и копии эксплуатационной документации на программное средство. Выполняется организацией, ведущей фонд алгоритмов и программ, или любой другой организацией, занимающейся передачей программных средств. Если эксплуатационная документация находится на бумажном носителе, процедура получения ее копии является обычной процедурой получения копии документов. Копирование машинных носителей информации производится на специальных устройствах, предназначенных для этой цели и имеющих встроенные средства контроля операции тиражирования. При отсутствии такого устройства тиражировать машинные носители можно на компьютере, имеющем в своем составе устройства, позволяющие работать с носителями

необходимого вида, и *программное обеспечение*, обеспечивающее процесс тиражирования. Существует ряд методов и средств, обеспечивающих правильность Т.п.с., напр.: дублирование *информации*; применение специальных самокорректирующихся *кодов*; введение дополнительной информации, позволяющей проверить правильность копии (продольный, поперечный и циклический контроли). Для защиты машинных носителей от несанкционированного тиражирования применяют: введение паролей; аппаратные ключи защиты (лазерная "прошивка", встроенные электронные ключи и др.); саморазрушающиеся коды и другие аппаратные и программные средства.

В отличие от тиражирования магнитных носителей, содержащих записанную на них информацию бытового содержания (напр., музыку), при котором незначительные искажения звука допускаются, при Т.п.с. даже одна ошибка делает их неработоспособными. Совокупность способов тиражирования машинных носителей образует многовариантность этого процесса, поэтому необходимо наличие технологии тиражирования.

Г.П.Афинович, С.А.Дорожкин, В.П.Поляков.

ТРАНСЛЯТОР (англ. translator — преобразователь) — компонент *программного обеспечения* ЭВМ, предназначенный для преобразования *программ*, написанных на некотором входном языке, в функционально эквивалентные программы на другом, объектном языке. Как входной, так и объектный языки являются *алгоритмическими языками* высокого или низкого уровня (см. *Уровень алгоритмического языка*). Т. играет роль посредника между программами, написанными человеком, и машиной. Если L_1 — входной алгоритмический язык, на котором написана программа ip , а L_0 — объектный язык, то Т. представляет собой программу t , осуществляющую преобразование программы ip на языке L_1 в программу op на языке L_0 , так, что при любых входных данных d $ip(d) = op(d)$ (рис.). Если входным языком является язык высокого



уровня, а объектным — машинный язык, то соответствующий Т. часто наз. *компилятором*.

Систематическая разработка Т. для любого входного языка *L_i* из некоторого класса языков составляет содержание автоматизации *программирования*, а соответствующие *инструментальные средства* такой разработки получили название систем построения трансляторов.

В.И.Отенко.

ТРАНСФОРМАЦИОННОЕ КОДИРОВАНИЕ — тип *кодирования*, при котором полное описание элемента *данных* трансформируется в код. Один из осн. принципов (наряду с *ассоциативным кодированием*) получения кодов из полного описания. Пример Т.к.: STOCKHOLM можно было бы закодировать как STOCK. Если в результате применения простого правила получается несколько одинаковых кодов, то необходимо применять более сложное правило: напр., STOCKHOLM мог бы стать STKLM. Достоинством этого метода является то, что дано полное описание элемента и метод, с помощью которого оператор может получить код (и наоборот). Трансформационный код прост для запоминания и употребляется чаще, чем ассоциативный.

И.Ю.Никонова.

ТРЕНАЖЁР (от англ. train — обучать, тренировать, воспитывать) — техническое средство профессиональной подготовки человека-оператора, предназначенное для формирования и совершенствования у *обучаемых* умений и навыков, необходимых для управления каким-либо объектом или технологическим процессом. Реализует функциональную *модель* технической части системы человек-машина и ее взаимодействие с внешней средой. В Т. можно упрощать или усложнять *учебную задачу*, изменять параметры и масштаб времени процессов, создавать критические ситуации. Возможны остановка процесса управления в любой момент времени для обсуждения возникшей ситуации, анализа решений и действий обучаемого, многократное воспроизведение нужной ситуации для закрепления навыков. Т. обеспечивает постоянный контроль качества деятельности обучаемого, позволяет фиксировать его психофизиологическое состояние, производит регистрацию параметров управляемых объектов и процессов, ошибок, допущенных обучаемым. Эти *данные* используются для управления процессом *обучения* и оценки действий обучаемых. На Т. можно возлагать *задачи* профессионального отбора, первоначальной подготовки операторов, восстановления утраченных навыков, повышения квалификации, переучивания на новую технику, изучения новых приемов управления, решения исследовательских задач проектирования, анализа ситуаций, сложившихся на реальном объекте, для поиска наилучших решений и др.

Т. включает моделирующее устройство (МУ), пульт руководителя обучения (ПРО), *рабочие места обучаемых* (РМО), средства имитации (СИ) окружающей среды, аппаратуру контроля и оценки действий обучаемых (АКО), а также устройства связи между ними. МУ формирует учебную информационную модель и вырабатывает данные, используемые в ходе обучения. В качестве МУ используется обычно вычислительный комплекс с соответствующим *программным обеспечением*, в котором реализуются модели динамики объекта управления или технологического процесса. В нем же обрабатываются данные о процессе обучения и вырабатываются управляющие воздействия по корректировке заданий. На ПРО сосредоточены органы управления и контроля вычислительного комплекса, средства имитации, обеспечивающие ввод начальных условий, контроль за состоянием управляемого объекта и действиями обучаемых, задание аварийных ситуаций, выдачу подсказок и управляющих воздействий за отсутствующих операторов. Для накопления и выдачи данных о качестве работы обучаемых и их состоянии используются *печатающие устройства, графопостроители*, магнитофоны и другая аппаратура. В РМО входят органы управления и средства отображения информации. СИ окружающей среды могут представлять собой малоразмерные макеты местности, проекционные устройства, электронные имитаторы шумов, качающиеся платформы и др. Рабочие места обучаемых и руководства обучением оборудуются средствами связи. С помощью перечисленной аппаратуры руководитель обучения контролирует функционирование всей аппаратуры Т. и целенаправленно управляет ходом тренировки.

Классифицируют Т. по пяти осн. признакам: назначению (специализированные и комплексные), числу одновременно обучаемых (индивидуальные и групповые), степени приспособляемости к индивидуальным особенностям обучаемого (адаптивные и неадаптивные), степени автоматизации управления процессом обучения (автоматические, автоматизированные, неавтоматизированные), способу размещения (стационарные и бортовые). Можно классифицировать Т. также по виду обрабатываемых задач, способу реализации и т.д. В соответствии с последним признаком Т. можно разделить на аппаратные, программные и смешанные. Специализированные Т. предназначены для подготовки операторов по к. -л. специальности, напр., летчика, телеграфиста, гидроакустика. Комплексные Т. обеспечивают совместную подготовку взаимодействующих между собой операторов, напр., пилотов, штурмана и механика самолета; операторов котельной, турбинной и электрической части электростанции; капитана, рулевого, гидроакустика и механика рыболовного траулера и др. В автоматизированных Т. анализ достигнутого уровня обучения и допущенных ошибок, а также управление самим процессом обучения по выбранному критерию с учетом индивидуальных особенностей обучаемых производятся авто-

матически. Высокая степень автоматизации возможна в простых Т. для выработки навыков в элементарных операциях по вводу информации в компьютер, обнаружению сигнала на фоне помехи и т.п. В более сложных Т. автоматизация является средством, облегчающим, но не исключающим творческую работу руководителя обучения.

Экономия средств от применения Т. достигается за счет меньших затрат на их создание и эксплуатацию по сравнению с использованием для целей обучения реальной техники. Большой экономический эффект создается за счет сокращения времени и улучшения качества подготовки операторов систем человек—машина, поскольку применение Т. позволяет имитировать различные ситуации, которые на реальной аппаратуре воссоздать трудно или просто невозможно. Особенно эффективны Т. при подготовке операторов сложных и дорогостоящих систем, а также водителей транспортных средств.

В СССР впервые Т. были разработаны в Центральном институте труда в 30-е гг. на базе физических моделей и простейших счетно-решающих устройств для приобретения навыков по обработке металлов, а также для обучения летчиков гражданской авиации. В дальнейшем за счет применения матем. моделей, аналоговой вычислительной техники и аппаратных средств имитации существенно расширились возможности представления в Т. динамических процессов в различных объектах и системах. Созданы Т. для подготовки операторов радиолокационных и гидролокационных станций, кранового оборудования, по управлению движением судов и самолетов, для подготовки операторов химических заводов, тепловых и атомных электростанций. С появлением и использованием в Т. *электронных вычислительных машин*, телевизионной и лазерной техники был достигнут современный уровень развития тренажеростроения. На Т. проходят подготовку все экипажи космических кораблей, самолетов и судов, персонал крупных электростанций, операторы сложных технологических процессов, военные специалисты. Увеличивается производство Т. Создаются хорошо оснащенные учебной техникой и Т. центры подготовки, совершенствуются методы подготовки на Т. См. также *Тренажер по программированию*.

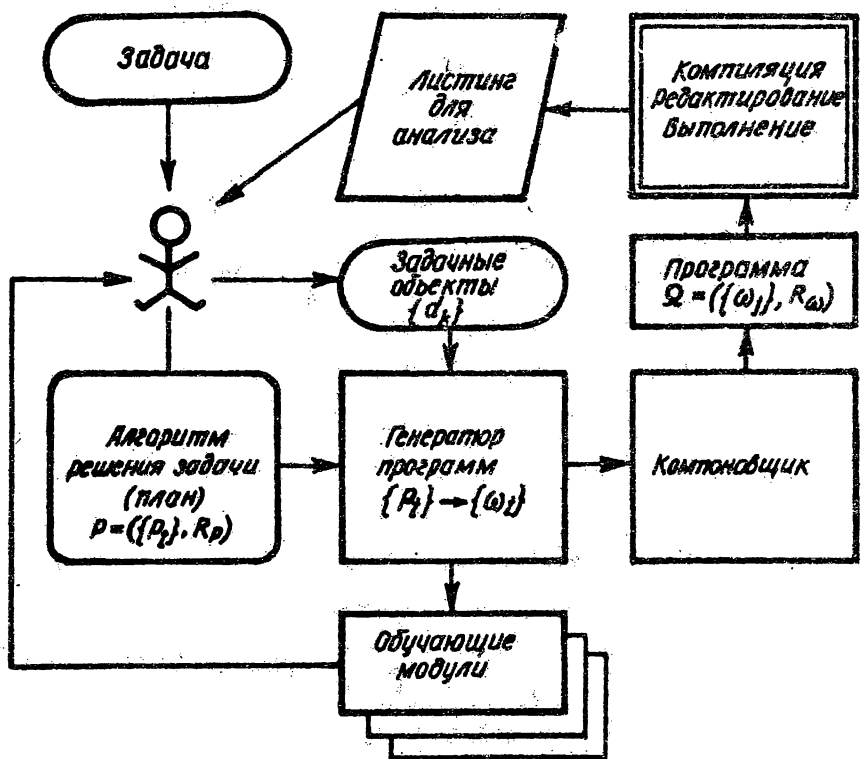
В.М.Зеленин.

ТРЕНАЖЕР ПО ПРОГРАММИРОВАНИЮ (ТП) — *автоматизированный учебный курс (АУК)*, предназначенный для выработки навыков использования операторов языка программирования при решении прикладных задач с помощью компьютера. Обеспечивает для обучаемого совместное с компьютером и под его управлением конструирование программ и немедленное получение результатов их выполнения при недостаточном знании языка программирования

(ЯП) и средств управления заданиями в *операционной системе* (ОС) компьютера. Тренировка по выработке навыков *программирования* производится на реальных задачах *пользователя*, что повышает мотивацию *обучения* по сравнению с использованием для этих целей традиционных АУК по дисциплинам программирования.

Основу ТП составляет генератор программ, осуществляющий автоматическую реализацию операции программирования и формирование программы на основе *алгоритма* решения задачи P , разрабатываемого обучаемым. Исходной *информацией* для генератора программ является план решения задачи $P = (\{p_i\}, R_p)$, где $\{p_i\}$ — множество компонентов плана, или алгоритмических операций (см. *Обучение алгоритмизации задач*), а R_p — отношение следования, задающее порядок выполнения $\{p_i\}$, напр., в виде блок-схемы алгоритма. По плану решения задачи P и конкретным задачным объектам $\{dk\}$, определяемым обучаемым, автоматически реализуется операция программирования и формируется программа $\Omega = (\{\omega_j\}, R_\omega)$, где $\{\omega_j\}$ — множество операторов изучаемого ЯП, а R_ω — отношение, задающее порядок выполнения операторов в соответствии с планом P . При ориентации ТП на ЯП высокого уровня (см. *Уровень алгоритмического языка*) задачные объекты $\{dk\}$, напр., имена переменных, списки форматов ввода—вывода, предикаты, задаваемые логическими выражениями, и др., запрашиваются у обучаемого в виде элементов и конструкций изучаемого ЯП. Если у обучаемого возникают трудности по определению задачных объектов на ЯП, подключаются обучающие модули, оказывающие помощь и подсказку в процессе совместного конструирования программы и обеспечивающие передачу необходимых ему знаний для самостоятельного выполнения требуемых действий. Примеры ТП для ЯП высокого уровня: режим *ДИПРОФОР*, реализованный в структуре *многофункционального автоматизированного учебного курса* (МФАУК) *АФРСДИТА*, предназначенный для выработки навыков программирования на языке *Фортран*; АУК *ДИАСО/ПЛ* — на языке *ПЛ/1*. При ориентации ТП на машинно-зависимые ЯП ассемблерного типа задачные объекты $\{dk\}$ определяются обучаемым в том виде, как они были заданы в плане P , и переформулировка их в термины ЯП осуществляется генератором программ, что исключает необходимость наличия в структуре ТП обучающих модулей. Примером ТП для машинно-зависимых ЯП является режим *диалогового программирования* на Ассемблере ЕС ЭВМ, реализованный в структуре МФАУК *АИДА*.

В структуру ТП (рис.) входят генератор программ, обучающие модули и компоновщик; компиляция, редактирование и выполнение программ осуществляются штатными средствами ОС компьютера. В зависимости от уровня подготовки пользователей ТП можно использовать как технологическое средство создания программного продукта



либо как учебно-тренирующее средство. Выбор того или иного способа применения ТП осуществляется по команде пользователя, которая интерпретируется отключением (или подключением) обучающих модулей. При использовании ТП для обучения и тренировки предполагается, что обучаемый поверхностно знаком или вообще не знаком с ЯП, однако имеет навыки алгоритмизации задач и может самостоятельно описать алгоритм решения задачи. Предварительная подготовка обучаемого может быть осуществлена традиционными методами либо с использованием средств компьютерной технологии обучения, как это имеет место при работе в системах АФРОДИТА и АИДА. Диалоговое взаимодействие обучаемого с ТП при совместном конструировании программы включает этапы: передача обучаемым имеющегося у него плана решения задачи для анализа и обработки средствами ТП; формирование средствами ТП промежуточной формы представления программы, соответствующей плану решения задачи; запрос со стороны ТП от обучаемого информации об объектах задачи и подключение при необходимости обучающих

модулей; формирование средствами ТП операторов программы, соответствующих алгоритмической операции плана и задачным объектам данной операции с предъявлением этих операторов обучаемому; генерация средствами ТП *текста* исходного модуля программы и предъявление его обучаемому; формирование компоновщиком задания для ОС; обработка задания и предъявление обучаемому распечатки с результатами выполнения.

Обучающий эффект при использовании ТП достигается за счет того, что в процессе диалогового взаимодействия обучаемый в любой момент времени может запросить помощь и получить со стороны ТП разъяснение или указание; ТП по результатам диагностики состояния обучаемого может отослать его к учебным и справочным материалам либо привести к подробному обучающему диалогу; осуществляется взаимодействие, при котором обучаемый под управлением ТП проверяет части формируемой программы; анализ обучаемым распечатки программы дает цельное представление о методах программирования, подкрепленное результатами счета. Использование ТП наиболее эффективно на начальных этапах обучения программированию и способствует сокращению времени, необходимого для предварительной подготовки обучаемых с целью допуска к самостоятельной работе на компьютере.

В.Д.Рынгач.

ТРЕНИНГ КОМПЬЮТЕРНЫЙ — вид тренинга, специально ориентированного на использование компьютера в целях формирования личности, развития мышления, внимания, памяти, творческого потенциала, способностей, выработки навыков саморегуляции и стрессоустойчивости, особенно в сложных трудовых и жизненных ситуациях. Широко известным Т.к. является система обучения *ЛОГО*, направленная на создание компьютерной среды, способствующей развитию мышления *обучаемых*.

Одна из разновидностей Т.к. — творческий тренинг — предполагает создание благоприятных психологических условий для реализации имеющегося у субъекта творческого потенциала, его развития и накопления в различных формах (в т.ч. и в неосознаваемой), а также рефлексивного анализа (осознания, фиксации) приемов и *стратегий* творческой деятельности. В этом смысле макс. жесткость и нефиксированность управляющих воздействий делает творчески тренинг противоположным по отношению к традиционному обучению. Однако при этом творческий тренинг следует отнести к организованному обучению (а не к стихийному научению), поскольку он ориентирован на целенаправленное формирование готовности к творчеству, творческого подхода к деятельности. Введение понятия “творческий тренинг” устраняет противоречие между обучением как системой управления, формирующей детерминированные и норми-

рованные системы знаний, умений и навыков, и творчеством, предполагающим новизну, нестандартность мыслительных действий и принятых решений. Творческий тренинг основывается на необходимых для решения задачи базовых знаниях, имеющихся у субъекта; в процессе тренинга должна сформироваться та психологическая структура, которая ответственна за превращение знаний в средства решения задач, т.е. в процессе творческого тренинга должна сформироваться стратегия творческой деятельности.

Осн. обучающим воздействием в этом процессе является система творческих задач. Осн. требования к подбору задач для творческого тренинга, форме их представления, организации процесса решения можно проиллюстрировать на примере тренинга в решении изобретательских задач. Известно, что в изобретательстве решение проблемы должно обладать новизной как для субъекта, так и для общества в целом. Новым обычно оказывается не только решение, но и постановка задачи. Полностью смоделировать эти особенности в учебных условиях невозможно, однако задачи в данном случае должны оставаться изобретательскими по своему предметному, техническому содержанию и психологической сути рассматриваемой ситуации, с требованиями принципиальной рационализации изобретения, что означает, в первую очередь, поиск проблемы и самостоятельную постановку задачи. Чтобы воссоздать это свойство в тренинговой ситуации, обучаемым предлагают тексты, содержащие нейтральное описание некоторой ситуации. Увидеть в этом описании изобретательскую задачу представляется самому решающему. Не имеет особого значения, на каком именно техническом материале поставлена изобретательская задача; главное, чтобы она предполагала самостоятельную постановку или, по крайней мере, доформулирование задачи, а также желательно несколько вариантов решения. Для оценки качества и эффективности решения тренинговой изобретательской задачи удобен метод экспертных оценок. Особое внимание акцентируется на двух творческих этапах решения — постановке задачи и становлении замысла, а само решение организуется на уровне вербализации замысла, графического эскизирования и мысленного (компьютерного) эксперимента. Одной из принципиальных психологических особенностей творческого тренинга на таких задачах является интерес и внимание к ошибке как к своего рода новому, творческому моменту, возникшему в процессе решения и потенциально ведущему не только и не столько к неудаче, сколько к новым поворотам и нестандартному видению задачи. Описанные выше психологические принципы творческого тренинга можно применить при индивидуальной, совместной (два-три человека в группе) и коллективной работе, для которой наиболее эффективно проведение творческой игры (см. Учебная игра). Для творческого тренинга наиболее применимы экспертные системы, что обусловлено их принципиальной ориентировкой на интеллектуальные и твор-

ческие возможности человека. Экспертная система предполагает и формирует *пользователя*, способного принимать ответственные решения самостоятельно, с помощью представленных ему знаний и рекомендаций более опытных экспертов. В этих условиях образец эффективной деятельности призван активизировать интеллектуальные возможности решающего задачу человека, стимулировать его нешаблонное, творческое мышление. Одной из наиболее разработанных является система творческого тренинга *КАРУС*. Проводятся исследования по построению системы Т.к. на базе *теории решения изобретательских задач*.

М.Л.Смульсон.

ТРЕНИРУЮЩАЯ СРЕДА — программно-технический комплекс, предназначенный для формирования и закрепления умений и навыков. Конечным результатом *обучения* с помощью Т.с. является формирование активных, рациональных, творческих умений применения *знаний* и навыков при управлении различными системами. Умения формируются в неразрывном единстве со знаниями и навыками, на их основе. С другой стороны, приобретенные умения позволяют легче освоить новые знания и навыки. Для приобретения не только умений и навыков, но и знаний служат *интегрированные обучающие тренирующие системы*.

Т.с. в общем случае представляет собой адаптивный *тренажер* с автоматическим управлением процессом обучения. Участие инструктора здесь минимально и проявляется в настройке обучающей подсистемы и, возможно, в корректировке ее параметров. Для обеспечения высокой степени автоматизации и адаптивности процесса обучения Т.с. должна включать: *имитационную модель объекта управления (ОУ)*; систему определения оптимальных (нормативных) действий; систему оценки действий тренируемого; систему диагностики состояния тренируемого; систему управления процессом обучения; систему регистрации процесса обучения.

Осн. методы обучения тренируемых в Т.с. — самообучение и автоматическое обучение. При самообучении тренируемый сам оценивает уровень своей подготовленности, выбирает соответствующие *задачи* и отрабатывает их по своему усмотрению. Большую помощь *обучаемому* здесь может оказать система автоматической оценки действий, которая немедленно или по завершении отработки задачи.. выдает результат тренируемому. Важным моментом является также возможность получения им необходимой помощи, для чего необходима система определения оптимальных действий. При автоматическом обучении система управления процессом обучения (или инструктор) формирует начальные параметры состояния ОУ, ставит задачу тренируемому, инициирует имитационную модель ОУ и начинает следить за деятельностью оператора. В

зависимости от его успехов и выбранной стратегии обучения изменяется темп выдачи заданий, их характер и сложность с целью максимально эффективного формирования и закрепления навыков и умений тренируемого. Уровень подготовленности и психофизиологического состояния тренируемого контролирует система диагностики. Такой контроль подразделяется на входной, текущий и конечный. Система регистрации заносит *информацию*, характеризующую как успешность самого процесса обучения, так и уровень обученности по задачам, в *базу данных* для возможного последующего контроля и разбора инструктором.

С.Т.Адрианов.

ТРЕНИРУЮЩИЕ ПРОГРАММЫ, программные тренажеры — программные средства подготовки специалистов, предназначенные для формирования и совершенствования у *обучаемых* профессиональных качеств, необходимых для управления материальными объектами, путем многократного выполнения отработываемых действий за *терминалом* универсального компьютера. Хотя Т.п. по способу реализации и используемым техническим средствам близки к традиционным *автоматизированным обучающим системам* (АОС), по целям и методам обучения они относятся к программным *тренажерам*. Однако различие между АОС и тренажерами в последнее время становится все более условным. Некоторые современные АОС включают тренирующие системы, а быстрое развитие адаптивных тренажеров на базе компьютеров приводит ко все большей степени *автоматизации процесса обучения* (тренажа), что характерно для АОС. Такие системы получили название *интегрированные обучающие тренирующие системы*.

Т.п. играют существенную роль в *компьютерной технологии обучения*. Особое развитие получили они в связи с появлением и совершенствованием *мини-ЭВМ* и *микро-ЭВМ*. Достоинствами Т.п. являются относительно невысокая стоимость и универсальность, т.к. их перенастройка сводится к замене Т.п. Они способствуют формированию интеллектуальных навыков восприятия, хранения и переработки *информации* специалистов, занимающихся анализом проблемных ситуаций, требующих оперативного мышления, сопоставления *информационных* и концептуальных моделей, принятием и исполнением решений. Главное в Т.п. не физическое подобие *рабочего места* тренируемого реальному, а макс. соответствие выводимой тренируемому информации реально протекающим во времени и пространстве процессам. Для устранения избыточности из нее обычно выделяют только необходимую для целей обучения часть. Значение Т.п. будет возрастать по мере того, как управление реальными эргатическими системами (система управления, составным элементом которой является человек-оператор) во все большей

степени будет осуществляться посредством общения оператора с компьютером, через него. Однако в настоящее время формирование исполнительных навыков управления большинством человеко-машинных систем с помощью Т.п. практически невозможно из-за неадекватности органов управления. Несмотря на это, разработано достаточно много программ для ПЭВМ, которые можно назвать тренирующими, хотя они больше относятся к игровым. С их помощью на экране графического дисплея детально отображаются приборная панель объекта управления, органы управления и вид внешней среды, открывающийся оператору, напр., из кабины автомобиля или самолета. Такие тренажеры часто являются адаптивными, т.е. автоматически изменяют в зависимости от успехов обучаемого стратегию и темп выдачи информации, генерируют экстремальные и нештатные ситуации, способствуют развитию навыков оперативного планирования действий оператора.

Типичной (и первой) Т.п. является *клавиатурный тренажер*. Создано множество таких программ для различных типов клавиатур и компьютеров. Прообразом таких Т.п. можно считать машину Паска (образца 1959) SAKI, которая представляла собой электромеханическую систему, использующую при выдаче очередного задания оптимальные стратегии в смысле *теории игр*, что позволяло значительно сократить время подготовки обучаемого.

С.Т.Адрианов.

ТРЕХКОМПОНЕНТНАЯ ЗАДАЧА — *познавательная задача*, в которой идёт речь о *процедуре* (Пр), переводящей некоторый предмет из начального состояния (НС) в конечное состояние (КС). В предмете любой Т.з. можно выделить три компонента (отсюда название), моделирующие соответственно НС, КС и Пр. В качестве примера рассмотрим известную шуточную задачу “превращения свиного уха в шёлковый кошелёк”. От этой материально направленной задачи следует отличать трёхкомпонентную познавательную задачу нахождения *способа решения* первой задачи, т.е. нахождения последовательности операций, которые следует осуществить, чтобы превратить свиное ухо в шёлковый кошелёк. Исходным состоянием предмета первой задачи является свиное ухо, а требуемым — шёлковый кошелёк. Исходным же состоянием предмета второй (познавательной) задачи служит описание превращения свиного уха в шёлковый кошелёк, не содержащее описания (достаточно полного) процедуры этого превращения (однако и свиное ухо, и шёлковый кошелёк описаны с достаточной полнотой и в этом смысле совершенно “равноправны” в отличие от их явного неравноправия в первой задаче). Требуемым состоянием предмета второй задачи является такое описание указанного превращения, в котором процедура его описана достаточно полно. Вместе с тем при анализе этой задачи,

при выделении компонентов её предмета могут указываться моделируемые ими начальное (НС — свиное ухо) и конечное (КС — шелковый кошелек) состояния, совпадающие с исходным и требуемым состояниями предмета первой задачи.

Классификация Т.з. (табл.) исходит из того, какие из объектов НС, Пр и КС известны, а какие неизвестны. На схеме Т.з. преобразования (рис.), которая является одной из конкретизаций общей схемы задачи, представлено различие состояний (НС и КС),

<i>Уровень модели</i>	<i>Требование задачи</i>	НС Пр КС		
	<table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">+</td> <td style="text-align: center;">+</td> <td style="text-align: center;">+</td> </tr> </table>	+	+	+
+	+	+		
<i>Уровень предмета</i>	<i>Предмет задачи в исходном состоянии</i>	НС Пр КС		
	<table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">+</td> <td style="text-align: center;">-</td> <td style="text-align: center;">+</td> </tr> </table>	+	-	+
+	-	+		
	<i>Исходное состояние</i>			

моделируемых компонентами предмета задачи, и состояний (исходного и требуемого) этого предмета в целом.

В учебных целях используют, как правило, задачи первых четырёх из указанных в таблице видов, однако и остальные могут быть не менее полезны для обучения. К шестому виду относятся, в частности предложенные укр. психологом Е.И.Машбицем задачи, где указывается к.-л. одна характеристика прямоугольного треугольника (напр., синус одного из острых углов) и требуется найти все характеристики, которые можно определить по этим данным.

Классификация трёхкомпонентных задач

№ п/п	Вид задачи	НС	Пр	КС	Пример формулировки задачи	Примечание
1	Задача исполнения	+	+	-	“Пользуясь таблицей синусов, найти синус $27^{\circ} 35''$ ”	
2	Задача преобразования	+	-	+	“Доказать, что $\sin 105^{\circ} = \cos 15^{\circ}$ ”	
3	Задача восстановления	-	+	+	“Пользуясь таблицей синусов, найти x , если $\sin x = 0,412$ ”	Решаются, в частности, при постановке диагноза
4	Задача построения (конструирования)	-	-	+	“Представить число 0,825 как к.-л. тригонометрическую функцию острого угла”	
5	Задача использования процедуры	-	+	-	“Дать пример нахождения синуса угла с помощью таблицы”	
6	Задача использования имеющегося состояния	+	-	-	“Указать значение какой-либо тригонометрической функции угла 48° ”	Решаются, в частности, в процессах прогнозирования

“+” - известный компонент; — неизвестный компонент.

В предмете познавательной задачи не обязательно выделять именно три компонента. Во многих случаях (в частности, применительно к матем. задачам, используемым в школьном обучении) полезно вводить в рассмотрение, наряду с указанными тремя компонентами, ещё и четвёртый (“базис решения задачи”), представляющий собой теоретическую или практическую основу для преобразования НС изменяемого предмета в КС посредством определённой процедуры (примером здесь может служить теорема, устанавливающая правомерность такого преобразования). Классификация четырёхкомпонентных задач даёт возможность, изменяя формулировки традиционных школьных матем. задач, получать задачи новых типов. Это позволяет существенно обогатить нахо-

дящийся в распоряжении учителя (или используемый в автоматизированном курсе) набор обучающих воздействий.

Г.А.Балл.

ТРУДНОСТЬ ЗАДАЧИ, уровень трудности задачи — мера фактического или прогнозируемого расходования ресурсов *решателя* (в т.ч. временных) на ее решение. Количественными характеристиками расходования ресурсов служат прежде всего его объем и интенсивность. В соответствии с этим выделяют, с одной стороны, интегральную Т.з. (трудоемкость) и, с другой, — дифференциальную Т.з., причем можно различать мгновенные значения последней и ее сред. значение для отрезка времени, в течение которого решается задача.

Для количественной оценки трудности решаемых людьми задач используют различные показатели — субъективные и объективные. Субъективные показатели отражают мнения либо самих субъектов, решающих задачи, либо экспертов (учителей, методистов, руководителей работ, проектировщиков и т.п.). В числе объективных показателей выделяют: а) физиологические (напр., изменения частоты пульса или дыхания, артериального давления и т.п.), характеризующие в основном дифференциальную Т.з.; б) поведенческие, описывающие объем расходования ресурсов (в т.ч. продолжительность процесса решения, количество потребовавшихся предъявлений исходных данных и т.п.); эти показатели характеризуют интегральную Т.з., но не всегда достаточны для ее адекватной оценки (напр., при малой вероятности достижения решения); в) оценивающие успешность деятельности (качество процесса решения задачи или достигаемого результата), в частности, вероятность самостоятельного успешного решения задачи субъектом, количество допущенных им ошибок, количество подсказок, которые пришлось дать, чтобы задача оказалась решена, и т.д.

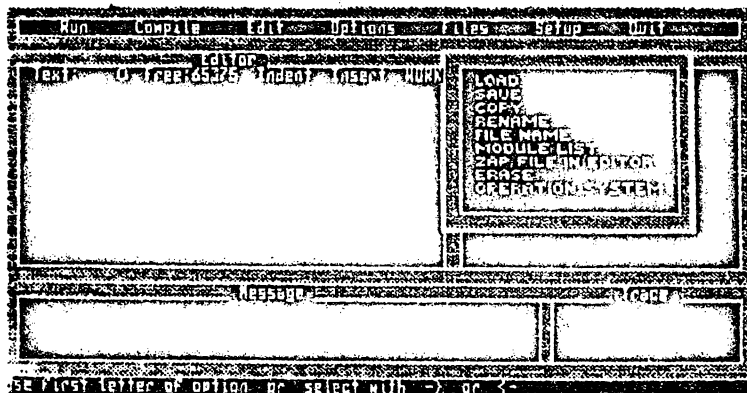
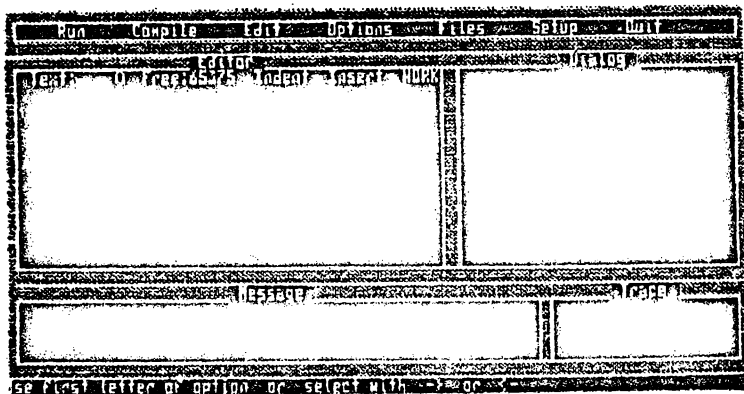
Иногда, оценивая Т.з., можно ограничиться к.-л. одним показателем. Так, трудность заданий, включаемых в педагогические и психологические тесты, оценивают обычно, исходя из вероятности их правильного выполнения или, точнее, количества испытуемых (которые входят в определенный контингент), правильно выполнивших задание. Вместе с тем во многих случаях целесообразно оценивать Т.з. совокупностью двух или более показателей (учитывая, напр., наряду с успешностью решения, время, затрачиваемое на его достижение). Получило применение (в частности, для оценки трудности доз учебного материала) представление Т.з. в виде функции двух переменных, возрастающей с увеличением продолжительности процесса решения задачи или иного показателя, характеризующего объем расходуемых ресурсов, и убывающей с повышением показателя успешности процесса решения. Этот подход,

однако, накладывает лишь весьма общие ограничения на вид функции, используемой в качестве меры Т.з. Большая определенность в установлении такой меры достигается при использовании метода, состоящего в том, что интегральная трудность φ задачи определяется объемом ресурсов, которые должен израсходовать субъект для достижения зафиксированного (эталонного) значения показателя успешности γ . Напр., в качестве меры Т.з. формирования навыка используют число упражнений, необходимых испытуемому для достижения требуемой эффективности работы. Если значение показателя успешности γ , которое удобно считать эталонным, практически не достигается, величину трудности φ часто можно найти путем экстраполяции зависимостей, установленных экспериментально в диапазоне значений показателя γ , не включающем его эталонного значения. Такая экстраполяция применяется в экспериментальных исследованиях эффективности *программированного обучения*. При решении задачи с помощью искусственных и человеко-машинных решателей в качестве показателей Т.з., наряду с временными затратами и показателями качества решения (точностью результатов и надежностью их получения), учитывается объем привлекаемых для решения ресурсов (напр., *памяти ЭВМ*). Понятия интегральной и дифференциальной Т.з. охватывают только некоторые аспекты многогранного содержания, которым обладает традиционное, интуитивное понятие Т.з. Специфическая сторона этого содержания, связанная в основном с новизной для субъекта требуемой интеллектуальной деятельности, находит объективное выражение в форме действия (в частности, материальной, внешнеречевой или умственной), необходимой субъекту для решения задачи. Иные его аспекты можно учесть с помощью таких понятий, как уровень проблемности и уровень нечеткости задачи.

Г.А.Балл.

ТУРБОПРОЛОГ — система логического *программирования* со встроенным *текстовым редактором*, со своим окружением, оконной поддержкой процесса создания *Пролог-программ*, развитой помощью по каждому из возможных режимов работы. Первые практические реализации Пролог-систем отражали осн. тезис логического программирования, требовавший тщательного, полного и корректного описания *задачи* на языке *Пролог*. Это описание является исполняемой логической *программой*. Сам же процесс логического вывода — это процесс исполнения данной логической программы. Теоретически *пользователь* полностью освобождался от представления *алгоритма* решения своей задачи. Однако практическое использование и дальнейшее развитие Пролог-систем на основе анализа результатов работы с ними приводило ко все большему нарушению классических основ логического программирования.

Одним из примеров “усовершенствования” и наращивания функциональности является система Т. фирмы Borland International Incorporated (США). При запуске система Т. выводит на экран стандартное расположение осн. окон (рис.а). Любой выбор из главного меню вызывает появление окна с подменю (рис.б).



Миним. конфигурация технических средств для системы: компьютер IBM PC/XT/AT или совместимый с ним; 384 Кб оперативной памяти ЭВМ; операционная система PC-DOS или MS-DOS версии 2.00 или последующих. Т. применяется для: производства прототипов практической проверки применения программ; контроля и управления производственными процессами через порты ввода/вывода компьютера; построения динамических реляционных баз данных; построения трансляторов; конструирования естественногоязыкового интерфейса; конструирования экспертных систем и их оболочек; конструирования символьных манипуляций для решения уравнений, диффе-

ренцирования, интегрирования и т.п.; доказательства теорем; построения интеллектуальных обучающих систем.

В языке системы Т., по аналогии с языками Си, Паскаль или Фортран, существуют стандартные типы данных: char, integer, real, string, file, symbol. Структура программы на Т.:

domains
persons, activity = symbol. (*описание типов данных*)
predicates
likes(persons,activity). (*описание предикатов *)
clauses
likes(elen,tennis). (*описание фактов *)
likes(john,football).
likes(eric,tennis).
likes(X,Y) if likes(elen,Y). (*описание правила *)
goal
likes(X,tennis). (*описание цели *)

Последовательность символов между (* *) является комментарием и никакого участия в выполнении программы не принимает. В языке Т. представлены арифм. операции, тригонометрические и алгоритмические функции, отношения и побитовые операции. Все функции определены на переменных типа integer и real, которые к моменту вызова данных функций должны быть связанными переменными, т.е. иметь значение.

Т. предоставляет пользователю ряд функций для организации ввода/вывода, работы с окнами, трехмерной графикой и звуком. Имеются средства для отладки и трассировки Пролог-программ, реализован доступ к некоторому подмножеству функций базовой системы ввода/вывода (BIOS) компьютера и его регистрам, возможность вызова из Пролог-программы процедур на ассемблера языке, языках ТурбоСи, ТурбоПаскаль. Наличие конструкций include и project позволяет организовать модульное создание Пролог-программ. Программы, созданные в Т., можно откомпилировать (см. Компилятор) и обработать программой LINK для получения программы, выполняемой отдельно, под управлением операционной системы. Однако следует помнить, что собрать готовую к выполнению вне Т. программу можно только тогда, когда все используемые в ней переменные будут связаны, т.е. к моменту редактирования получат определенные значения.

Т. обладает достаточно богатым набором средств для разработки мощных реализаций, но здесь уже пользователю не обойтись без представления алгоритма работы своей программы.

А.А.Сахно.

ТЮТОР (от англ. Tutor — наставник) — авторский язык программирования, предназначенный для написания обучающе-кон-

тролирующих учебных программ. Разработан в 1967 в США. Используется в *автоматизированных обучающих системах* Тенкор и PLATO. Алфавит языка определяется *клавиатурой* используемого компьютера. Переменные T. в системе PLATO обозначаются *символом* Y_n , где $n=0+99$, и могут принимать как строковые, так и числовые значения. Количество и наименование переменных в языке T. системы Тенкор определяются программистом.

Возможности T. для *программирования* обучающих курсов: вычисление арифм. выражений с использованием чисел с фиксированной запятой; условное выполнение *команд*; анализ и классификация ответов *обучаемого*; вывод графических изображений; индивидуализация *обучающей программы*; формирование HELP-последовательностей. Программа на T. состоит из множества единиц (рис.).

Единица всегда начинается с заголовка, который состоит из оператора начала единицы unit и параметра — имени единицы. За заголовком следует группа фрагментов, каждый из которых состоит из *операторов* обработки данных и блока *анализа ответа обучаемого*. Блок анализа ответа состоит из точки ввода *ответа*, группы операторов обработки данных и группы операторов классификации ответа. Группа операторов обработки данных располагает средствами формирования графических изображений на экране *дисплея*, вычисления арифм. выражений, установки функциональных клавиш и т.д. Связь между единицами осуществляется операторами передачи управления и функциональными клавишами. В качестве примера рассмотрим следующую программу:

```
*—unit:   Вопрос1
next      Вопрос2
at 10:5
write Кто изобрел электрическую лампочку?
arrow 12:5
answer Яблочкин
endarrow
*—unit:   Вопрос2
back      Вопрос1
at 10:5
write Кто изобрел телефон?
arrow 12:5
answer Александр Бэлл
at      14:5
write Очень хорошо!
endarrow
```

Программа состоит из двух единиц, которые наз. Вопрос1 и Вопрос2, причем, с какой единицы начинается выполнение программы, решает *пользователь*. Предположим, что программа начинается с единицы Вопрос1. Группа операторов обработки данных стирает экран дисплея, связывает функциональную клавишу NEXT с

Заголовок единицы

Операторы обработки данных

Группа операторов анализа ответа

Точка ввода ответа

Операторы обработки данных

Классификация ответа

Операторы обработки данных

Операторы обработки данных

единицей Вопрос2, устанавливает курсор в строку 12 позицию 5 экрана дисплея, выводит на экран *текст* вопроса. Блок анализа ответа начинается оператором *agrow*, выполняющим установку курсора и ввод ответа. Ответ обучаемого считается правильным, если он совпал со строкой, указанной в операторе *answer*. В этом случае выполнение единицы завершается, иначе выполняется передача

управления на оператор *arrow*. При завершении единицы ожидается нажатие функциональной клавиши *NEXT*, после которого будет выполнена единица *Вопрос2*. Единица *Вопрос2* выполняется аналогично единице *Вопрос1*. Есть лишь два исключения. Первое — при правильном ответе обучаемого на экран выдается реплика “Очень хорошо”. Второе — функциональная клавиша *back* связывается с единицей *Вопрос1*, связь функциональной клавиши *NEXT* с *Вопрос1* разрушается. Если при завершении единицы нажата клавиша *BACK*, то единица *Вопрос1* будет выполняться повторно. Если нажата клавиша *NEXT* - выполнение программы завершится.

В.А.Третьяк.

УЛЬТРАФИОЛЕТОВОЕ ИЗЛУЧЕНИЕ, ультрафиолетовые лучи (УФЛ) — электромагнитное излучение, занимающее спектральную область между видимым и рентгеновским излучением в пределах длин волн от 400 до 10 нм. Источником УФЛ являются, напр., видеотерминалы на электронно-лучевых трубках. Экран видеотерминала излучает УФЛ с длиной волны 400—315 нм и интенсивностью от 0,001 до 0,12 Вт/м. Такие уровни УФЛ не опасны для здоровья — они во много раз меньше допустимых параметров (общепринятая норма — 10 Вт/м). Соблюдение правил работы за видеотерминалом дает возможность пользователю свести к минимуму влияние УФЛ на его здоровье (см. *Рабочее место, Режим работы с видеотерминалом*).

А.С.Коваленко.

УНИВЕРСАЛЬНЫЙ ЯЗЫК — алгоритмический язык, предоставляющий набор средств для решения широкого спектра задач. Часто служит основой для создания компьютеров с новой архитектурой (*Лисп, Пролог, Смолток-80*). По способу представления и обработки данных У.я. разделяются на *императивные языки, функциональные языки и декларативные языки*. По технологической направленности У.я. подразделяют на *языки спецификаций, языки проектирования, и языки программирования*.

В.А.Третьяк.

УПРАВЛЯЮЩАЯ ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА — вычислительная машина для обработки информации, поступающей в процессе управления, и выдачи управляющей информации непосредственно на исполнительные органы или человеку-оператору. Управляющие ЭВМ классифицируются по назначению (для обучающих комплексов, аэрокосмических систем и т.п.), по принципам технической реализации (цифровые, аналоговые и гибридные), по возможностям применения (широкого назначения — для нескольких классов объектов и специализированные — для одного

типа объектов), по выполняемым функциям (централизованного контроля, советчики и др.).

В структуру управляющих ЭВМ входят преобразователи информации из непрерывного вида в дискретный на входе и из дискретного вида в непрерывный на выходе. Такие ЭВМ используются в качестве центрального звена *локальной вычислительной сети*. С их помощью осуществляется централизованный контроль за всей сетью в общем и за каждым компьютером в отдельности.

В.П.Козлова.

УРОВЕНЬ АЛГОРИТМИЧЕСКОГО ЯЗЫКА — одна из качественных характеристик *алгоритмического языка*. В зависимости от *контекста* имеет одно из двух возможных значений: 1) семантический разрыв между алгоритмическим языком, заданным в определении *алгоритма*, и рассматриваемым; 2) семантический разрыв относительно машинного языка конкретного компьютера. В первом случае под семантическим разрывом подразумевается некоторая качественная характеристика, выражающая сложность отображения выбранного алгоритмического языка в язык, заданный в определении алгоритма. Следовательно, термин “алгоритмический язык” высокого уровня означает достаточно большой семантический разрыв между рассматриваемым языком и выбранным определением алгоритма. Во втором случае семантический разрыв выражает сложность реализации рассматриваемого языка на машинном языке данного компьютера.

В качестве примера двойственности толкования термина “У.А.Я.” могут служить возможные его применения к языку *Смолток-80*. Он считается алгоритмическим языком высокого уровня согласно первому и второму значению термина для компьютеров традиционной архитектуры. Но для Смолток-ЭВМ он не может быть языком высокого уровня во втором значении термина, так как семантический разрыв в этом случае достаточно мал.

В.А.Третьяк.

УРОВЕНЬ СЛОЖНОСТИ ЗАДАЧИ — то же, что и *сложность задачи*.

УРОВЕНЬ ТРУДНОСТИ ЗАДАЧИ — то же, что и *трудность задачи*.

УСТОЙЧИВОСТЬ ИЗОБРАЖЕНИЯ — субъективный параметр, характеризующийся стабильностью во времени параметров изображения, выводимого на экран (*яркости*, геометрических размеров и

координат). Первым признаком, по которому пользователь субъективно оценивает качество дисплея, является У.и. Изображение должно быть полностью устойчивым, чтобы пользователь в любых условиях не замечал мерцания самого экрана или представленной на нем информации. Как только символ появляется на экране, он сразу же начинает угасать. Скорость угасания зависит от времени послесвечения определенного вида фосфора, использованного в дисплее. Чтобы символ на экране не исчезал, его нужно постоянно регенерировать. Если частота регенерации будет слишком низкой, это вызовет мерцание символа (см. Фликкер).

Для обеспечения У.и. в дисплеях на электронно-лучевых трубках рекомендуется применять виды фосфора с коротким или средним временем послесвечения; для дисплеев с негативным изображением (светлые символы на темном фоне) применяют частоту регенерации 50 Гц (фосфор P4 и P39) или 55—60 Гц (фосфор P31); для дисплеев с позитивным изображением (темные символы на светлом фоне) применяют частоту регенерации не менее 70 Гц. Иногда наблюдается вертикальное и горизонтальное периодическое смещение элементов изображения (дрожание изображения). Подобная нестабильность изображения при большой плотности информации на экране повышает утомление зрительное пользователя.

С.А.Миронченко, В.С.Пашковский.

УСТРОЙСТВА ВВОДА — ВЫВОДА ИНФОРМАЦИИ — *периферийные устройства*, с помощью которых пользователь может вести активный диалог с компьютером. Осуществляют ввод информации в компьютер и вывод из него результатов обработки, а также выполняют необходимые при этом преобразования данных из одной формы представления в другую.

По функциональному назначению У.в.—в.и. можно условно разделить на несколько классов: устройства ручного ввода (*клавиатура, манипулятор ручной*), средства визуализации (*видеомонитор, дисплей*), *печатающие устройства*, устройства ввода—вывода графической информации (*графопостроители, устройства кодирования*), *накопители, терминалы*, устройства речевого ввода—вывода. У.в.—в.и. подключают к компьютеру с помощью контроллеров, реализующих стандартные и специализированные алгоритмы сопряжения. К стандартным средствам связи компьютера с У.в.—в.и. относительно невысокого быстродействия относятся контроллеры, реализующие последовательные (RS-232 C, Стык C2) и параллельные (ИРПП, ИРПП/М, CENTRONIX) протоколы связи. Специализированные контроллеры (адаптеры) обеспечивают сопряжение с накопителями, управление отображением информации на экранах терминалов, подсоединение компьютера к локальным вычислительным сетям, связь с телефонными и телеграфными каналами, аппаратурой

передачи данных, сопряжение с объектами управления. Кроме технических средств для подключения У.в.—в.и. используют драйверы — специальные программы, ориентированные на управление конкретным типом устройств. Драйверы стандартных устройств образуют в совокупности базовую систему ввода — вывода (BIOS) и размещаются, как правило, в постоянных запоминающих устройствах (ПЗУ) компьютера. Драйверы специализированных устройств часто размещаются в ПЗУ, устанавливаемых на платах соответствующих контроллеров, либо загружаются с внешних носителей.

Существуют три осн. метода ввода — вывода информации, обеспечивающих взаимодействие процессора с У.в.—в.и.: программный, с прерыванием программы и с прямым доступом. Программный ввод — вывод осуществляется по инициативе процессора, и обмен данными с У.в.—в.и. происходит под управлением процессора в моменты времени, определяемые программой. При вводе — выводе с прерыванием программы инициатива принадлежит У.в.—в.и., которое подключается ко входу запроса прерывания процессора. Для того чтобы передать или принять данные, У.в.—в.и. изменяет уровень сигнала на этом входе. Процессор в ответ завершает выполнение текущей команды и переходит к выполнению программы, называемой подпрограммой обработки прерываний, с целью продолжения информационного обмена. Ввод — вывод с прямым доступом к памяти программируется процессором, но инициируется У.в.—в.и., и пересылка данных между У.в.—в.и. и памятью процессора происходит без к.-л. участия последнего.

*И.А.Емченко, В.П.Козлова, В.Л.Леонтьев,
З.Ш.Примакова, В.И.Сигалов.*

УТОМЛЕНИЕ ЗРИТЕЛЬНОЕ — функциональное состояние организма, выражающееся во временном снижении уровня зрительной работоспособности вследствие временных изменений оптических и мышечных структур глаза. При использовании видеотерминалов возникает в двух случаях: 1) при низком качестве технических и программных средств, неправильной организации рабочего места, нарушении эргономических и гигиенических требований к оборудованию рабочих помещений; 2) при нарушении режима работы с видеотерминалом, в частности при ежедневной длительной работе с экраном. У пользователей видеотерминалов возникают при У.з. возникают такие нарушения, как *астенопия*, функциональные изменения в зрительном анализаторе, патологические изменения зрительных функций.

Для пользователя работа с экраном на электронно-лучевой трубке связана с определенными специфическими особенностями: необычны размер, форма, контраст предъявляемых символов; поверхность экрана блестит, имеет отражения, вызывающие ослепление; часто возникают затруднения в регулировании расстояния между глазом

■ экраном. К функциональным изменениям в зрительном анализаторе пользователей относят изменения аккомодации, изменения конвергенции, гетерофорию, изменения рефракции и остроты зрения; изменения чувствительности к мельканиям, сужение зрачка.

Неустойчивость изображения на экране представляет собой фактор, влияющий на критическую частоту слияния зрительных мельканий. Сужение зрачка обычно связывают с явлениями отражений от экрана и необходимостью работать одновременно с негативным изображением (экран) и позитивным (учебник). С работой за видеотерминалом связаны два вида патологии зрения: катаракта и миопия.

В.М.Бондаровская, А.С.Коваленко, В.Г.Мартиросова.

УЧЕБНАЯ ДЕЯТЕЛЬНОСТЬ — вид человеческой деятельности, направленный на усвоение *обучаемым* знаний, умений и навыков, овладение им способами действий по решению различных классов *задач*. Особенностью У.д. является то, что ее цели и результаты состоят в изменении *обучаемого*, а не в изменении предметов, с которыми он взаимодействует. В процессе У.д. *обучаемые* воспроизводят реальный процесс создания людьми *понятий*, образов, ценностей и норм, воспроизводят в своем сознании те теор. богатства, которые человечество накопило и выразило в идеальных формах духовной культуры.

Различают ближайшие и отдаленные цели У.д. К ближайшим целям относятся усвоение определенных *знаний*, овладение конкретными умениями и навыками, решение отдельных *учебных задач*. Наиболее важные и специфические цели У.д. связаны с усвоением способов действий, инвариантных для *способов решения* некоторых классов задач. Отдаленные цели У.д. отражают развитие личности в процессе обучения и направлены на формирование способностей (прежде всего способностей к освоению человеческой культуры, “научить учиться”), определенных эмоционально-ценностных отношений к явлениям действительности (воспитание, умение решать разнообразие классы задач). Достижение целей У.д. обеспечивается системой учебных действий, зависящих от объекта усвоения, способа его представления, требуемого уровня усвоения и т.д. Формирование развитой У.д. — одна из важнейших целей *обучения*, достичь которую можно уже в младшем школьном возрасте.

Е.Д.Маргулис.

УЧЕБНАЯ ЗАДАЧА — *задача*, решаемая в процессе *учебной деятельности*. В обучении предлагаемые *обучаемым* У.з. выступают в качестве важных средств решения дидактических задач.

Существуют разные трактовки понятия “У.з.”: а) традиционная, согласно которой под У.з. понимается специфический вид задания,

даваемого обучаемым, — чаще всего такое задание, которое требует от них более или менее развернутых мыслительных действий; б) развитая в рамках задачного подхода к исследованию и построению учебной деятельности, исходящая из того, что эта деятельность, как и любая иная, имеет заданную структуру, т.е. представляет собой систему процессов решения задач, в данном случае учебных; в зависимости от процесса, играющего осн. роль в их решении, выделяют, наряду с мыслительными У.з. (должны занимать в *учении* ведущее место), также У.з. перцептивные, мнестические, имажинативные (активизирующие соответственно восприятие, память, воображение), *коммуникативные задачи* и др.; в) развитая в рамках разработанной в отечественной психологии концепции учебной деятельности; здесь рассматриваются У.з. (их иногда наз. специфически учебными), ориентированные на усвоение обучаемыми общих (подходящих для многих ситуаций) способов действий. Примером специфически учебных задач являются задачи, которые требуют от школьников установления матем. и логических отношений между элементами прямоугольного треугольника. Знание этих отношений позволяет затем легко решать любые задачи на нахождение тех или иных элементов прямоугольного треугольника по к.-л. заданным элементам.

Г.А.Балл.

УЧЕБНАЯ ИГРА — игра, предполагающая достижение учебных целей.

Использование игр в учебных целях имеет давние традиции. Первые упоминания о применении игр для *обучения* грамоте и высказывания об обучающих функциях игр встречаются уже у древнегреческих философов Платона и Аристотеля. Французский философ Монтень отмечал “игры детей — вовсе не игры и правильнее смотреть на них как на самое значительное и глубокомысленное занятие этого возраста”. На важную роль игр в обучении и воспитании, использования игровых элементов (напр., загадок), указывали многие выдающиеся педагоги. Речь шла главным образом об использовании отдельных игровых элементов; первые попытки систематизации *обучающих игр* были предприняты немецким педагогом Ф.Фребелем. Популяризаторами широкого применения У.и. являлись известные итальянские педагоги М.Монтессори и Д.Колоцца. В 19 з. в России была сделана попытка организовать весь учебный процесс как комплекс познавательных игр (И.Б.Базедов). В русской педагогике проблеме детских игр уделяли внимание многие ученые прежде всего К.Д.Ушинский, А.И.Сикорский, В.П.Вахтеров. Важное значение игры в умственном развитии и повышении эффективности обучения отмечали выдающиеся педагоги Н.К.Крупская, П.П.Блонский, А.С.Макаренко и В.А.Сухомлинский.

Разработка теории детской игры, психологические механизмы игры, влияние ее на развитие человека и возможность целенаправленного использования в педагогической практике привлекали внимание известных ученых-психологов М.А.Басова, Л.О.Выготского, Д.Б. Эльконина и др. Осн. парадигматика их исследований связана с трактовкой игры как специфического вида человеческой деятельности, возникающей в результате общественных потребностей и предшествующей трудовой деятельности человека (по выражению Г.В.Плеханова — “игра — дитя труда”). Теория игровой деятельности строилась преимущественно на материале изучения игр дошкольников, и в конкретном историческом контексте создавалась на основе критики ряда теорий игры, разработанных буржуазными учеными (биологизаторской теории предупреждающей игры, теории функционального удовольствия; теории “избытка сил”; теории “стихийной игры”; теории глубинных влияний; теории неудовлетворенных возможностей; теории игрового универсума; теории динамики замещений). Теория детской игры наиболее разработана: игра в дошкольном возрасте рассматривается как ведущий тип деятельности, который содержит в себе все тенденции развития ребенка, является источником развития и создает зоны ближайшего развития. Конкретизация внимания исследователей на проблемах детской игры во многом повлияла и на линии анализа игры как культурного и социального феномена в целом. В теоретико-методологическом плане это проявляется в тенденции изучать игру как деятельность, хотя во “взрослых” играх она, как правило, “вплетена” в другие виды деятельности — учебную, трудовую и т.д. В строгом смысле слова лишь развлекательные игры (в частности, *приключенческие игры*) могут определить собственно игровую деятельность (т.е. ее цели, средства, продукты задаются в рамках самой игры, а не привносятся в нее, напр., из труда, учения и общения). В онтогенетическом плане детская игра рассматривается и как источник по развитию игры в более старшем возрасте, напр., обучающая игра рассматривается как результат эволюционирования ролевой игры, в которой роль является скрытой, а действие правил — открытым. В настоящее время игры “общения”, многообразные имитационно-моделирующие и деловые игры, в которые “играют” преимущественно студенты и управленческие работники, все интенсивнее проникают в учебный процесс средней школы, выступая в большинстве случаев в качестве учебных и обучающих: их осн. цель — изменения в самом играющем, т.е. новации в системе знаний, умений и навыков, интеллектуальное и психическое развитие, освоение культурных норм и пр. Различать их следует по критерию прямого (осознаваемого обучаемым в качестве цели деятельности) продукта: если эти новации выступают как прямой продукт деятельности, есть основания считать игру учебной. В традиционной же деловой игре (которая может включать имитацию и моделирование идеальных ситуаций) осн. цель связана

с принятием решений в имитируемой ситуации в условиях специально организованной коллективной деятельности ("метод мозгового штурма", синектика и т.д.).

Игровые методы приобрели большую популярность в зарубежной системе образования. Их брали на вооружение представители практической всех научных школ и направлений в области педагогики и психологии — от амер. педагогов-прагматиков Д.Дьюи и У.Килпатрик ("школа свободного воспитания") до швейц. психолога Ж.Пиаже, амер. психолога Д.Брунера и др. ученых, внедряющих ныне новые когнитивистские тенденции в развитии современной педагогической теории и практики за рубежом. Характерной для западных педагогов и психологов является переоценка роли игры в обучении и развитии, которая отражается в предложениях придавать игровую форму (прежде всего благодаря широкому использованию игр-драматизаций) всей учебной деятельности на всех уровнях системы образования. В отличие от такой крайней точки зрения большинство ученых полагают, что использование У.и. дает положительный эффект только тогда, когда их удельный вес в учебном процессе не превышает некоторых ограничений. На педагогическую теорию и практику США и других западных стран заметное влияние оказала концепция обучения амер. педагога и философа Д.Дьюи, в которой игровой драматизации отводилось центральное место. В основу концепции был положен принцип "учения через делание", игра рассматривалась как "инстинктивное и неизбежное проявление ребенка". Поэтому использование игр ведет к "естественному обучению", не противоречащему "свойственному природе ребенка инстинкту игры". Аргументы в пользу широкого применения игр и драматизаций в школе их приверженцы черпали как в повышении эффективности обучения (возрастание интереса к приобретению новых знаний, активизация совместного учения и взаимообучения школьников, создание благоприятного психологического климата в отношениях между преподавателем и обучаемыми, стимулирование инициативы и самостоятельности обучаемых в приобретении умений и навыков использования специальной и справочной литературы, акцент на произвольном запоминании), так и в более успешном протекании процесса социализации, лучшей подготовке обучаемых к жизни за пределами школы. Следовательно, речь шла об игре как средстве реализации прагматической концепции "воспитания для выживания", "воспитания приспособленности к жизни", которые получали наиболее развернутое воплощение в системе игр, моделирующих социальную среду и, в частности, "жизненную карьеру" (игры "Демократия", "Школа", "Король острова" и пр.). Эта концепция была разработана в 60-х гг. 20 в. кафедрой общественных отношений университета Дж.Гопкинса в США (в ней содержатся игры, моделирующие профессиональные, юридические, производственные и др. виды отношений). Кроме того, особое внимание к

играм в зарубежной школе объясняется акцентом на развитие эмоционального, аффективного в личности обучаемого, а также направленностью обучения на усвоение осн. понятий науки, отраженной в учебном предмете, с помощью “проблемных упражнений”, среди которых важное место занимает У.и., поскольку она, по выражению амер. ученого Г.Спрейга, позволяет свести понятия до такого уровня, что они “воспринимаются всеми внутренностями”. По мнению амер. исследователя Дж.С.Коулмена, в области разработки и применения учебных *имитационных игр* можно выделить три этапа. На первом доминировало “принятие на веру”: предложения носили форму пропаганды игр, оптимистических гипотез о том, что они дают и как могут применяться. На втором были получены экспериментальные результаты, показавшие, что игры в настоящей их форме имеют серьезные недостатки; существующие (преимущественно тестовые) способы оценки эффективности игр не являются адекватными и поэтому не могут способствовать их совершенствованию. Для этого этапа характерно мнение о том, что игры, повышая *мотивацию* обучаемых, не повышают степень усвоения знаний, и овладение с их помощью умениями решать проблемы или стимуляция критического мышления осуществляются не более эффективно, чем при использовании обычных методов обучения. По мнению ряда отечественных ученых, игровая форма сознания прежде всего способствует активации мыслительной деятельности, но не имеет особых преимуществ в качестве организационной формы обучения. Третий этап в развитии имитационных У.и., охватывающий и настоящее время, отличается, по словам Дж.С.Коулмена, реалистическим оптимизмом. Были, в частности, проведены многочисленные исследования, выявившие особенности различных игр, пути их изменения с целью повышения обучающего эффекта, с учетом контингента обучаемых и т.д., показавшие, что игры не только генерируют интерес, но также и обучают”.

У.и. — наиболее удобное из множества понятий, нередко употребляемых в аналогичном смысле, — обучающая игра, дидактическая игра и др., поскольку здесь подчеркивается включенность ее в учебную деятельность как особой организационной формы учения. Причем отличие У.и. от обучающей игры состоит в ее свободном протекании (типа play) — отсутствии явно выраженного управления по достижению учебных целей.

Возможны различные представления игры: как деятельности, как процесса и как особой формы учебной деятельности. Представление об игре как о деятельности наиболее распространено, однако в строгом смысле оно применимо только к играм дошкольников (поскольку здесь игровая деятельность является ведущей) и к “развлекательным” играм, когда обучение выступает как артефакт, а учебные приобретения — как побочный продукт деятельности. Цели, средства и продукты деятельности здесь непосредственно связаны с игрой;

результатом, к которому стремится играющий, является выигрыш. Цели и продукты У.и. определяются учебными целями и тем содержанием, которое должно быть усвоено в ходе игры, поэтому рассмотрение У.и. как вида игровой деятельности не продуктивно. Другое представление игры связано с ее анализом как процесса, “вплетающегося” в иную (в данном случае — в учебную) деятельность. Соответственно цели, средства и продукты игры должны быть рассмотрены в рамках учебной деятельности. Здесь на первый план выдвигается специфика соотношений между прямым и побочным продуктом деятельности. Достижение игровых целей (выигрыша, приза и пр.) оказывается значимым не само по себе, а лишь постольку, поскольку обеспечивается достижение учебных целей. Конституирующие процесс правила игры непосредственно не связаны с учебным содержанием, однако оказывают существенное влияние на эффективность учебной деятельности. Осн. требование, которое следует учитывать при разработке игры, — достижение игровых целей должно предполагать в качестве необходимого условия достижение учебных целей. Осн. задача в проектировании У.и. — обеспечить, при сохранении ее эмоциональной привлекательности, акцент на прямых продуктах деятельности, т.е. обеспечить осознание учащимися учебных целей как наиболее приоритетных в процессе У.и. Последнее из описанных представлений У.и. связано с анализом ее как особой формы учебной деятельности. Принципиальное значение здесь имеет возможность отличить У.и. от игры развлекательного плана и от не игры (игровых элементов), которые “вплетаются” в учебную деятельность, существенно не изменяя ее. Такие элементы (напр., использование “сказочных” персонажей, героев любимых книг и пр., различные мотивирующие приемы, связанные с мультипликацией и звуковыми эффектами, если игра реализована на компьютере, красочным фоном игры и т.д.) оказывают влияние на мотивационную и содержательные стороны учебной деятельности, поскольку не только повышают *мотивацию* к деятельности, но и могут привносить в нее новые объекты и отношения между ними. Развлекательные игры, “вплетающиеся” в учебную деятельность (напр., как поощрение за достигнутый учебный успех), могут повлиять на мотивационную сторону учебной деятельности, не затрагивая ее содержательную и операционную стороны. И только У.и. изменяет операционную сторону учебной деятельности и ведет к возникновению новых *учебно-игровых задач*. Это происходит потому, что в У.и. имеется по крайней мере одно правило, не лежащее в предметной области учебной задачи; оно не задано в знаковой системе, в которой решается задача (не заимствовано, в частности, из химии, физики и других предметов, на материале которых строится игра). Это “порождающее игру” правило позволяет отличать У.и. от игровых элементов и от не игры. При таком подходе У.и. может быть рассмотрена как особая форма учебной

деятельности, а процесс игры — как решение *учебно-игровых задач*. Указанные “порождающие игру” правила привносят в У.и. ту условность, которая, по словам психолога П.С. Выготского, является необходимым компонентом игры.

Особое внимание к теории и технологии применения игр в обучении вызвал феномен *компьютерных игр*, ставший ныне полноправным элементом культуры человека: отчуждение игр в виде компьютерной *программы* предъявляет особые требования к технологизации представлений о процессе и результатах игры. Вместе с тем, именно использование компьютера в качестве средства обучения позволяет осуществить в процессе игры *индивидуализацию обучения*, способствовать реализации заложенного в процессе игры дидактического потенциала, осуществить принцип наглядности, имитировать явления и процессы действительной и вымышленной реальности, а также организовать среду для свободного поиска решений представленной педагогом *учебной задачи*. Поэтому проблемы проектирования и применения компьютерных учебных и обучающих игр нужно рассматривать как важное и самостоятельное направление *компьютерной технологии обучения*.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

УЧЕБНАЯ ИНФОРМАТИКА — раздел *информатики*, изучающий вопросы создания и “встраивания” *компьютерной технологии обучения* (КТО) в процесс подготовки кадров на всех уровнях: школа—вуз—производство (наука). Функции компьютера в образовании весьма разнообразны — от управления органами народного образования до организации досуга *обучаемых* во внеурочное время. Что касается осн. функций компьютера, то он выступает и как объект изучения, и как средство *обучения*. Это предполагает усвоение *знаний, умений и навыков*, позволяющих успешно использовать компьютер при решении разнообразных *задач* или, иными словами, овладение *компьютерной грамотностью*. Кроме того, компьютер — мощное средство обучения, способное значительно повысить его эффективность. В зависимости от того, как используется компьютер, можно выделить два направления его применения в учебном процессе. Если компьютер используется только как средство *учебной деятельности*, то его функции почти не отличаются от тех, которые он выполняет в рамках других видов деятельности — научной, производственной. Возможности применения компьютера здесь весьма многообразны — от справочной системы до средства моделирования некоторых ситуаций. Это может обеспечить существенный образовательный эффект, однако нельзя сравнивать это со специфически педагогическим применением компьютера, поскольку не выполняется основная для системы обучения функция — управляющая. Выполнение функции обучения — наиболее существенная характеристика

применения компьютера в учебном процессе. Она определяет, какие *обучающие воздействия (задачи, вопросы, подсказки)* будут выданы обучаемому, и тем самым задает деятельность, которую тот должен осуществлять. Таким образом, компьютер моделирует обучающую задачу и реагирует на вопрос так, как это делает в аналогичной ситуации педагог. Четкую грань между “учебными” и “обучающими” функциями компьютера можно провести лишь поверхностно. Если *информационно-справочная система* выдает информацию, аналогичную той, которая содержится, напр., в матем. справочнике, то очевидно, что компьютер выступает здесь как средство учебной деятельности. Если же эта система выдает информацию с учетом конкретной задачи, которую решает обучаемый, причем информацию о том, какой прием решения следует применить, как лучше спланировать поиск *способа решения* или как проверить его правильность, то здесь компьютер выполняет и обучающие функции. Наиболее эффективно на базе компьютера осуществляется руководство самостоятельной деятельностью обучаемых, что дает возможность индивидуализировать обучение. В первую очередь — это контроль и закрепление приобретенных знаний, формирование и развитие практических навыков и умения их применять. Осн. формой организации *взаимодействия человека-машинного* является *диалог*, что обеспечивает и учебно-контролирующие функции, и возможности совместного кооперативного или игрового решения заданий, стоящих перед обучаемыми. С появлением компьютеров на учебных занятиях изменяется представление о реализации *принципа наглядности*. Прежде всего сам компьютер (экран его *дисплея*) выступает как наглядное средство, образно и динамично моделирующее на экране любой процесс и явление. Компьютер обладает уникальными возможностями моделирования; создаваемые и “проигрываемые” на нем матем. модели (формулы, таблицы, графики, диаграммы и т.д.) наглядны и открывают большие возможности для анализа и исследования изучаемых процессов и объектов. Педагогически обоснованные сферы применения моделирования развивают умственные способности обучаемых, улучшают и расширяют дидактические возможности понимания и усвоения учебного материала.

Являясь основой для создания КТО, У.и. наряду с этим преследует главную цель — повышение компьютерной грамотности обучаемых, работников различных отраслей, всего населения. Выделяют три группы вопросов, составляющих содержание компьютерной грамотности: 1) понятие об *алгоритме*, его свойствах, средствах и методах описания алгоритмов, *программе* как форме представления алгоритма для компьютера; основы *программирования* на одном из языков *программирования*; практические навыки обращения с компьютером; 2) принцип действия и устройство компьютера и его элементов; 3) применение и роль компьютеров в производстве и др. отраслях деятельности человека. Органической частью компьютерной грамот-

ности является умение общаться с компьютером. Для пользователя это в основном умение работать за дисплеем машины, т.е. овладеть клавиатурой дисплея, уметь подготовить машину к работе, вводить в память ЭВМ числа и переменные, корректировать введенные данные, вводить и отлаживать программу, подключать к компьютеру периферийные устройства, запускать и останавливать машину. Важным компонентом компьютерной грамотности является умение пользоваться имеющимися в программном обеспечении ЭВМ пакетами прикладных программ. С компьютеризацией обучения во всем мире связаны надежды повысить эффективность учебного процесса, обеспечить массовое индивидуальное обучение, уменьшить разрыв между требованиями, которые общество предъявляет подрастающему поколению, и тем, что в действительности дают школа и вуз.

В.Я.Валах, А.М.Довгялло.

УЧЕБНАЯ СРЕДА ЛИСП-ЛОГО — учебная среда, являющаяся расширением языка Лисп, позволяющим изучать программирование на этом языке на основе философии обучения Лого. Развитие языка Лисп на основе ценных элементов языка Лого содействует рассмотрению его возможностей. Включение в Лисп набора функций Лого упрощает изучение языка Лисп, сохраняя в то же время его возможности. С первых попыток работы с функциями Лого программист осваивает применение функций Лисп; следующий шаг включает работу со сложными функциями, которые очень просто вызываются. Эффект главных команд Лого заключается в их простоте и легкости изображения на графическом экране. В то же время “графика типа Лого” дает широкие возможности для исследования отдельных идей, в которых даже простые программы могут дать неожиданные результаты. Машинная графика представляет собой сильное и легко модифицируемое средство общения между человеком и компьютером. Графическое изображение информации, сложные структуры и отношения можно воспринимать в более короткий срок, в большем объеме и с меньшими ошибками. Поэтому машинная графика стала важным средством интерфейса человека с компьютером, приспособленным к специфическим требованиям людей. Расширение языка Лисп набором графических функций (табл.) представляет собой попытку более эффективного выявления его качеств в обычных приложениях либо при использовании методов искусственного интеллекта, когда графические обработки выполняют только положительную роль.

Представления команд Лого были внедрены в качестве функций в версии TC-LISP, являющихся эффективными для терминалов, располагающих графическими средствами, совместимыми с ТЕКТ-TRONIX 40XX (румынские терминалы типа DAF 220 или 52 и др.).

Набор графических функций, взятых из языка Лого для расширения языка Лисп.

Функция	Эффект
CLEARSCREEN (CB)	Удаляет графический экран
CLEARTEXT (CT)	Удаляет текстовый экран
FULLSCREEN (FS)	Удаляет целый экран для графики
TEXTSCREEN (TS)	Использует целый экран для текста
MIXEDSCREEN (MS)	Смешанное использование экрана (последние 5 строк для текста)
SETTEXT	Определяет размер текстовой зоны на смешанном экране
HIDETURT (HD)	Убирает курсор
SHOWTURTLE (SD)	Вызывает курсор
BACK (BK)	Смещает курсор назад
FORWARD (FD)	Смещает курсор вперед
RIGHT (RT)	Вращение курсора направо
LEFT (LT)	Вращение курсора налево
PENDOWN (PD)	Опускает карандаш
PENERASE (PE)	Опускает резинку
PENUP (PU)	Поднимает карандаш
SETL	Определяет тип линии для трассировки
SETPS	Определяет цвет для карандаша для трассировки
HOME	Возвращает курсор в исходную позицию
HEADING	Передаёт направление курсора
SETHEADING (SETH)	Определяет новое направление курсора
SETPOS	Перемещает курсор в специфицированную позицию X,Y
SETX	Перемещает курсор в специфицированную позицию X
SETY	Перемещает курсор в специфицированную позицию Y
XCOR	Передаёт курсору координату X
YCOR	Передаёт курсору координату Y
FENCE	Ограничивает перемещение курсора только в графической зоне экрана

Функция	Эффект
WRAP	Применяет тороидальную геометрию при перемещении курсора на экране
WINDOW	Позволяет курсору выводиться вне видимой на экране графической зоны

Перед применением поднабора функций машинной графики типа Лого необходимо определить тип экрана. Учитывая, что диалог между обучаемым и компьютером происходит постоянно, чаще всего избирается смешанный экран, образованный из графической и текстовой частей. Текстовая часть занимает последние 5 строк экрана, а графическая — остальную его часть. Для этого используется функция MIXEDSCREEN (MS). Размер текстовой зоны увеличивается или уменьшается с помощью функции SETTEXT *n*, где *n* — количество строк экрана, занятых текстовой зоной; на смешанном экране текстовая зона применяется для диалога, а графическая зона — для перемещения курсора (TURTLE); курсор изображается стрелкой, ориентация которой показывает направление перемещения. Первоначально курсор не изображен на экране. Для его вызова применяется функция SHOWTURTLE (ST). Сначала он помещается в позицию "HOME" (в центре экрана) и направлен вверх. Можно перемещать курсор вперед (функция FORWARD) или назад (функция BACK) на определенное число шагов, ориентировать направо (функция RIGHT) или налево (функция LEFT) на определенное число градусов, сравнительно с текущей позицией. Напр., функцией FORWARD 50 (FD 50) курсор перемещается вперед на 50 шагов, а функцией RIGHT 45 (RT 45) поворачивается на 45° направо. Можно переместить курсор вперед или назад, опустить вниз (PENDOWN) карандаш или поднять его (PENUP), опустить резинку (PENERASE). Если карандаш опускается, можно трассировать непрерывную линию или пунктир определенного цвета. Функциями SETLT и SETPS определяется тип линии и соответственно цвет. Тем самым можно трассировать непрерывные линии. Если резинка опущена вниз, то, следуя по предыдущей трассе, можно стирать ее. Для рисования определенных объектов на экране нужно определять функции Лисп. Следовательно, для построения квадрата, имеющего сторону, равную 50, исходя из текущей позиции курсора можно использовать функцию Лисп (PATRAT), с определением (DEFUN PATRAT ()

```
(DO I 1 (INC I) ( I 4)
(FORWARD 50)
(RIGHT 90)
)
)
```

Эта функция Лисп является эквивалентом следующей процедуры Лого:


```
TO PATRAT
REPEAT 4[FD 50 RT 90]
```

и при его вызове имеет тот же эффект. Если строится квадрат любого размера, нужно определить функцию PAIRAT, в которой сторона передается как аргумент в момент ее вызова. Определение функции имеет следующий вид:

```
(DEFUN PAIRAT (L)
(DO I 1 (INC I) ( I 4)
(FD L)
(RT 90)
)
)
```

Вызовом:

```
: (PATRAT 50)
```

на экране строится квадрат со стороной 50. Эквивалентом в Лого является *процедура*

```
TO PATRAT :L
REPEAT 4 [FD 50 RT 90]
```

Если к функции PATRAT прибавляется второй параметр А, представляя угол вращения направо после перемещения вперед на L шагов курсора, и работа функции возобновляется, тогда можно получить очень интересный эффект. Пусть эта функция POLI:

```
(DEFUN POLI (L A)
(FD L)(RT A)
(POLI L A)
)
```

Напр., вызовом:

```
: (POLI 50 120)
```

получаем равносторонний треугольник со стороной 50, а величинами:

```
: (POLI 80 144)
```

рисует звезду с пятью углами. Изменяя сторону L приращением L после каждого трассирования в форме:

```
(DEFUN SPI (L A I)
(F D L) (RT A)
(SPI (+ L I) A I)
```

можно обследовать множество геометрических форм типа спирали. Здесь возникли две рекурсивные функции: POLI и SPI. Рекурсивность является эффективным средством и ее можно использовать для получения сложных элементов.

Как явствует из приведенных примеров, легко начать программирование на Лисп, используя графику типа Лого. В то же время этот тип графики позволяет проводить исследования, в которых с помощью простых программ можно получить неожиданные результаты. Кроме того, постоянная взаимосвязь с системой играет важную роль в процессе изучения программирования.

М. Жумару.

УЧЕБНО-ИГРОВАЯ ЗАДАЧА — учебная задача, представленная в игровой форме. В обучении У.-и.з. выступают в качестве средств для повышения познавательной активности *обучаемых*, создания положительного отношения к обучаемому материалу. У.-и.з. можно получить путем введения во множество целей учебной задачи определенной игровой цели, которой соответствует результат, не входящий в предметную область учебной задачи (выполнить учебную задачу за ограниченное время, особым способом и т.п.). Если ввести условие, что за определенную сумму очков игрок получит некоторое поощрение (призовую игру, титул чемпиона и т.п.), то учебная цель будет достаточно завуалирована, чтобы обучаемый переставал акцентировать на ней внимание. Следовательно, изменяя соотношение учебных и игровых целей в У.-и.з., педагоги или обучающая программа способны влиять на эмоциональную сферу обучаемых для значительного повышения их познавательной активности. Можно предложить также следующий способ получения У.-и.з.: операции, которые обучаемый должен применить для решения учебной задачи, интерпретируются некоторым реальным или условным смыслом вне контекста предметной области этой задачи. Напр., нахождение решения системы двух линейных уравнений можно интерпретировать попаданием стрелы в движущуюся мишень. У.-и.з. можно рассматривать как учебную задачу в сочетании с различными игровыми элементами (игровыми персонажами, мультипликацией, соревнованием и пр.), направленными на создание положительного эмоционального фона, поддержку *мотивации* деятельности.

Э.Л.Ивахненко, Е.Д.Маргулис, А.Е.Стрижак.

УЧЕБНО-ИГРОВАЯ ПРОГРАММА — программа, предназначенная для реализации алгоритма некоторой учебной игры на компьютере. Программируемый калькулятор обеспечивает ввод последовательности команд в память ЭВМ для реализации простейших алгоритмов игр. Результаты каждого шага игры выводятся в числовой форме.

На базе компьютеров серии ЕС в начале их появления был создан ряд игр различного назначения, где компьютер выступал в качестве мощного вычислителя, позволявшего создавать программные модели для выработки оптимальных условий функционирования моделируемых систем. Такие игры наз. играми с помощью компьютера. К ним относятся деловые игры “Планы и стимулы”, “Директор”, теле-деловая игра “Кибернетический фитотрон”, созданная в Институте кибернетики АН Украины, и многие др. Сюда можно отнести также шахматные программы высокого уровня, для которых наиболее важна мощность компьютера, а не средства, поддерживающие взаимодействие с человеком. Те же компьютеры

серии ЕС с алфавитно-цифровыми терминалами позволили создавать диалоговые У.- и.п. с возможностью включать во взаимодействие с пользователем и псевдографические изображения. Эти программы обеспечили быструю (1—2 с) реакцию на воздействие человека и дали возможность опосредованно представить одного из игроков. С появлением графических цветных терминалов и персональных электронных вычислительных машин начали интенсивно создаваться игровые программы, широко используемые в сфере досуга, образования и в качестве компонента в профессиональной деятельности (для психологической разрядки, адаптации к новой технике и др.). Эти игры уже по праву стали называться компьютерными играми, т.к. появилась возможность представить на экране дисплея персонажи игры; одними (одним) из них управлял человек посредством периферийных устройств (клавиатура, манипулятор ручной), а другими (другим) — компьютер, причём только фантазия автора и искусство программиста накладывают ограничения на изображение игровых ситуаций и объектов.

Программа алгоритма игры состоит из двух частей: первая реализует её логическую (внутреннюю) структуру, вторая отражает процесс игры на экране терминала — интерпретирует знаковые модели объектов и связей в графические. Логическая структура подавляющего большинства компьютерных игр имеет три иерархических уровня: операционный, тактический и стратегический. Каждому уровню соответствуют свои операции (функции). Операционный уровень включает операции, выполняемые программой между двумя действиями играющего. Это передвижение на один шаг всех движущихся объектов, изменение игровых ресурсов (напр., времени игры), выяснение ("опрос"), воздействовал ли играющий на клавиатуру либо др. периферийные устройства. Тактический уровень с внешней стороны — это совокупность действий, приводящих к достижению локальной цели. На этом уровне происходит некоторое игровое событие, после него сюжет изменяется в том или ином направлении — в зависимости от того, достигнута ли играющим данная локальная цель. Тактический уровень включает операционный, но имеет также и свою собственную часть, куда входят операции: контроля деятельности играющего и оценки игрового события, выбора следующего события, ввод новых объектов и связей, задание параметров для операций оперативного уровня и т.п. На этом уровне реализуется, как правило, одна итерация игры. Переход к следующей, более сложной, сопровождается заданием новых параметров для операционного уровня, вводом новых объектов и связей и т.п. Стратегический уровень включает тактический, а также операции инициализации всех объектов и фона, задания параметров и критериев оценки для тактического уровня и окончания игры, фиксации результатов игры. Этот уровень представляет собой основную программу, реализующую сценарий игры. Именно эта часть отражает

творческий замысел разработчика и наиболее разнообразна в различных играх.

В У.-и.п. критерием оценки соответствуют *обучающим* *воздействиям*, т.к. именно с их помощью оцениваются результаты действий игрока и принимается решение о дальнейшем продвижении по сценарию игры. Каждому из рассмотренных уровней соответствует определенный характер деятельности в игре. Так, для успешной деятельности на операционном уровне необходима хорошая психофизиологическая реакция. Здесь развиваются концентрация внимания, быстрота реакции, психомоторная координация. Для тактического уровня требуются эмоциональная устойчивость, зрительная память, адекватное восприятие и осознание ситуации, умение быстро принимать решения. Стратегический уровень относится к интеллектуальной деятельности. Здесь необходимы воображение, логическое и комбинаторное мышление.

Э.Л.Ивахненко, А.Е.Стрижак.

УЧЕБНО-ТЕХНОЛОГИЧЕСКИЕ ЛИНИИ И МОДУЛИ (УТЛ и УТМ) — комплекс программных и технических средств, имеющих единую регламентированную технологию применения для организации и проведения учебного процесса. Различают УТЛ двух типов: инструментальные и прикладные. Инструментальная УТЛ (ИУТЛ) представляет собой комплекс программных средств, автоматизирующих все этапы разработки учебных средств. Она предназначена как для создания традиционных печатных учебных средств — учебников, задачников, чертежей и т.п., так и нетрадиционных — *программ учебного назначения (ПУН)*. ИУТЛ состоит из инструментальных учебно-технологических модулей (ИУТМ), каждый из которых поддерживает к.-л. этап разработки учебного средства, обеспечивает создание определенного типа прикладных программ либо нескольких типов программ. Прикладная УТЛ (ПУТЛ) — это комплекс традиционных учебных средств и/или прикладных ПУН, объединенных общей целью функционирования и тестирования к предварительной подготовке *обучаемых*. ПУТЛ состоит из прикладных УТМ (ПУТМ), представляющих собой *учебные программные модули и/или данные*, обеспечивающие решение *учебных задач* в определенной предметной области. Примером ПУТМ может служить *автоматизированный учебный курс*, реализованный с помощью специализированной системы программирования или авторской системы. Напр., в системе COSTOC курс, как правило, состоит из 10 модулей уроков, печатной документации и так наз. нулевого урока, включающего сведения о содержании курса, уровне и стиле изложения материала, а также предварительный обзор уроков и их фрагментов. Осн. требованием к реализации и оформлению УТМ является выполнение условий совместимости их между собой по

данным и/или управлению. Кроме требования программной совместимости, ПУТМ должна быть обеспечена совместимостью по методике обучения и содержанию учебного материала. Необходимым условием разработки ИУТМ является их концептуальное единство. Это единство означает, что программные средства, из которых состоит ИУТМ является их концептуальное единство. Это единство означает, что программные средства, из которых состоит ИУТЛ, должны представлять собой либо одну интегрированную систему, либо такой набор авторских средств, который обеспечивает общую технологию работы с ними и преемственность результатов их работы.

Преимущества описанного подхода к созданию прикладного и инструментального программного обеспечения *компьютерной технологии обучения*: возможность создания учебных программ или их компонентов экспертами в данной проблемной области, не являющимися специалистами по *программированию*; эффективное использование в зависимости от потребностей как отдельных ИУТМ, так и любого их сочетания; возможность компоновки конкретной УТЛ, задаваемой целью обучения, учебным планом и уровнем подготовки обучаемых, из набора ПУТМ; унифицированное хранение, преемственность и многократность использования *знаний* при создании учебных средств в данной проблемной области; возможность уточнения знаний при реализации и сопровождении ПУН; многообразие ПУН, возможность выбора наиболее подходящей по уровню подготовки обучаемого и его учебной цели; унифицированный *интерфейс* с различными ПУН; использование результатов тестирования и оценки качества при выборе наиболее подходящей программы учебного назначения ПУН.

Т.А.Лисюк, В.А.Петрушин.

УЧЕБНЫЕ СРЕДЫ — 1) Специально созданные или выбранные в естественных условиях обстановки, способствующие постановке и решению *учебных задач* (напр., комната со специально отобранными игрушками для развития познавательных способностей дошкольников, автодром — полигон для совершенствования навыков вождения автомобиля). 2) Класс *программ учебного назначения*, поддерживающий педагогическую функцию *учения*. Не содержат средств управления познавательной деятельностью *обучаемого* и предназначены для предоставления ему по крайней мере одной из следующих услуг: информационно-справочное обслуживание; инструментальное обслуживание; конструирование *имитационных моделей*; решение *задач*. Как правило, входят в состав программных средств *компьютерной учебной среды*.

Информационно-справочные системы (ИСС) учебного назначения представляют собой автоматизированные справочники по изучаемому предмету. В отличие от традиционных *баз данных* такие

ИСС содержат фактографическую информацию в текстовом и/или графическом виде. Взаимодействие с ИСС осуществляется на подмножестве естественного языка или на формальном языке запросов. "Интеллектуальность" ИСС определяется количеством различных типов запросов и степенью точности их "понимания".

Инструментальная система (ИС) предоставляет набор операций (инструментов), с помощью которых обучаемый может решать задачи. Наиболее выдающимися проектами по созданию ИС для обучения считаются проект *Лого* Массачусетского технологического института (МТИ) и система ШКОЛЬНИЦА, разработанная в вычислительном центре Сибирского отделения РАН. В последние годы в МТИ разрабатывается новый проект интегральной компьютерной среды для обучения. Эта ИС предназначена не только для обучаемых, но и для преподавателей и включает в интегрированной форме средства: обработка текстовой информации, включая структурирование файлов; использования, модификации написанных ранее и создания новых программ; работы с БД, в основном в терминах неформальных БД (журналов успеваемости, записных книжек и т.п.); широкого применения и программирования графических возможностей для поддержки всего вышеперечисленного. В основу этой ИС положен язык BOXER, развивающий Лого и представляющий набор стандартных типов "ящиков" (box—ящик) и операций над ними для визуализации действий. Используя средства языка, каждый пользователь системы может построить свой "мир". Другой путь создания ИС состоит в разработке небольших ИС с ограниченными возможностями, автоматизирующих некоторые части традиционных практических занятий. Такие ИС, активно разрабатываемые в Московском госуниверситете, получили название практикумов. Преимущества этих ИС: невысокие затраты на их реализацию и возможности функционирования на ПЭВМ с ограниченными ресурсами.

Основное из предъявляемых к ИС требований состоит в доступности ресурсов систем и наглядности её работы для обучаемого. При разработке ИС осн. принцип должен быть не только "что вижу, то имею", но и "что имею, то вижу", т.е. система не должна иметь скрытых ресурсов: все состояние системы должно быть на экране перед обучаемым. Важным направлением является разработка эффективных методов взаимодействия пользователя с ИС. Ныне активно исследуется методология объектно-ориентированного взаимодействия с компьютером. Сущность подхода состоит в том, что объект, с которым пользователь имеет дело в ИС, должен иметь некоторый "чувственный образ" на экране компьютера (пиктограмма, рисунок, текст). Работа пользователя с этим объектом состоит в том, что он выбирает объект (с помощью курсора, манипулятора ручного и т.п.), а затем преобразует его. Новое состояние образует "мир" пользователя, а его работа является деятельностью по преобразованию "мира" в некоторое требуемое состояние.

У.с., предназначена для *имитационного моделирования*, дает обучаемому средства представления некоторого процесса и позволяет собирать и обрабатывать характеристики процесса. Большое внимание при разработке систем данного типа уделяется визуализации изучаемого процесса. Примером имитационно-моделирующей У.с. может служить *STEAMER*.

У.с. предназначенная для решения задач, дает обучаемому возможность решать некоторый (довольно узкий) класс задач в заданной *предметной области*. Обучаемый формулирует задачу и поставляет учебные данные для её решения. Система выбирает или синтезирует *алгоритм* решения задачи и выдает результаты. При этом используются методы синтеза алгоритмов по заданной спецификации задачи или *экспертные системы*.

В.А.Петрушин.

УЧЁБНЫЕ СТРУКТУРЫ ЗНАНИЙ — это учебные и методические знания, представленные подлежащим образом для реализации в *обучающих системах*. Являются формой *представления знаний* о типовых ситуациях, возникающих при *проектировании* обучающих систем. Для представления У.с.з. целесообразно использовать *фреймы*. Примеры У.с.з.: *модель предметной области, модель обучаемого, методика обучения, стратегия обучения, тема, урок, задание.*

Модель предметной области можно представить в виде графа, каждому узлу которого соответствует изучаемое *понятие*, напр., объект, отношение, аксиома. Между узлами графа существует два типа связей, определяющих возможную последовательность изложения учебного материала и множество априорных понятий для изучаемого понятия. Обозначим все множество узлов графа через U . На множестве U выделяется подмножество главных узлов (УГ), в которых сосредоточен весь необходимый материал учебного курса. Специфика этих узлов состоит в том, что уровень изложения материала и уровень сложности *учебных задач* ориентируются на *обучаемого* с достаточно высокой квалификацией. Подмножество остальных узлов разбивается на непересекающиеся подмножества альтернативных узлов (УАi), объединение которых также содержит весь необходимый материал по курсу обучения, но уровень его изложения более низкий и рассчитан на менее подготовленного *польз. зателя*. На множестве U определено отображение $УГ \rightarrow \{УАi\}$. Последовательность переходов от урока к уроку, временное или постоянное изменение уровня *сложности задачи*, а также выбор материала для повторения определяются методикой обучения.

С каждым узлом графа, описывающим *предметную область* (ПО), связаны такие У.с.з., как задание, урок, тема. Самой элементарной У.с.з. является задание — вопрос или императив,

требующий действия обучаемого. Среди заданий выделяют учебные и контрольные. С помощью учебных заданий осуществляется изучение материала текущего урока. Они могут быть нескольких уровней сложности. Высший отражает сложность изучаемого материала или отношение порядка, если задания упорядочены. Контрольные задания делятся на априорные и апостериорные. Априорные определяют степень готовности обучаемого к восприятию текущего урока или темы, апостериорные — степень усвоения материала текущего урока.

Следующая У.с.з. — урок. Он включает несколько учебных заданий (обычно это три-четыре задания разных уровней сложности), апостериорные (одно — два задания одного или разных уровней сложности) и априорные контрольные задания (если для усвоения учебного материала текущего урока необходим материал не только предыдущего, но и ранее пройденных уроков). Выделяются главные и альтернативные уроки. Обучаемый начинает работу с главным уроком; если он справляется с предлагаемыми заданиями, то обучение продолжается в рамках главных уроков. При большом количестве ошибок дальнейшее обучение продолжается по ветке альтернативных уроков по крайней мере до окончания темы. Альтернативные уроки содержат более подробный и легкий для усвоения учебный материал.

Уроки объединяются в темы, включающие несколько главных и альтернативных уроков. В рамках каждой темы обучаемый может работать с главными или альтернативными уроками (в зависимости от степени усвоения предыдущих тем); при слабом усвоении учебного материала текущей темы его можно перевести с главной ветки на альтернативную.

Структуры фреймов для темы, урока, задания:

ТЕМА

<идентификатор темы>

<идентификаторы уроков по данной теме>

<текст — пояснение того, что будет знать обучаемый после изучения темы, комментарии>

<примеры использования изученных понятий>

УРОК

<идентификатор урока> <вводный текст>

<текст для режима априорного контроля>

<текст для режима обучения соответственно уровням сложности>

<определения>

<идентификатор темы — для априорного контроля>

<идентификатор урока — для априорного контроля>

<текст для режима контроля>

<заключительный текст> (включает реплики, пояснение дальнейших действий и т.п.)

ЗАДАНИЕ

<идентификатор задания> <текст задания>

<вспомогательный текст 1: релевантная информация>

<вспомогательный текст 2: указание пути решения>

<название темы> <название урока>

<степень сложности>

<признак использования> (задание уже выполнялось или нет)

<анализ ответа обучаемого> <пошаговое решение задачи>

<графическое сопровождение>

С проектированием деятельности *преподавателя* (системы обучения) связаны модель обучаемого, методика и стратегия обучения.

Модель обучаемого заполняется на основании выполнения правил, отражающих особенности поведения обучаемого. Существует достаточно много предложений по разработке моделей обучаемого. Четырехуровневая модель обучаемого содержит: локальный уровень, содержащий *информацию* о выполнении обучаемым последнего учебного задания; текущий уровень, содержащий результаты изучения текущего урока; глобальный уровень, содержащий результаты изучения всего курса обучения и последовательность прохождения узлов графа, отражающего ПО обучения; психологический уровень, отражающий результаты априорного тестирования. Локальный уровень используется для выбора очередного *обучающего воздействия* рамках одного урока; текущий и глобальный — для выбора очередного урока или темы для изучения. Основой для формирования этих трех уровней модели является анализ работы обучаемого. Уровень тестирования служит: для определения компетентности обучаемого при решении *задач* (одна из ветвей всесторонней экспертизы обучаемого — профессионализм); для установления тем, по которым у обучаемого могут быть затруднения, и для активизации соответствующей части *базы знаний* по ПО (экспертиза — диагностика); для определения типов задач, которые необходимо предоставить обучаемому для оптимизации процесса обучения, а также видов демонстраций и объяснений, которые необходимо использовать (предметная экспертиза).

Методика обучения определяет последовательность изложения учебного материала, критерии выбора очередной порции (соответствующего задания или урока) в зависимости от результатов обучения.

Стратегии обучения определяют способ изложения материала и правила выбора методики. Предлагается две стратегии обучения для реализации в обучающих системах: рассматривается осн. понятие курса, а затем предлагается решить ряд задач, с его использованием; обучаемому предлагается решить несложную задачу, в процессе решения которой система определяет, какие понятия изучаемой темы обучаемому неизвестны или непонятны, и предлагает

учебный материал в ходе решения задачи по мере необходимости. В рамках каждой стратегии могут существовать и альтернативные.

Структуры фреймов для модели обучаемого, методики обучения и стратегии обучения:

МОДЕЛЬ ОБУЧАЕМОГО

- <оценка изучения задания> <оценка изучения урока>
- <оценка изучения темы и последовательности прохождения узлов графа>
- <оценка мастерства>
- <оценка индивидуальных особенностей обучаемого>
- <стиль изложения материала> <дизайн>
- <форма предъявления материала>
- <названия тем, по которым будут затруднения>

МЕТОДИКА

- <правила выбора порции учебного материала>
- <правила корректировки модели обучаемого> (учет ошибок, сделанных обучаемым, и определение пригодности методики)

СТРАТЕГИЯ

- <способ изложения материала>
- <правила выбора методики>

Над У.с.з. можно производить различные операции. Приведем список операций, которые были положены в основу реализации процесса обучения с помощью У.с.з. (слева даны типы входных параметров, справа — выходных).

1. Реализация процесса обучения на основе модели обучаемого, методики и стратегии обучения:

НАУЧИТЬ: → модель обучаемого.

2. Построение модели обучаемого на основе результатов априорного тестирования:

ПОСТРОИТЬ_МОДЕЛЬ_ОБУЧАЕМОГО: → модель обучаемого

3. Корректировка модели обучаемого на основе *анализа ответов обучаемого* по заданиям, урокам, темам:

КОРРЕКТИРОВАТЬ_МОДЕЛЬ_ОБУЧАЕМОГО:

модель обучаемого, методика → модель обучаемого.

4. Выбор стратегии:

ВЫБРАТЬ_СТРАТЕГИЮ:

модель обучаемого → стратегия, модель обучаемого.

5. Переход от одной стратегии к другой:

ИЗМЕНИТЬ_СТРАТЕГИЮ: модель обучаемого, стратегия → модель обучаемого, стратегия.

6. Выбор методики на основе модели обучения и выбранной стратегии:

ВЫБРАТЬ_МЕТОДИКУ: модель обучаемого, стратегия → модель обучаемого, методика.

7. Изменение методики в зависимости от *ответов* обучаемого на учебные задания:

ИЗМЕНИТЬ_МЕТОДИКУ: модель обучаемого, методика → модель обучаемого, методика.

8. Выбор очередной темы согласно таблице решений:

ВЫБРАТЬ_ТЕМУ: модель обучаемого, модель предметной области, методика → модель обучаемого, тема.

9. Выбор главного или альтернативного урока согласно таблице решений.

ВЫБРАТЬ_УРОК: модель обучаемого, модель предметной области, методика → модель обучаемого, урок.

10. Выбор учебных и контрольных заданий различных уровней сложности согласно таблице решений:

ВЫБРАТЬ_ЗАДАНИЕ: модель обучаемого, модель предметной области, методика → модель обучаемого, задание.

В.В.Колос, С.П.Кудрявцева.

УЧЁБНЫЙ ДИАЛОГ — *диалог*, реализующий процесс *обучения* и направленный на достижение его ближайших и отдаленных целей. Обычно характеризуется не наличием двух или нескольких партнеров общения, а двух и более смысловых позиций относительно темы, предмета диалога. При таком рассмотрении диалог между человеком и компьютером является по сути техническим опосредствованным общением между двумя людьми (*пользователем* и *разработчиком системы*) и характеризуется замещением разработчика техническим посредником (компьютером) с передачей последнему определенных полномочий в *принятии решений* (т.е. возможности автономного поведения). Эта форма общения естественного межчеловеческого общения, что отчасти компенсируется предоставляемыми компьютерными средствами реализации мыслительной деятельности пользователя.

Диалог при *компьютерном обучении* должен обеспечивать психологическую комфортность общения при сохранении его педагогической направленности; для этого необходимы высокая мера *адаптации* (индивидуализация), симметричность общения (равенство инициативы коммуникантов), равная структура партнерства (паритетность начал общения), высокая личностная отнесенность, полимодальность (свойственная естественному общению множественность форм и способов представления *информации*), которая в компьютерной системе может реализовываться посредством сочетания *текста*, графики, чертежей, анимации, цвета, звука, музыкального сопровождения, а также различных видов вводимых с *клавиатуры* ответов *обучаемого* конструируемых, выборочных и пр.

Осн. средства общения в У.д. — языки общения, языки *программирования* (ЯП) высокого уровня (см. *Уровень алгоритмиче-*

ского языка) и человеко-машинный интерфейс. К языкам общения относятся *естественные языки ограниченные* и специализированные формальные языки, напр. командные, представляющие собой командные запросные системы *программного обеспечения*, ориентированные на пользователя. Двухсторонняя природа человеко-машинной коммуникации требует дифференцированного подхода к коммуникативным средствам в зависимости от роли человека как отправителя или получателя сообщения. Для этого разрабатываются языки общения человека с машиной (ЯЧМ), отвечающие принципам психологической естественности. Человеко-машинный интерфейс представляет собой комплекс средств, позволяющих реализовать диалоговое взаимодействие между человеком и компьютером без использования ЯП. Осн. требования к интерфейсу: простота изучения и использования; дружелюбность; интеллектуальность; применение в качестве каналов информационного обмена естественных средств общения естественного языка, звучащей речи, зрительных образов. В обучающих системах структура интерфейса усложняется за счет введения дополнительного блока, решающего задачи обучения.

Осн. особенности У.д. и налагаемые на него ограничения связаны с характером опосредствования. Прежде всего это языковые ограничения: лимитированность словаря, жесткость синтаксиса, стереотипность речи, регламентированность формы сообщения, редукция образно-ассоциативных компонентов, смысловых оттенков и подтекста. Диалог в обучающей системе представляет собой письменный диалог в реальном масштабе времени и соответственно обладает как чертами письменной речи (монологичной по своему характеру и отличающейся сознательным отношением к средствам выражения и регламентированностью формы сообщения), так и чертами естественного диалога (со спонтанностью реакции, передачей инициативы между коммуникантами, ситуативностью и эллиптичностью живой речи и т.д.). Противоречие между естественным диалогом и рамками письменной речи повышает напряженность общения. Гипертрофия роли языковой структуры сообщения, обусловленная неприменимостью внелингвистических средств передачи его семантического наполнения (мимики, жеста, интонации), приводит к перегрузке зрительного анализатора человека информацией в буквенно-цифровом коде и ограничению модальности общения (т.е. качественных характеристик и форм представления передаваемой в процессе общения информации в буквенно-цифровом коде и ограничению модальности общения (т.е. качественных характеристик и форм представления информации), что также вызывает дискомфорт. Для устранения необходимо шире применять графику, цвет, звук. Эффективно использование в информационном обмене динамических моделей окружающей среды в виде пространственно-временных паттернов, что позволяет сохранить полимодальность общения, необходимую для оптимального восприятия. Соотношение используемых модальностей

зависит от возраста обучаемых, предметного содержания и целей обучения.

Временные ограничения со стороны компьютера как работающего устройства оказывают определенное психологическое давление на пользователя. Для снятия этой напряженности необходимо обеспечить обучаемому возможность произвольного входа в диалог и выхода из него в любой точке взаимодействия. Серьезной проблемой является ограниченность возможностей компьютера в понимании сообщений пользователя, поскольку процесс понимания текста требует актуализации дополнительных знаний. Громоздкий вспомогательный диалог, направленный на уточнение смысла общения, деформирует процесс общения и вызывает негативные эмоции у пользователя.

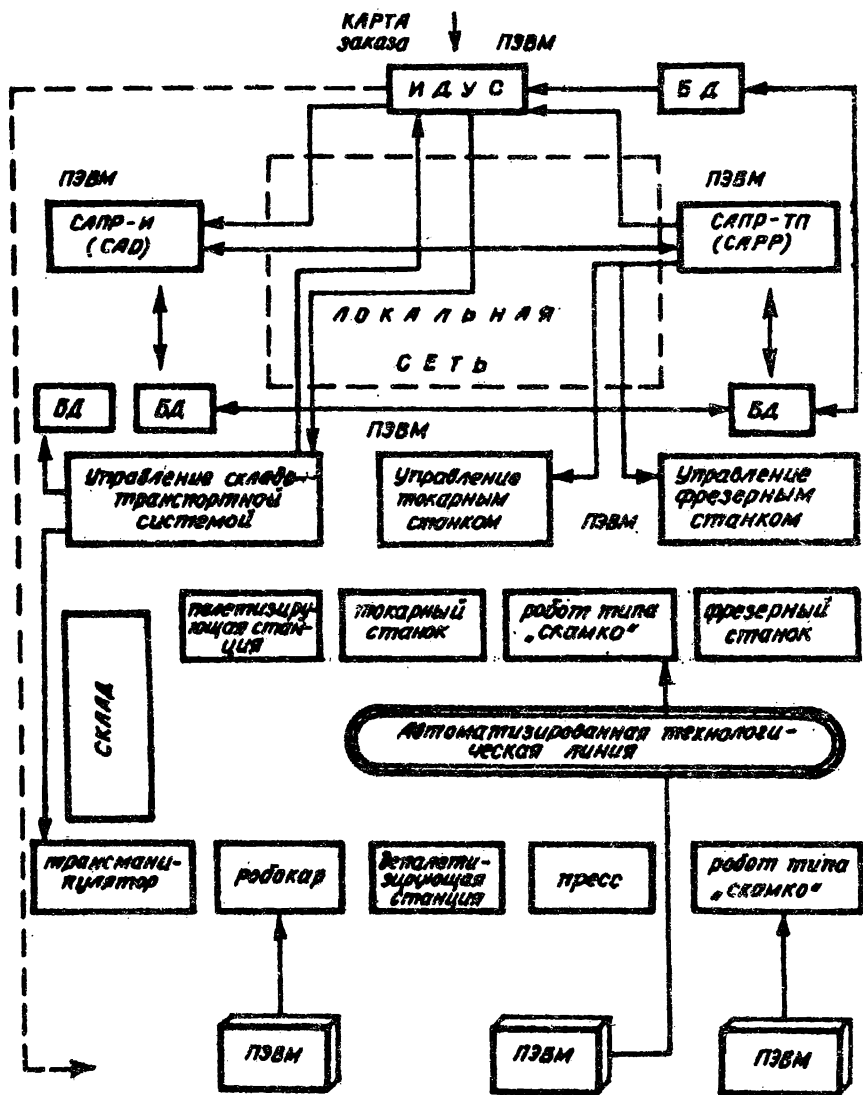
Эффективность У.д. зависит также от корректной организации обратной связи в обучении, правильного выбора времени, формы и меры помощи, адекватного оценивания деятельности обучаемого. Нарушение этих правил приводит к разрешению общения, не позволяя достичь цели обучения.

Теор. анализ У.д. предполагает как разработку формальной структуры учебного диалога (логико-лингвистические аспекты, организация процесса общения), так и неформальной, собственно психологической структуры, реализующей общепсихологические принципы построения диалога обучаемого с компьютером. Выделяют такие психолого-педагогические проблемы построения диалога в системе обучаемый — компьютер: общепсихологические принципы построения диалога обучаемый — компьютер; организация процесса общения; содержательные аспекты общения, лингвистические аспекты (выбор языка общения, построение текста сообщения, его форма, размер); модальность сообщения (тип предъявления информации и ответов обучаемых). Имеющиеся данные позволяют утверждать, что использование компьютеров ведет к преобразованию всех этапов решения задач — ориентировки, исполнения, контроля, а также способствует актуализации потребности в общении.

В.В.Андреевская, Е.Ю.Комиссарова, Е.Д.Маргулис.

УЧЕБНЫЙ КАБИНЕТ ГИБКОГО АВТОМАТИЗИРОВАННОГО ПРОИЗВОДСТВА, мини-ГАП — учебный кабинет, включающий систему обучения и комплекс технических устройств, соответствующих устройствам современной автоматизированной технологической линии машиностроения, выполненных в малогабаритных размерах, но сохраняющих все функциональные возможности этих устройств.

Комплекс технических устройств мини-ГАП (рис.) включает: автоматизированный склад; робокар; транспортную систему; палетизирующую станцию; депалетизирующую станцию; робот-манипулятор; токарный станок; фрезерный станок. Комплекс занима-



—————> Прямая передача данных (по каналам связи)
 - - - - -> Косвенная передача данных (через ГМД)

ет школьный учебный кабинет и включает все реальные производящие робототехнические устройства, необходимые для выпуска изделий машиностроения небольших размеров и сложности.

Осн. компонентами программно-информационного обеспечения мини-ГАП являются система управления мини-ГАП, подсистемы управления технологическими участками и подсистема обучения. В систему управления мини-ГАП входят все компоненты промышленного ГАП. Система обеспечивает функционирование технических устройств мини-ГАП в процессе производства машиностроительных изделий и состоит из следующих осн. подсистем: информационной диспетчерской управляющей системы (ИДУС), включающей подсистему оперативного планирования и управления; системы автоматизированного проектирования машиностроительных изделий (САПР-И); системы автоматизированного проектирования технологических процессов (САПР-ТП); системы управления складско-транспортным хозяйством (СКЛАД); подсистемы управления технологическими участками: токарным станком, фрезерным станком, роботом-манипулятором, робокаром; локальной сети.

ИДУС предназначена для автоматизации процессов управления всем комплексом технических устройств мини-ГАП с учетом оптимальных режимов его работы. Для этого ИДУС осуществляет: сбор и обработку информации о ходе управляемого процесса; оперативное планирование (уточнение) работы ГАП в оптимальном режиме; контроль выполнения оперативных планов; получение текущей информации о ходе процесса и выполнения оперативных приказов; получение данных о состоянии ГАП; фиксация контролируемых точек ГАП; динамическое отображение на мониторе протекающих процессов на ГАП в реальном времени; формирование заказов на изготовление деталей; связь с другими подсистемами; формирование количественных характеристик по готовой продукции. ИДУС может работать в двух режимах — обучающем (система меню с графическим интерфейсом и возможностью ручного вмешательства) и автоматическом (все управление на себя берет ПЭВМ, т.е. контроль и управление полностью осуществляет ЭВМ). В этом случае диспетчер функционирует как самоприспосабливающаяся система.

САПР-И (САД) предназначена для автоматизации проектирования машиностроительных изделий и подготовки конструкторской документации на изделие. Содержит базу данных типовых технологических поверхностей и деталей аналогов для мини-ГАП. Для этого САПР-И осуществляет все функции проектирования заказанного ИДУС изделия и передает по локальной сети САПР-ТП чертеж и спецификацию изделия. САПР-И может работать в двух режимах: обучающем (система меню с графическим интерфейсом, обучающими воздействиями и информацией) и проектировочном (обеспечивается процесс автоматизированного проектирования деталей).

САПР-ТП предназначена для автоматизации оптимального планирования работы мини-ГАП методами сетевого планирования, управления и разработки технологических карт и программ для управления фрезерным и токарным станком. С этой целью САПР-ТП осуществляет: ведение БД, описывающей технологические процессы мини-ГАП (корректировку, добавление, удаление, ввод новых сетевых моделей); управление (анализ результатов планирования, формирование и передача диспетчеру плана работы мини-ГАП); связь с подсистемами по сети (прием и передача сообщений по электронной почте); планирование (выбор альтернативной сети модели); формирование сетевой модели выполнения операций на ГАП; взаимодействие с системой средствами оконной графики. Имеются средства связи технолога с диспетчером и складом. Используются режимы диалога (для обучения и корректировки информации) и отображения статистической информации.

СКЛАД предназначен для учета накопления и расходования запасов материалов, а также управления этими процессами. Решаемые проблемы: определение наличия или отсутствия запасов, не приводящих к непроизводительным потерям; ограничение чрезмерного накопления запасов, поскольку это связано с омертвлением ресурсов и переполнением складских помещений. Задачи: определение наиболее рационального количества запасов, самой выгодной стратегии их пополнения и расходования. Система управления складом осуществляет: удовлетворение спроса на материалы; заказ на пополнение материалов; определение количественных характеристик по материалам на определенные моменты времени; определение единиц готовой продукции, исходя из имеющихся запасов; управление спросом и учет фактического наличия материалов; прием и удовлетворение заявок; формирование и передачу маршрутной карты. Работает в диалоговом (отображение статистической информации в виде форм и накладных) и учебном (имитация работы робота манипулятора на складе для решения учебных задач) режимах.

Система обучения основывается на интеграции различного рода компьютерных средств учебного назначения (от текстовых редакторов и графических редакторов до интеллектуальных обучающих систем) в единый информационно-методический комплекс, поддерживаемый развитыми инструментальными и авторскими системами. Разработка системы обучения при таком подходе позволит генерировать учебно-технологические линии (см. Учебно-технологические модули и линии), эффективно реализующие цели обучения для различных контингентов обучаемых (от младших школьников до слушателей системы повышения квалификации и переподготовки). Эта система отражает архитектуру системы управления мини-ГАП и её информационную базу. Основное назначение — управление учебной деятельностью обучаемых для постепенной подготовки их к умению работать с компонентами промышленных ГАП. Система

обучения включает: набор учебных программ для различных контингентов; пакет учебных программ по предметной области (ПО) ГАП; пакет автоматизированных практикумов по ПО ГАП; набор УТМ по компонентам ГАП; деловую игру по выпуску продукции. Содержание обучения по ПО ГАП включает ПО практически по всем разделам *инфоматики* и достаточно сложно для усвоения не только для школьников и студентов, но и для специалистов с высшим образованием. Проектирование учебной деятельности обучаемых реализуется на основе применения *тезаурусно-сетевой модели* ПО. Реализация этой модели осуществляется на базе специальной словарной системы. УТЛ системы обучения могут применяться для обучения основам *информационной культуры* начиная с младшего школьного возраста. Сюда входят диалоговые программные модули для реализации учебно-игрового взаимодействия детей с графическими моделями робототехнических устройств и решения задач по управлению этими устройствами в эмоционально привлекательной обстановке, что значительно повышает их познавательную активность. Такие модули могут функционировать на простейших учебных персональных компьютерах, что позволит широко применять их в начальных и средних учебных заведениях, не имеющих возможности приобрести комплекс технических устройств мини-ГАП. Для проектирования учебной среды необходимо выделить учебные цели. По начальному уровню обучаемого и структуре ПО системы управления ГАП учебные цели составляют три уровня: 1) управление ГАП в диалоговом режиме; 2) управление посредством функционально-ориентированного языка системы управления (язык описания протокола управления технологическим участком); 3) управление посредством микропрограммирования. Завершающим этапом обучения является деловая игра по выпуску продукции машиностроения на автоматизированной технологической линии в мини-ГАП.

К.Атанасов, Э.Л.Ивахненко, А.Е.Стрижак, С.Ушкалов.

УЧЕБНЫЙ МАТЕРИАЛ НА МАШИННЫХ НОСИТЕЛЯХ — информационные и контролирующие материалы, учитывающие содержательную сторону обучения и дидактические критерии, на основе которых формируется деятельность обучаемых. Информационный материал раскрывает содержательную сторону обучения и включает определение понятий, изложение фактов, раскрытие закономерностей протекания процессов и явлений, характерных для изучаемого предмета. Контролирующий материал предназначен для оценки деятельности обучаемых и представляет собой контрольные вопросы, задания, упражнения, которые они должны выполнить для закрепления информационного материала. По результатам работы с контролирующим материалом осуществляется функция управления обучением, определяющая деятельность по усвоению учебного ма-

териала. Производительность труда автора на этапе подготовки учебного материала существенно повышается при использовании инструментальных средств КТО, таких, напр., как *текстовые редакторы, графические редакторы, настольные издательские системы*, позволяющие оперативно тиражировать и, при необходимости, вносить изменения в содержание и структуру учебного материала. У.м. на м.н., подготовленный таким образом, можно представить на экране дисплея в виде печатных инструкций, учебных пособий, методических указаний, документации и др. материалов, используемых, напр., для организации *самостоятельной работы студентов*. Важно подчеркнуть, что в этом случае компьютер используется только как средство подготовки и визуализации учебного материала. Если обучение осуществляется с использованием *автоматизированных учебных курсов (АУК), тренирующих диалоговых систем, диалоговых игр* и т.п., то информационный и контролирующий учебный материалы могут занимать различное положение относительно обучающей программы, управляющей последовательностью предъявления учебного материала, что определяет три осн. способа представления учебного материала. 1. Учебный материал встроен в обучающую программу. Такая форма представления учебного материала (встроенная документация) используется для программ, рассчитанных на небольшое контактное время, характеризующихся малым объемом информационного материала и предназначенных для изучения отдельных *понятий* и частных разделов дисциплин. Примером такого представления учебного материала может быть встроенная документация учебно-программных средств на машинных *носителях информации*. 2. Учебный материал выделен из обучающей программы и подготовлен в виде печатного документа. В этом случае в *программе* содержатся ссылки на печатный документ в виде: ИЗУЧИТЕ РАЗДЕЛ <номер> ПОСОБИЯ <названия> или ВЫПОЛНИТЕ ЗАДАНИЕ <номер задания>. Эта форма представления учебного материала используется при необходимости иметь универсальную программу. Программа, реализующая принятый *алгоритм управления учебной деятельностью*, не привязана к конкретной дисциплине, а её перестройка на др. дисциплину заключается в замене печатных пособий и ссылок на их разделы. *Ответы* обучаемых оцениваются по принципу ПРАВИЛЬНО—НЕПРАВИЛЬНО без к.-л. разъяснений допущенных ошибок. Реализация таких программ достаточно проста, однако они получают однообразными, не вызывают интереса у обучаемых и, как следствие этого, обладают невысокой дидактической эффективностью. В то же время нельзя не отметить одно существенное преимущество этих программ, связанное с наличием у обучаемых печатных пособий, с которыми можно работать вне *обучающей системы*, а работу с компьютером применять для выяснения возникших вопросов и закрепления самостоятельно изученного материала. Эта форма представления

учебного материала не зависит от изобразительных возможностей дисплея компьютера и, следовательно, применима для изучения любых дисциплин. Пример: документация автоматизированных учебных курсов. 3. Учебный материал выделен из обучающей программы и подготовлен на машинном носителе в отдельном наборе данных. Этот способ сочетает в себе возможности первых двух, т.к. в программе содержатся специальные команды обращения к набору данных для вывода на экран обучаемого необходимых записей учебного материала и сохраняется возможность получения из этого же набора данных печатных копий учебного материала. Обучающая программа значительно усложняется и создается так, что реагирует на специальные команды обучаемого с помощью которых им осуществляется выбор способа представления учебного материала — либо воспринимать его, читая с экрана, либо воспринимать, читая печатный текст учебного пособия. Разработчики обучающих программ видят в каждом из способов представления учебного материала свои недостатки. В первом случае это значительное увеличение объема хранимой программы, сложность внесения изменений в текст программы (для этого необходимо перепрограммировать, а иногда и перетранслировать изменяемый модуль учебного курса) и, наконец, планируемое содержание учебного материала полностью зависит от изобразительных возможностей терминального оборудования (цвет, графика и т.д.). Во втором случае возникают проблемы, связанные тиражированием печатных документов. Их подготовка на машинном носителе может производиться с помощью текстовых и графических редакторов, однако распечатка в достаточном количестве экземпляров с машинного носителя на бумагу весьма проблематична. Для такого способа представления учебного материала обучающая программа получается универсальной, но, вместе с тем, её обучающие возможности значительно снижаются: отсутствуют разъяснения допущенных ошибок, всякого рода помощь и т.д. При хранении учебного материала на машинном носителе в отдельном наборе данных проблемы тиражирования учебного материала смягчаются, т.к. его всегда можно получить на дисплее компьютера, но вопросы, связанные с изобразительными возможностями дисплея, стоят по-прежнему остро. Компенсировать отдельные недостатки позволяет сочетание различных форм представления информационных и контролирующих учебных материалов, что позволяет достичь высокой дидактической эффективности обучающих программ.

Для программ достаточно большого объема получил широкое распространение такой способ представления учебного материала, когда информационный материал выделен из программы и подготовлен либо в виде печатного пособия, либо на машинном носителе в отдельном наборе данных, а контролирующий материал полностью погружен в программу. Обычно в этом случае программа выводит на экран обучаемого пред-текст, содержащий краткое введение в

изучаемое понятие или раздел предметного содержания. После этого следует отсылка к печатному пособию для детального изучения материала, предъявление контрольных заданий и анализ ответа обучаемого с подробными разъяснениями допущенных ошибок и, после отработки контрольных заданий, выдается пост-текст, подводящий итог по данному разделу или понятию. Преимущество такого способа включения учебного материала в обучающую программу заключается в том, что пособие можно использовать для самостоятельной работы, а программа является дополнением к печатному пособию, значительно повышающим его дидактическую эффективность. Диалог с компьютером, характерный для этого способа представления информационного учебного материала выглядит так:

К: Рассмотрим действительные константы. Действительные константы в языке ФОРТРАН имеют две формы представления:

1. Основная форма (F-форма);
2. Форма с порядком (E-форма).

Изучите раздел 4.2, страница 10.

О: Помоги.

К: Обращаю Ваше внимание на то, что:

- при записи действительных констант вместо десятичной запятой употребляется точка;
- точность представления действительных констант до 7-ми значных цифр;
- допустимая область значений для действительных констант приблизительно от $5.4 \cdot 10^{**}(-79)$ до $7.2 \cdot 10^{**}(75)$.

Изучите раздел 4.2, страница 10.

О: Готов.

К: Проверим, как вы усвоили учебный материал. Представьте число 197 в виде действительной константы в F-форме.

О: Помоги.

К: Еще раз напоминаю, что действительная константа в F-форме обязательно должна содержать точку, которая определяет целую и дробную части. Для целого числа дробная часть — это нуль. Представьте число 197 в виде действительной константы в F-форме.

О: 197.0

К: Правильно. Определите, является ли запись — .8 действительной константой?

О: Не является.

К: Ответ неправильный. В записи действительной константы десятичная точка может стоять в начале числа и в этом случае целая часть равна нулю. Вы плохо справились с предлагаемыми заданиями. Еще раз внимательно изучите раздел 4.2, страница 10 или обратитесь к преподавателю за консультацией.

Если информационный учебный материал подготовлен на машинном носителе в отдельном наборе данных, то вместо адресации (ссылка на раздел и страницу) этот материал специальной частью обучающей программы (процессором вывода) предъявляется на экран. Для подготовки информационного учебного материала в отдельном наборе данных автор АУК может использовать систему управления базами данных либо специализированное инструментальное средство (процессор ввода), на вход которого поступает описанное с помощью операторов формализованного языка учебное пособие (табл.). Подготовленное так пособие помещается в разделе библиотеки учебного материала курса. Для произвольного доступа к библиотеке учебного материала курса в распоряжение обучаемого предоставляются команды. Выдав команду БИБЛИОТЕКА, обучаемый получает на экран список учебных материалов, имеющих в библиотеке курса. После указания требуемого пособия на экран выводится его содержание, из которого можно выбрать нужный материал, указав его номер, соответствующий рубрикации содержания. Применение такого способа представления учебного материала повышает дидактическую эффективность обучающей программы за счет предоставления обучаемому сервисных средств. Примером реализации такого курса может служить многофункциональный АУК АИДА.

Список операторов формализованного языка для подготовки учебного пособия

№	Оператор	Синтаксис	Функция
1	Пособия	ПОСОБИЕ (имя)	Задаёт имя пособия, должен быть первым оператором в последовательности текста пособия
2	Содержания	СОДЕРЖАНИЕ	Указывает, что следующие за этим оператором строки описывают содержание пособия
3	Раздела	РАЗДЕЛ(имя)	Указывает имя раздела, служащее ключом поиска. За оператором следуют строки текста, выводимые на экран при запросе раздела
4	Главы	ГЛАВА(имя)	Указывает имя главы, служащее ключом поиска. За оператором следуют строки, выводимые на экран при запросе главы
5	Пункта	ПУНКТ(имя)	Указывает имя пункта, служащее ключом поиска. За оператором следуют строки, выводимые на экран при запросе пункта

УЧЁБНЫЙ ПРОГРАММНЫЙ МОДУЛЬ — базовый функциональный компонент *автоматизированного учебного курса*, ориентированный на решение некоторой педагогической задачи. Представляет собой сочетание *программы учебного назначения* (ПУН) и учебно-методических материалов, определяющих педагогическую задачу У.п.м., а также регламентирующих и поясняющих применение ПУН при решении педагогической задачи. Как правило, У.п.м. ориентирован на решение достаточно конкретной педагогической задачи, которой в практике учебного процесса обычно соответствует урок (или занятие) по определённой теме. У.п.м. может быть моделирующим (входящая в его состав ПУН осуществляет моделирование поведения реальных или абстрактных объектов); информационно-справочным (ПУН предоставляет необходимую для данного урока текстовую или графическую *информацию*); контролирующим (отработка навыков и умений, получаемых на данном уроке) и др. См. также *Учебно-технологические линии и модули*.

В.И.Отенко.

УЧЕНИЕ — процесс, направленный на приобретение *знаний, умений и навыков, способов действий* по решению различных классов задач. К этому процессу применимы и общие закономерности научного познания, хотя и в несколько специфической форме (не ставится задача открытия новых научных истин, а требуется лишь их творческое усвоение). Многие виды знаний приобретаются *обучаемыми* не посредством созерцания изучаемых объектов, а через рассказ *преподавателя*, описания, инструкции и т.д. У. продолжается в течение всей жизни человека, свойства его (состав и структура, факторы и детерминанты) являются предметом *дидактики* и педагогической психологии. У. имеет первостепенное значение для решения задач *обучения* подрастающего поколения. Необходимо различать понятия “учение” и “учебная деятельность”. При *учебной деятельности* именно учебные приобретения являются *прямыми продуктами деятельности*, в то время как при У. эти приобретения могут выступать как побочный продукт к.-л. иной (не только не учебной, но даже и не познавательной) деятельности. При рассмотрении У. принципиальное значение имеет теория, объясняющая его сущность, механизмы и закономерности. В педагогической психологии существует целый ряд теор. концепций, содержащих различные представления об У. и опирающихся на них.

Говоря о зарубежных теориях У., подразумевают прежде всего концепции, выработанные в бихевиоризме, гештальтпсихологии, школе Ж. Пиаже, информационном подходе к когнитивной (познавательной) психологии. В рамках бихевиоризма У. рассматривалось

как изменение внешних реакций при изменении внешних стимулов. При этом нивелировались различия между обучением человека и животных: У. отождествлялось с формированием условного рефлекса, механизмами У. оказывались смежность условных стимулов с реакциями и повторение этой смежности, а также замещение внешних двигательных реакций. Выделялись две формы У.: при задавании связей в готовом виде извне и при самостоятельном поиске связей; все структуры У. можно описать с помощью формулы "ситуация — ответная реакция". Продолжением и развитием бихевиористических представлений о процессах У. и обучения стала концепция оперантного поведения и инструментального научения амер. ученого Б.Ф.Скиннера — основоположника *программированного обучения*, в которой сохранилась осн. бихевиористская схема — ситуация, реакция, подкрепление. Гештальтпсихологическая концепция У. основана на принципах структурности и целостности, представлениях о процессе познания и У. как переструктурировании прежних структур опыта (происходящем в соответствии с законами сходства, очередности, завершенности, прегнантности), о несводимости свойств сложных продуктов познания и У. к свойствам их частей. Опыт, относящийся к культурным образованиям (речь, письмо, чтение и пр.) усваивается только через подражание. Облегчение У. при подражании объясняется направлением (с помощью образца) внимания на то, что ранее не входило в структуру опыта. У. в целом имеет две фазы: образование новых форм целостности (успех) и сохранение и воспроизведение новых форм (память). Опыт приобретаетсЯ либо в ситуации самостоятельного спонтанного взаимодействия с объектами, либо через подражание. В работах швейц. ученого Ж. Пиаже систематизированы положения о функционировании различных форм познания и приобретения опыта, составляющих определенную систему взглядов как на процесс, так на структуру У. Виды приобретения опыта различаются по трем основаниям: 1) постепенный или скачкообразный (кратковременный) процесс; 2) неосознаваемый или сознательно контролируемый процесс; 3) отражение явных или неявных связей. В результате выделены также виды приобретения опыта: У. (постепенный, неосознаваемый процесс усвоения любых связей); индукция (постепенное сознательно контролируемое усвоение любых связей); озарение (скачкообразное познание неявных связей, осознаваемое и неосознаваемое); восприятие (скачкообразное усвоение явных связей, осознаваемое и неосознаваемое). В концепции У. амер. психолога Р. Ганье, отразившей влияние информационного подхода к когнитивным процессам, предметом усвоения выступают интеллектуальные навыки (различение, объединение, классификация и др. операции, осуществляемые с помощью знаков), словесная информация (высказывания, утверждения, описания), познавательные стратегии (навыки осуществления У., запоминания, воспроизведения, мышления и т.д.),

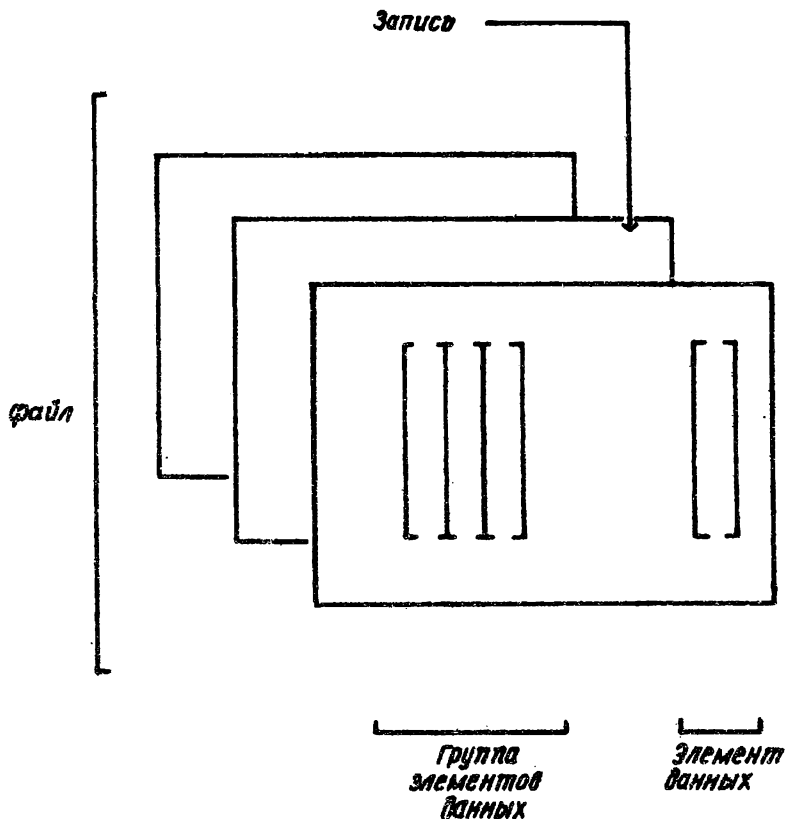
моторные навыки движения, отношения и мотивы. При этом микроструктура У. включает внешние воздействия, внутренние процессы, внешние реакции и подкрепление. Внешние воздействия осуществляются и в начале, и в ходе протекания внутренних процессов, описываемых на основе общей модели переработки информации, включающей внимание и избирательное восприятие, кратковременное запоминание, кодирование и переход в долговременную память, хранение, воспроизведение, генерализацию ответа, внешнее выполнение действий, обратную связь в обучении. Все перечисленные процессы направляются и контролируются специальными процессами контроля исполнения и образами конечных результатов. Эти компоненты модели переработки информации вписываются в осн. фазы процесса У. и составляют их содержание. Общая структура процесса У. содержит фазы восприятия, усвоения, хранения, поиска в памяти и воспроизведения, выполнения и контроля. Параллельно всем фазам осуществляется контроль за их протеканием.

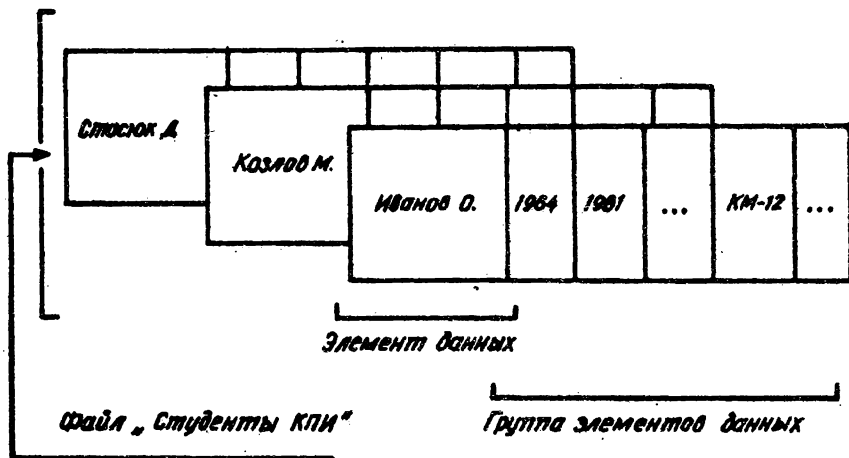
Ряд теорий У. разработан и в отечественной психолого-педагогической науке. Так, ассоцианистская теория У. содержит концептуальные представления об У. и обучении как процессах, построенных на ассоциативных связях. Ассоциация полагается в качестве центрального понятия психической деятельности. Процесс усвоения при таком подходе рассматривается как образование ассоциаций и их систем. В теории поэтапного формирования умственных действий У. рассматривается как система определенных видов деятельности, выполнение которых приводит обучаемого к новым знаниям и умениям. Центральным звеном является действие, как единица деятельности У. Выделяются некоторые характеристики действия (форма, обобщенность, развернутость и освоенность), различные типы ориентировочных основ действий, указываются способы их формирования. В соответствии с теорией учебной деятельности школьников У., играющее ведущую роль в умственном развитии, осуществляется прежде всего через содержание усваиваемых знаний. Развивающий характер учебной деятельности в младшем школьном возрасте связан с тем, что её содержанием являются прежде всего теор. знания. Учебная деятельность строится в соответствии со способом изложения научных знаний, способом восхождения от абстрактного к конкретному. Единицей учебной деятельности выступает учебная задача, решаемая школьниками путем выполнения определенных действий: преобразование условия задачи с целью обнаружения всеобщего отношения изучаемого объекта; моделирования выделенного отношения для изучения его свойств в "чистом виде"; построение системы частных задач, решаемых общим способом; контроль за выполнением предыдущих действий; оценка усвоения общего способа как результата решения данной учебной задачи.

В отечественной педагогической психологии в последние годы наиболее широко используется понятие “учебная деятельность”, в зарубежных теор. конструкциях термины “учение”, “учебная деятельность”, “научение” и др. часто используются как синонимы. См. также *Проблемное обучение*.

Е.Д.Маргулис.

ФАЙЛ (англ. file — досье, картотека) — поименованная совокупность однотипных по структуре и способу использования логических записей, т.е. поименованных совокупностей *элементов данных* одного или различных типов либо совокупностей групп элементов данных (рис.). Содержит большие объемы информации и размещается на внешних носителях памяти ЭВМ. Ф. располагает некоторыми отличительными сведениями, позволяющими отделить один Ф. от другого, определить последнюю запись Ф. и т.д.





Существует простая (рис.) и сложная организация Ф. данных. В сложном Ф. одновременно существует несколько отношений между записями. Назначением его является обеспечение выполнения необходимых видов работ с Ф. за миним. время, причем подлежащие выполнению действия предсказуемы и заранее известны. Существуют разновидности сложного Ф., такие, как мультисписковые, мультисвязанные и т.д. Сложность организации хранения Ф. зависит от конкретной комбинации операций, которые необходимо выполнить над ним (найти запись, прочитать её, модифицировать, удалить запись и т.п.).

А.П.Ильяшенко, М.Е.Козлов.

ФАЙЛОВЫЕ ВИРУСЫ — тип компьютерных вирусов, заражающих любой тип исполняемых файлов, существующих в конкретной операционной системе (ОС). Для MS ДОС осн. объектами заражения являются файлы типа .COM и типа .EXE. Наиболее просто заражаются COM-файлы, представляющие собой почти точную копию участка памяти ЭВМ с загруженной программой. Единственная требуемая настройка при загрузке COM-файлов состоит в загрузке сегментных регистров значениями, соответствующими месту загрузки программы. Значительная часть COM-файлов начинается с команды перехода, обходящей данные, содержащиеся в начале программы. Простейший нерезидентный Ф.в., заражающий COM-файлы в текущем каталоге и дописывающий свое тело в конец файла, можно представить в виде следующей неформальной спецификации.

Шаг 1: Восстановить первые три *байта* программы. Три байта зараженной программы, сохраненные в теле вируса, пересылаются на свое старое место.

Шаг 2: Проверить среду. Проверить версию ОС. Если не подходит, перейти к шагу 11.

Шаг 3: Найти очередную жертву (жертвы). Найти в текущем каталоге очередную СОМ-файл. При неудаче перейти к шагу 11.

Шаг 4: Проверить зараженность потенциальной жертвы. Считать область, позволяющую установить, заражен уже данный файл вирусом или нет. Проверить содержимое этой области. Если файл уже заражен, перейти к шагу 3, иначе перейти к шагу 5.

Шаг 5: Проверить, подходит ли жертва для заражения. Если сумма длин файла и вируса больше 64 К, перейти к шагу 3, иначе перейти к шагу 6.

Шаг 6: Снять атрибут “только чтение” и запомнить в своем теле дату создания программы.

Шаг 7: Обеспечить передачу управления вирусу. Считать первые три байта зараженной программы и записать их в своем теле. Сформировать команду перехода на байт, следующий за концом программы, и записать соответствующие три байта в начало заражаемой программы. При неудаче перейти к шагу 11, иначе перейти к шагу 8.

Шаг 8: Дописать тело вируса в конец заражаемой программы и установить режим дозаписи. Переписать свое тело в конец заражаемой программы. При неудаче перейти к шагу 11, иначе перейти к шагу 9.

Шаг 9: Восстановить дату создания зараженной программы. Записать в элемент каталога, соответствующий заражаемой программе, дату, сохраненную в теле вируса.

Шаг 10: Восстановить атрибут “только чтение” Присвоить заражаемому файлу атрибуты, установленные у файла до заражения и сохраненные в теле вируса. При неудаче перейти к шагу 11, иначе перейти к шагу 10.

Шаг 11: Выход. Восстановить содержание регистров и передать управление программе-вирусоносителю.

Как видно из приведенной схемы, конкретный Ф.в. распространяется не мгновенно по всей файловой системе, а постепенно, по мере заражения файлов в текущем каталоге. Поэтому от момента появления в файловой системе первого зараженного файла до заражения всех файлов в каталоге проходит некоторое время, зависящее от интенсивности использования зараженной программы. Наиболее уязвимыми с точки зрения блокирования размножения Ф.в. являются шаги 6—9. Для блокировки этих шагов обычно используют различные резидентные программы (см. *Антивирусы*). Кроме того, если на шаге 4 вирус использует для маркировки зараженных файлов легко воспроизводимый признак (напр., простановку в поле даты

значения 62 секунды — излюбленный признак для Ф.в.), можно блокировать его распространение, присвоив этот признак всем заражаемым данным вирусом файлам. Этот прием используется так наз. программами-вакцинами.

Тело вируса при заражении может вставляться в начало, конец или середину файла. При вставке в начало файла первые блоки (или все тело) заражаемой программы обычно переписываются в конец, поэтому до передачи управления зараженной программе вирус должен предварительно переписать их на первоначальное место, заменив ими собственный код. С этой целью вирус должен переместить свое тело или хотя бы соответствующую часть своего кода так, чтобы она не была затерта при перезаписи. При вставке в конец файла необходимо обеспечить передачу управления коду вируса до начала работы зараженной программы. Для COM-файлов это обеспечивается заменой нескольких первых байтов программы (обычно трех) на команду перехода к началу вируса (инсталлятору). При этом сами первые байты обязательно должны быть сохранены где-то в теле вируса, что обеспечивает возможность их восстановления (поиск этих байтов — составная часть работы любого фага). Когда инсталлятор вируса получает управление, он в первую очередь восстанавливает измененные вирусом байты в первоначальном виде. Вставка в середину файла встречается редко. Во-первых, этот способ используется узкоспециализированными Ф.в., поражающими определенный класс программ, особенности структуры которого заранее известны, напр., только командный процессор MS DOS. Во-вторых, вставка в середину возможна, если в файле имеется большая легко определяемая область данных (обычно нулей или одинаковых символов). Для вирусов, цепляющихся к концу EXE-файла, вставка в середину возможна, если в заголовке этого файла неправильно указывается его длина. В-третьих, файл может быть заражен несколькими однотипными вирусами (обычно дописывающими свой код в конец файла). В этом случае вирус, первым получивший управление, отгесняется к середине файла последующими. Указанные случаи довольно часто не учитываются создателями антивирусов, в частности детекторов и фагов. В результате файл может быть не детектирован как зараженный или вылечен неправильно.

Помимо отдельных файлов, резидентные Ф.в. заражают и память компьютера. Предельно упрощая, эту память можно рассматривать как еще один файл, который можно заражать, дописываясь в "голову", т.е. в область младших адресов свободного участка памяти, в "хвост", т.е. в область старших адресов свободного участка памяти, и, наконец, в "середину", т.е. в область адресов, уже используемых ОС или какой-нибудь программой (старшие адреса вектора прерываний, буфера и т.д.).

Резидентные Ф.в. можно представлять как состоящие из двух относительно независимых частей: инсталлятора и модуля обработки прерываний, который, в свою очередь, состоит из ряда программы обработки. Несколько упрощая, можно считать, что на каждое

перехватываемое прерывание своя программа обработки. Инсталлятор получает управление при выполнении зараженной программы. Он обрабатывает один раз после запуска зараженной программы и его целесообразно рассматривать как специализированный Ф.в., заражающий оперативную память и, возможно, обычные файлы. В последнем случае инсталлятор можно рассматривать как Ф.в., доработанный для заражения оперативной памяти. Структуру инсталлятора можно упрощенно представить следующим образом:

Шаг 1: Проверить версию MS ДОС.

Шаг 2: Восстановить измененные байты зараженной программы.

Шаг 3: Проверить зараженность оперативной памяти. Если память заражена, то передать управление программе-вирусоносителю, иначе перейти к шагу 4.

Шаг 4: Закрепиться в оперативной памяти. Переписать свое тело в заданный участок оперативной памяти и выполнить некоторые действия по закреплению этого участка за собой.

Шаг 5: Перехватить требуемые прерывания. Заменить адреса в соответствующих элементах вектора прерываний на адреса своих программ обработки.

Шаг 6: Передать управление зараженной программе.

Функционирование резидентного Ф.в. можно представить как две стадии: стадию инсталляции и резидентную стадию. При запуске зараженной программы управление получает инсталлятор, который обеспечивает закрепление вируса в оперативной памяти, перехват требуемых прерываний и, возможно, маскировку (чтобы затруднить свое обнаружение среди резидентных программ). В дальнейшем каждый раз при возникновении одного из перехваченных вирусом прерываний управление получает программа обработки соответствующего прерывания. При этом вирус может анализировать поступивший *запрос* и, в зависимости от его вида, выполнять те или иные действия. Напр., загрузка программы в оперативную память и ее выполнение в MS ДОС реализованы с помощью прерывания 21—2В. Если вирус перехватывает это прерывание, то он может определить имя файла, из которого выполняется загрузка, и заразить программу в этом файле. Способ поиска “жертвы” у резидентных вирусов существенно отличается от способа, используемого нерезидентными вирусами. Нерезидентные вирусы получают управление после загрузки в память зараженной программы, а затем ищут “жертву”, используя *информацию* о расположении выполняемых программ. Если “жертва” найдена, то она заражается, а затем управление возвращается зараженной программе. Резидентные вирусы реагируют на определенные прерывания. Получив управление после перехваченного прерывания, они выполняют соответствующие действия. —

Н.Н. Безруков.

ФЕЛНЕТ — локальная сеть учебного назначения, построенная на основе восьмиразрядной вычислительной машины IO-151. Состоит из блока управления с внешними запоминающими устройствами, печатающим устройством и рабочими местами обучаемых. Рабочее место обучаемых оборудовано персональным компьютером, работающим в операционной системе CP/M с большим количеством программ (трансляторов с наиболее распространенных языков программирования, программ-редакторов, функционирующих на базе WORDSTAR), с базой данных и др. Внешние устройства разделены сетевыми связями. Преподаватель может руководить работой обучаемых и проверять её. Он может запустить любую обучающую программу, послать обучаемому сообщение, перенести изображение с экрана дисплея обучаемого на дисплей управляющего компьютера. Ф. применяют в школах и вузах.

П. Горват.

FLEX — экспериментальная интеллектуальная обучающая система, предназначенная для обеспечения программиста на языке Лисп набором услуг, соответствующих его уровню. Первоначально информация о пользователе (обучаемом) заносится в систему диспетчером или формируется при первом подключении к системе на основании небольшого анкетирования. Модель обучаемого включает статистические данные о его работе в системе и набор ситуаций, считающихся “критическими” для данного обучаемого. Множество всех “критических” ситуаций в каждый момент фиксировано, но может пополняться диспетчером в ходе отдельного диалога с FLEX. Реакция системы на “критические” ситуации и правила их применения также задаются диспетчером. “Критической” ситуацией считается вхождение в последовательность S-выражений, задаваемых обучаемым, некоторых выделенных подвыражений или пар подвыражений. Обнаружение “критической” ситуации вызывает реакцию системы, во время которой может происходить диалог с обучаемым, коррекции S-выражений и изменение модели обучаемого. Изменение модели влечет изменение реакции на некоторую “критическую” ситуацию или исключение данной ситуации из модели, т.е. считается, что обучаемый “усвоил” данную ситуацию и помощь ему более не нужна. Следовательно, FLEX использует накрывающую модель обучаемого. Использование модели обучаемого позволяет FLEX адаптироваться к уровню конкретного пользователя, что дает возможность авторам системы считать её прототипом “интеллектуальной” операционной системы. FLEX разработана в 1985—86 в Институте проблем передачи информации.

В. А. Петрушин.

ФЛИККЕР (англ. flicker — мерцание) — медленные флуктуации электрических токов и напряжений в электровакуумных и газоразрядных электронных приборах, обусловленные испарением атомов вещества катода и др. явлениями. При использовании видеотерминалов Ф. рассматривается как неустойчивое состояние изображения на экране, зрительно воспринимаемое в виде резкого падения и увеличения яркости символов вследствие их регенерации на экране дисплея.

По мере увеличения частоты регенерации достигается такой её уровень, при котором Ф. исчезает, а пользователь воспринимает изображение на экране как непрерывное и однородное. Этот уровень определяет критическую частоту слияния мерцаний. При установлении связи между частотой слияния мерцаний и частотой регенерации изображения необходимо учитывать время послесвечения определенного вида фосфора. Различают следующие виды Ф.: мерцание цветности; цветовые мерцания; мерцание изображения; мерцание полей; межстрочные мерцания; яркостные мерцания.

С.А.Миронченко, В.С.Пашковский.

ФОНД АЛГОРИТМОВ И ПРОГРАММ (ФАП) — система регистрации, испытаний, накопления и распространения алгоритмов и программ. Основная задача ФАП — создание условий широкого использования алгоритмов и программ. ФАП создается по территориально-отраслевому принципу. Территориальные, отраслевые и межотраслевые специализированные ФАП входят в состав Государственного ФАП (ГосФАП). В ГосФАП регистрация алгоритмов и программ осуществляется централизованно. Испытания, накопления и распространения алгоритмов и программ осуществляются организациями, ведущими ФАП.

Обычно ФАП состоят из программного и информационного фондов. В программный фонд включаются готовые для непосредственного использования на *компьютере* программы на машинных носителях информации с необходимыми комплектами программных документов, предназначенные для широкого распространения среди организаций - пользователей. Информационный фонд комплектуется программными документами общего характера, информационными материалами по вопросам формирования и ведения ФАП, разработки и использования программ.

Организации, ведущие ФАП, поставляют программные средства пользователям и оказывают им научно-технические услуги по использованию программных средств. В своей работе организации, ведущие ФАП, тесно взаимодействуют с разработчиками программ. Программные средства, принятые в ГосФАП, приравниваются к печатным работам. ФАП организаций и учреждений в систему ГосФАП не входят.

Л.Д.Бабко.

ФОРМАЛИЗАЦИЯ СЕМАНТИКИ — математически строгое описание способа интерпретации программ, формул, спецификаций и других текстов языка программирования (ЯП). Осн. цель Ф.с. придание текстам ЯП однозначного смысла, для того чтобы можно было точно отвечать на вопросы типа: являются ли два заданных текста (семантически) эквивалентными (т.е. совпадает ли их смысл)? является ли заданная формула, выражающая некоторое утверждение о предметной области (семантически), истинной или ложной? корректна ли заданная программа по отношению к спецификации? корректно ли компилятор транслирует программы заданного ЯП?

Методы Ф.с. можно разделить на модельные, логические и алгоритмические. Модельные методы основаны на построении модели языка. При этом перечисленные выше вопросы сводятся к вопросам о свойствах соответствующей модели. Логический метод заключается в построении некоторого исчисления логического, позволяющего доказывать утверждения о тех или иных свойствах текстов формализуемого ЯП. Для широкого класса ЯП можно воспользоваться к.-л. общим исчислением, дополняя его различными аксиомами. Такой способ Ф.с. наз. аксиоматизацией (см. *Логическое описание предметной области, Верификация программы*). Алгоритмический метод заключается в построении программ, отражающих семантику формализуемого ЯП и позволяющих отвечать на определенный класс вопросов о смысле текстов этого языка. Напр., компилятор для ЯП является одной из форм Ф.с. этого языка, т.к. позволяет вычислять результаты программы при заданных входных данных. Однако такая формализация ЯП весьма слаба и не позволяет даже судить о функциональной эквивалентности двух программ. Выбор того или иного метода Ф.с. ЯП существенно зависит как от ЯП, так и от целей формализации (т.е. от того, на какой класс вопросов должна отвечать заданная формализация).

В.М. Антимиров.

ФОРМАТ ИЗОБРАЖЕНИЯ (от лат. *formatum* — оформление) система организации информации на экране дисплея, предполагающая её разбивку на определенные области, участки, окна, адресуемые элементы и секторы. Определяется программным обеспечением, зависит от особенностей деятельности, цели обучения и характера учебной информации. Важнейшие параметры Ф.и. текстовой информации на экране дисплея: пространственная ориентация изображения (горизонтальная или вертикальная), межстрочный и межзнаковый интервал, длина и количество строк, зонирование экрана; графической информации — фрагментирование, масштабирование и формирование окон.

При разработке Ф.и. учитываются возраст обучаемых (напр., для дошкольников и младших школьников желательны более короткие

строки), возможности и опыт *пользователей*, условия, предоставляемые техническим обеспечением, а также следующие эргономические требования и рекомендации: не следует заполнять весь экран информацией или одной фигурой; важнейшую информацию желательно располагать в середине экрана; повторяющуюся информацию следует размещать на тех же позициях; ориентация текстовой информации на экране дисплея должна быть вертикальной, расстояние между строками должно быть большим, чем высота *символов* (текстовую информацию лучше представлять через строку). Характеристики Ф.и. влияют на психические процессы, протекающие при усвоении *знаний* с помощью компьютера, — восприятие, внимание, память, мышление. Особенности Ф.и. обуславливают эффективность деятельности пользователей, сказываясь на скорости их работы, ошибках, зрительной и умственной утомляемости. Цветовая и информационная перенасыщенность Ф.и., медленное и сложное форматирование вызывают неудовлетворенность, снижение *мотивации* к работе и обучению с помощью компьютера, раздражение и утомляемость пользователя.

В.М.Бондаровская, Т.Дамянова, Н.И.Повякель.

ФОРМАТ СООБЩЕНИЯ — способ расположения и представления сообщений на экране *дисплея*, предполагающий их разбивку на законченные по смыслу, легко идентифицируемые и понимаемые участки и окна. Проектируется с учетом размерных характеристик экрана дисплея, особенностей общего форматирования экрана (см. *Формат изображения*). При разработке Ф.с. важны близость его к установившимся формам, приоритетность и психологическая значимость сообщения, объемов представляемой на экране и в сообщении *информации*, логика расположения её на экране, смысловая значимость сообщения (см. *Читабельность текста*). При выборе Ф.с. нужно учитывать эргономические требования к организации осн. и вспомогательной информации на экране дисплея (см. *Меню*), к использованию различных способов кодирования изображения (см., напр., *Цветовое кодирование изображения*).

Н.И.Повякель.

ФОРТРАН (от англ. FORmula TRANslation — преобразование формул) — *язык программирования* (ЯП), предназначенный для решения научно-технических задач. Является первым ЯП высокого уровня (см. *Уровень алгоритмического языка*). Разработан в 1957 в США. Несмотря на устаревшие концепции, Ф. остается одним из наиболее распространенных ЯП, широко применяется для реализации численных методов. В настоящее время используются стандарты Ф.-IV и Ф.-77. Особенности Ф. — направленность на обработку числовых *данных*, статическое распределение *памяти ЭВМ*, локаль-

ность переменных, невозможность рекурсии. В Ф.-IV нет развитых управляющих структур — полного условного оператора, цикла с предусловием. Из-за специфики Ф. (цикл с постусловием, общие блоки и др.) при работе часто возникают побочные эффекты, некорректная обработка особых случаев и другие труднообнаруживаемые ошибки. Для изучения Ф. созданы различные автоматизированные учебные курсы (АУК) и системы, напр., *Афродита*.

Применение Ф. в качестве инструментального ЯП для создания АУК и систем ограничено, т.к. Ф. имеет слабые возможности для обработки текстов, других нечисловых типов данных и сложных структур. Ф. применяют для реализации имитационных моделей, используемых в обучающих системах и тренажерах.

В.А.Бардадым.

ФРЕЙМ (от англ. frame — рамка) — иерархически организованная структура данных, предназначенная для представления знаний о некоторой стереотипной ситуации. Концепция Ф. была предложена в 1974 амер. ученым М.Минским. Структура Ф. состоит из нескольких уровней. Верхние уровни образованы понятиями, всегда истинными по отношению к предполагаемой ситуации. На нижних уровнях имеются вершины-терминалы (ячейки, слоты), которые заполняются конкретной информацией при вызове Ф. Каждый слот может устанавливать условия, которым должны удовлетворять его значения. Простые условия определяются маркерами. В более сложных случаях задаются отношения между значениями для нескольких слотов. В автоматизированных обучающих системах Ф. применяются для описания отдельных фрагментов предметной области, при организации естественного языкового интерфейса, для описания различных ситуаций, возникающих в процессе обучения. См. также *Фреймовые методы представления знаний*.

О.Н.Золотопуп, В.В.Колос.

ФРЕЙМОВЫЕ МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ — методы декларативного представления знаний, базирующиеся на использовании фреймов. С каждым фреймом ассоциирована информация разных видов, напр., как используют данных фрейм, что может повлечь за собой выполнение фрейма и т.д. Фреймы — наиболее богатый способ представления знаний; этот способ полнее, чем семантические сети и продукции (см. *Продукционные методы представления знаний*), однако труднее для реализации. Фреймы могут быть связаны как с различными аспектами данного явления, так и друг с другом и с некоторым набором специальных процедур. При описании сложных объектов с помощью фреймов возникают проблемы выбора фрейма, перехода от одного фрейма к другому, а также манипулирования фреймовыми структурами данных. С по-

мощью фреймов можно представить как декларативные, так и процедурные знания. Фреймы могут содержать отсутствующие значения, извлекаемые из других фреймов, семантических сетей или продукций — это отличает их от других методов представления знаний. Разработаны специальные языки представления знаний, основанные на идее фрейма. Примерами могут служить KRL, FRL, OWL.

Язык KRL был разработан как формальный язык для представления знаний в системе понимания естественного языка. В нем предусмотрены специальные средства для сопоставления фреймов. Структура фреймов позволяет рассматривать объекты в разных аспектах. Язык выделяется своей универсальностью и полнотой описания. В языке FRL в структуре фреймов делается акцент на создание конкретных экземпляров фреймов. Система фреймов имеет четко выраженное иерархическое построение. В распоряжение пользователя предоставляется расширенный вариант языка Лисп. Язык OWL базируется на синтаксических и семантических структурах предложений английского языка. Как и в языке FRL, в нем применяется иерархия понятий.

В современных разработках идея фрейма используется при описании знаний с помощью структур достаточно высокого уровня абстрактности. Применение высокоуровневых структур знаний в диалоговых системах и, в частности, в автоматизированных обучающих системах представляется весьма перспективным при анализе ответов обучаемых в том случае, когда ответ является связным текстом. Фреймы удобно использовать при разработке моделей предметных областей (по предмету обучения) в тренирующих и обучающих системах. Кроме того, Ф.м.п.з. широко используются в реализации естественноречевого общения между обучаемым и автоматизированной обучающей системой.

О.Н.Золотопун, В.В.Колос.

ФРЛ (англ. FRL — FFrame Language — рамочный (фреймовый) язык) — язык представления знаний, реализованный как надстройка над языком программирования (ЯП) Лисп. Этим определяется работа с данными и внешний вид программы. В обучении применяется для создания интеллектуальных обучающих систем. Осн. типом данных в ФРЛ является фрейм, который представляет собой структуру вида:

(Имя фрейма

(имя первого слота

(имя первой ячейки

(первое данное

(первый комментарий: сообщение)

.....

(i-й комментарий: сообщение))

```

(второе данное
.....))
(j-е данное .....))
(имя второй ячейки
.....))
(имя k-й ячейки
.....))
(имя второго слота
.....))
(имя l-го слота
.....))

```

Значением слота может быть имя другого слота, но уровень вложенности не должен превышать 5. Для доступа к информации во фрейме используется индикаторный путь, состоящий из перечисления имен фрейма, слота, ячейки и данного. Различают фреймы-прототипы и фреймы экземпляры. Фреймы-прототипы хранят общие знания о предметной области (ПО), они являются обобщением понятия типа, используемого в традиционных ЯП. Фреймы-экземпляры содержат конкретные знания о ПО и обобщают понятие структурной переменной.

В ФПЛ реализованы четыре операции над фреймами: FASSERT — ввод в базу знаний фрейма, имя которого указывается в аргументе этого оператора; FPUT — добавление новых данных в фрейм; FGET — извлечение информации из фрейма; REMOVE — удаление информации из фрейма. В ФПЛ существует восемь типов стандартных имен слотов, которые имеют следующее содержание: SVALUE — непосредственно данные; SDEFAULT — информация, используемая по умолчанию; SIF ADDED — имя процедуры, вызываемой при выполнении оператора FPUT для данного фрейма; SIF REMOVED — имя процедуры, вызываемой при выполнении оператора REMOVE для данного фрейма; SIF NEEDED — имя процедуры, вызываемой при выполнении оператора FGET для данного фрейма; SREQUIRE — имена предикатов, определенных на значениях данного слота; AKO — имя родового фрейма (более высокого уровня); INSTANCE — имя видового фрейма (более низкого уровня, заполняется динамически при обработке слота AKO). Кроме традиционных для Лиспа управляющих структур, в ФПЛ имеется возможность передавать управления между фреймами, используя содержимое слота AKO (AKO-связи) и INSTANCE (INSTANCE-связи). Такая организация управляющих структур позволяет формировать сети фреймов. *Интерфейс с пользователем* мотивируется с помощью процедур, имена которых указаны в слотах SIF ADDED, SIF REMOVED и SIF NEEDED.

В.А.Третьяк.

ФУНКЦИОНАЛЬНЫЙ ЯЗЫК — алгоритмический язык, представляющий средства для описания задачи в виде системы взаимно рекурсивных функций. В основе Ф.я. лежит денотационная модель процесса вычислений, согласно которой конечное состояние программы определяется как результат применения некоторой функции к исходному состоянию. В соответствии с технологической направленностью Ф.я. используют для проектирования и спецификации программ (МЕТА-IV, Z), а также как языки программирования для создания интеллектуальных программ и систем (Лисп, Плэнер, Рефал и др.). По уровню проблемной ориентированности Ф.я. разделяются на проблемно-ориентированные языки (ФРЛ), машинно-ориентированные языки (Лисп для Лисп-машин) и универсальные языки (Лисп, Рефал, МЕТА-IV).

В.А.Третьяк.

ЦВЕТОВОЕ КОДИРОВАНИЕ ИЗОБРАЖЕНИЯ — 1) Использование условной цветовой системы для кодирования и передачи сообщений. 2) Выбор цветовых характеристик изображения. Применяется для передачи информации о состоянии или значимости объектов. Ц.к.и. позволяет структурировать информацию, обеспечить её большую читабельность (см. *Читабельность текста*) и идентифицируемость, а также выделять важную информацию. Обеспечивает малое время обнаружения и идентификации информации на экране. Ц. к.и. следует пользоваться при необходимости быстрого обнаружения местонахождения нужного символа на экране с неупорядоченной информацией или на плотно загруженном экране; для выделения сигнала тревоги, предупреждающей информации об изменении к.-л. параметра; для выделения подсказки; при группировке однородной или разнородной информации; при организации многооконного интерфейса. Выбор цветов и их сочетаний определяются возможностями конкретного компьютера и оказывают воздействие на интеллектуальную деятельность пользователя, его работоспособность и эффективность работы.

При Ц.к.и. нужно соблюдать следующие рекомендации. Использовать дополнительные цвета (при увеличенном яркостном контрасте) для привлечения внимания пользователя. При загрузке экрана дисплея большим количеством информации необходимо принимать различные цвета для группирования информации по функциональным признакам (визуальный ключ). Следует минимизировать количество цветов, используемых на экране дисплея одновременно; количество элементов информации, кодированных не основным цветом, не должно превышать 4—5. Нужно использовать цвет фона для различных областей информации и цвет переднего плана для выделения линий внутри областей; нежелательно приме-

нять дополнительный цвет для символов и фона, т.к. это вызывает *утомление зрительное*. Не следует выходить за пределы значений, допустимых для яркостного контраста, необходимо использовать нормальный яркостный контраст изображения (светлые знаки, темный фон) для осн. информации и инверсный яркостный контраст в случае, если требуется высокая точность восприятия. Цвета, используемые для кодирования информации и их применение должны быть общеупотребляемыми. Следует соблюдать единые принципы кодирования различных элементов информации и единую последовательность в использовании цветов; применять: желтый цвет — для предостережения или предупреждения; оранжевый или красный — для прекращения работы, обозначения сбоя или предупреждения о существенной ошибке, зеленый — для обычной информации или кодирования положительного события (напр., положительного ответа системы на *запрос* пользователя); нежелательно применять красный и зеленый цвета для цветовых контрастов, т.к. многие пользователи имеют красно-зеленую аномалию зрения. Для кодирования информации, предъявляемой пользователю непрерывно и необходимой ему только изредка, следует применять нейтральные цвета. Необходимо учитывать и психологические воздействия различных цветов на особенности восприятия и психофизиологическое состояние пользователей в ходе *обучения*. Напр., черный цвет глушит раздражения, помогает сосредоточению, его можно использовать как фон для важнейшей информации. Синий цвет подчеркивает дистанцию, им можно осуществлять пространственное кодирование.

Т.Дамянова, Н.И.Повякель.

ЦЕЛЕВОЕ ОБУЧЕНИЕ — комплекс организационных мероприятий, средств и методов *обучения*, обеспечивающих изучение учебного материала по каждой учебной дисциплине, на всех этапах подготовки специалиста во всех видах *учебной деятельности*. Рассматривая и сопоставляя обучение на различных возрастных уровнях (дошкольное, школьное, вузовское и послевузовское), можно прийти к выводу о необходимости единой целевой установки. Конечные цели обучения по окончании вуза должны определять не только форму и содержание процесса обучения в вузе, но и быть определяющими для процесса обучения на ранних этапах — в средней школе и в процессе дошкольного обучения. Планируя и организовав послевузовское обучения в системе повышения квалификации, нельзя ставить специалиста в жесткие рамки целевых установок. Определенная избирательность в этом отношении — результат творческого подхода зрелой сформировавшейся личности к своему самосовершенствованию.

Современная методология Ц.о. состоит в том, что четко и детально выраженные цели итогов обучения определяют цели

обучения на низких уровнях иерархии: к целям профилирующих (специальных) дисциплин, от них к общинженерным, общетехническим, далее к общенаучным и так вплоть до средней школы. Эта методология характерна и для применения *компьютерной технологии обучения*, которая должна обеспечивать достижение необходимых результатов обучения на всех уровнях.

А.Е.Денисов.

ЧИТАЕЛЬНОСТЬ ТЭКСТА — комплекс свойств, облегчающих узнавание, интерпретацию, понимание, скорость и надежность переработки *информации*, представленной буквенно-цифровыми *символами*, объединенными в смысловые группы (напр., предложения или сложный *текст*). Обусловлена прежде всего познавательными процессами человека. Ч.т. больше зависит от расположения и группировки слов на экране *дисплея*, разбивки по строкам и их объединения в предложения, чем от специфических черт отдельных символов (см. *Читаемость информации*).

И.Ю.Никонова, Н.И.Повякель.

ЧИТАЕМОСТЬ ИНФОРМАЦИИ — комплекс специфических свойств буквенно-цифровых *символов*, позволяющих четко идентифицировать и отличать их друг от друга. Связана со зрительными процессами, включенными в чтение. Можно говорить о читаемости отдельных слов или даже букв (см. *Читаемость текста*). На Ч.и. влияют: физические параметры символа, т.е. физические измерения, форма, начертание и другие характеристики символов *дисплея*, включая рекомендуемую гарнитуру; световые условия, цвет, угол наблюдения и т.п. (*яркость, контраст, цветовой контраст, блескость, горизонтальный и вертикальный угол наблюдения, мерцания*); форматы, т.е. факторы, касающиеся того, как представлена информация (заглавные буквы по сравнению со строчными, количество символов в строке, наличие и размер "полей" и т.п.). Эргономичность значений этих характеристик приводит к уменьшению времени чтения или доли ошибок и соответственно к лучшей читаемости.

И.Ю.Никонова, Н.И.Повякель.

ЧТЕНИЕ ТЭКСТА — знаковое общение *автора* и читателя посредством печатного *текста*, направленное на восприятие и понимание *информации*, содержащейся в тексте, приводящее к росту *знаний* читателя, формированию его личностных смысловых образований. Может выступать необходимой составной частью другой деятельности (напр., учебной или познавательной), являясь её ориентировочной основой. Осн. этапы процесса Ч.т.: 1) декодирование или распознавание отдельных слов; 2) сравнение элементов текста

(слов, терминов, словосочетаний) с эталонами, хранящимися в долговременной памяти, и уточнение их значений в зависимости от *контекста*; 3) установление семантических и синтаксических связей между отдельными словами, словосочетаниями, предложениями и более крупными текстовыми образованиями (сверхфразовыми единствами, смысловыми блоками) и соотнесение полученной информации с системой знаний.

Условием достижения адекватного *понимания* текста читателем является сформированный механизм переработки семантической информации, включающий следующие познавательные операции: общую ориентировку в тексте, его содержании и строении; структурирование текста, позволяющее выделить его внешнюю (поверхностную) структуру; группировку материала по отдельным смысловым темам; сжатие информации, которое благодаря укрупнению её единиц позволяет удерживать материал в оперативной памяти на протяжении всего времени чтения; переструктурирование текстового материала, которое определяется семантикой текста и целями чтения, ведет к построению внутренней (смысловой) структуры сообщения; разворачивание содержания текста в мысленной или речевой форме на основе схемы припоминания.

Ч.т. — не одностороннее воздействие текста на читателя, пассивно воспринимающего его, а активное взаимодействие между автором и читателем. Такое взаимодействие выражается, во-первых, в ориентации автора текста на предполагаемого читателя, предвидение им мотивов и целей чтения, имеющегося у читателя уровня знаний, его интересов и потребностей; во-вторых — в активных диалогических реакциях читателя на воспринимаемый текст. Наиболее распространенные из них: согласие — несогласие, *вопрос — ответ*, сомнение, удивление, дополнение. *Диалог* с текстом предусматривает также прогнозирование развития его содержания, выдвижение гипотез и личностное включение читателя в текст, т.е. сопоставление воспринимаемой информации со своими знаниями, взглядами, ценностными ориентациями. Понимание текста не всегда приводит к реконструкции смысла, заложенного в текст его автором. Читатель как активный субъект мышления, понимания, коммуникации строит свой “встречный текст”, который базируется на смысловой основе исходного сообщения, но может дополнять, доконструировать эту основу своим пониманием проблемы, освещаемой в тексте. Такое дополнение, привнесение своего смысла в текст зависит от знаний, установок читателя, целей его чтения, мотивов той деятельности, в которую это чтение включено.

Одним из осн. условий оптимизации процесса Ч.т. является правильная его организация, включающая постановку цели чтения, его планирование, выбор оптимального способа работы с текстом. К наиболее распространенным способам Ч.т. относятся: быстрый просмотр текста, в ходе которого создается самое общее представление

о его содержании; выборочное чтение, предназначенное для поиска определенной информации; полное последовательное чтение без текущего анализа текста; штудирующее чтение с проработкой содержания текста и осмыслением его структуры. Для оптимизации работы по анализу текста используются *алгоритмы чтения*, представляющие собой определенную последовательность умственных действий при взаимодействии с текстом. Чтение по алгоритму позволяет существенно увеличить его скорость без нарушения качества понимания текста. Наиболее распространенные алгоритмы чтения: дифференциальный, имеющий три блока (составление смысловых рядов, выявление смысловых рядов, выявление *ключевых слов*, выделение осн. концепта текста), и интегральный, включающий выделение осн. идеи текста, ведущих смысловых тем, фактического материала.

Н. В. Чепелева.

ЭВРИСТИЧЕСКИЕ СРЕДСТВА (от греч. *εὕρισκω* — нахожу) — *средства решения проблемных познавательных задач*, которые несут для *решателя* определенную *информацию* и дают *возможное Э.с.* благодаря этому *уменьшить уровень трудности задачи*. Всякое Э.с. характеризуется силой, т.е. тем, в какой мере его применение уменьшает (или может уменьшить) трудность рассматриваемой задачи, и широтой сферы применимости, т.е. объемом класса задач, трудность которых можно уменьшить благодаря применению этого средства.

Различают материальные, материализованные и идеальные Э.с. Напр., при решении задачи, связанной с анализом работы к.-л. технического устройства, его действующая *модель* может служить материальным Э.с. В качестве материализованных Э.с. выступают печатные инструкции, графические схемы, устные указания руководителя работы и т.п. Те же указания, когда решающий задачу человек вспоминает их, либо та же схема, как она существует в его представлении, или существующий в его сознании наглядный образ анализируемого предмета, а также любые *знания* субъекта, которые можно использовать для решения задачи, — это идеальные Э.с. Кратко рассмотрим некоторые виды Э.с.

Оперативные знаковые системы характеризуются (применительно к задачам некоторого класса F) тем, что: а) из комп. лентов этих систем можно строить предметы задач класса G , служащих моделями задач класса F ; б) решатель обладает *операторами*, позволяющими производить над указанными компонентами *операции*, системы которых могут обеспечивать решение задач класса G ; в) использование задач класса G в качестве подзадач класса F дает возможность решить или облегчить решение последних. Напр., для решения арифм. задач можно привлекать различные знаковые

системы, с помощью которых осуществляются соответственно “способ предметного моделирования”, арифм. и алгебр. способы решения.

Эвристические предписания о выполнении тех или иных процедур, способствующих решению задачи. В отличие от *алгоритмов* и *квазиалгоритмов* здесь не требуется эффективность или квазиэффективность предусматриваемых предписанием операций. Напр., предписание, направленное на получение квазиалгоритма (“учебного алгоритма”), обеспечивающего построение матем. объекта на основе его генетического определения, требует: 1) выделить операции, необходимые для построения объекта; 2) найти рациональную их последовательность; 3) определить логические условия их выполнения; 4) выделить дополнительные операции, производимые в случае невыполнения логических условий; 5) построить “учебный алгоритм”.

Эвристические рекомендации. Примером могут служить советы по решению матем. “задач нахождение” (принадлежащие амер. математику Д.Пойа): “Рассмотрите неизвестное. Постарайтесь припомнить знакомую задачу с тем же или подобным неизвестным. Сохраните только часть условий, отбросив остальные; в какой мере теперь определяется неизвестное? Как можно его варьировать? Сумеете ли вы вывести что-нибудь полезное из данных? Сможете ли вы придумать другие данные, из которых можно было бы определить неизвестное?”

Стратегии решения задачи, намечающие общую организацию процесса решения. Таковы, напр., выделенные при изучении *творческой деятельности конструкторов стратегии* поиска аналогов, комбинирования, конструирования и др.

Г.А.Балл.

ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ компьютерной технологии обучения — соотношение экономических результатов и затрат при реализации программно-методического инструментария *автоматизированного обучения*, создания, обогащения *баз знаний* по различным предметным областям, контроля знаний с использованием компьютера. Расчет Э.э. необходим для оценки и выбора наиболее целесообразных направлений развития этих средств и методов, определения масштабов внедрения мероприятий, связанных с *компьютерной технологией обучения* (КТО). Экономическим расчетам эффективности подлежат все виды деятельности по разработке и использованию средств КТО: разработка и совершенствование *автоматизированных обучающих систем* (АОС); разработка *автоматизированных учебных курсов* (АУК); разработка методических материалов по проектированию и применению средств КТО; функционирование специальных учебных *фондов алгоритмов*

и программ; автоматизированное обучение и контроль знаний без отрыва и с отрывом от производства (на базе учебных центров).

Объектом анализа Э.э. использования средств КТО является соотношение показателей затрат ресурсов (материальных, трудовых, финансовых, временных и др.) и результатов (количества обученных *пользователей*, количества сотрудников, прошедших контроль знаний, количества разработанных учебных курсов и т.д.). В расчетах показателей Э.э. средств КТО учитываются все общественно-необходимые затраты, связанные с их использованием.

Для индивидуального и группового обучения без отрыва от производства рассчитываются затраты: на приобретение АОС; на приобретение (или разработку) отдельных АУК или библиотеки АУК; связанные с освоением АОС и библиотеки АУК системными программистами в организации-пользователе; на заработную плату *обучаемым* во время обучения и контроля знаний; эксплуатационные затраты на функционирование компьютера при обучении и контроле знаний. При определении Э.э. деятельности специализированных фондов по внедрению средств КТО в организациях-пользователях рассчитываются затраты: на проектирование и создание специализированных фондов; связанные с разработкой и комплектованием фондов; на разработку методических материалов по применению средств КТО; на содержание и эксплуатацию фондов в части затрат на централизованные накопление, хранение, тиражирование и поставку средств КТО; суммарные сопутствующие затраты всех пользователей на приобретение и освоение этих средств; суммарные текущие затраты всех пользователей, связанные с эксплуатацией средств КТО.

При расчете показателей Э.э. деятельности учебных центров по использованию средств КТО для групповых видов обучения и контроля знаний с отрывом от производства определяются затраты: на проектирование и оборудование центра; на освоение *преподавателями* средств КТО; на содержание и эксплуатацию учебных центров (включая затраты на заработную плату преподавателям и эксплуатационные затраты на содержание компьютеров); суммарные затраты организаций-пользователей на выплату командировочных расходов обучаемым и оплату курсов обучения. Для расчета перечисленных стоимостных показателей используются такие данные: среднемесячная заработная плата одного обучаемого и одного преподавателя; коэффициенты начислений на заработную плату; количество обученных и проконтролированных пользователей в одной организации и во всех организациях, применяющих средства КТО; количество обучаемых в одной группе; количество организаций-пользователей фондов; количество преподавателей в учебном центре; количество обучаемых в учебном центре; среднее количество АУК, приобретенных организацией-пользователем; среднее количество АУК, распространяемых через фонд; среднее количество АОС,

распространяемых через фонд; среднее количество АУК, которые ведет один преподаватель; среднее количество комплектов АОС и АУК, распространяемых через фонд; среднее количество АУК, которым обучается один специалист; время обучения одного пользователя одному курсу; расход машинного времени при освоении АОС и АУК преподавателями, системными программистами и организаторами обучения; расход машинного времени при обучении; средняя отпускная цена одного машино-часа компьютера с учетом коэффициента распределения времени его работы в режиме мультипрограммном; текущие расходы на содержание и эксплуатацию ПЭВМ за один час; средняя стоимость разработки и внедрения АОС; средняя стоимость разработки АУК; средняя отпускная цена фонда при поставке АОС и АУК и услуг по их внедрению; средняя стоимость обучения одного пользователя одному курсу с помощью КТО; средняя стоимость мероприятия по контролю знаний и т.д. Рассчитанные на основе перечисленных данных показатели капитальных и текущих затрат сравниваемых технологий обучения сопоставляются при обязательном соблюдении принципа тождественности сравниваемых вариантов по результатам обучения. Используя варианты сопоставления различных технологий обучения и контроля знаний и полные перечни исходных показателей, отражающих конкретные условия применения средств КТО, можно с экономической точки зрения обосновать целесообразность выбора или разработки определенной технологии для предприятий, организаций или народного хозяйства в целом.

С.А.Дорожкин, В.М.Кругликова.

ЭКРАНА РАЗМЕРЫ — размерные характеристики экрана дисплея. Должны выбираться в соответствии с особенностями решаемых обучаемым задач (напр., графических), а также объемом информации, которая должна ему предъявляться. Если экран дисплея слишком мал для выполняемой задачи, повышается плотность выводимой информации, что, в свою очередь, увеличивает время поиска. Неэргономичность Э.р. (несоответствие характеристикам решаемых задач и особенностям восприятия человека) может привести к снижению работоспособности обучаемого и эффективности его работы.

В.М.Бондаровская, Н.И.Позякель.

ЭКСПЕРТНАЯ ИГРА — программный модуль, предназначенный для извлечения экспертных знаний в процессе игры между экспертом и подсистемой извлечения экспертных знаний *экспертной сист. змы*. Извлечение экспертных знаний в Э.и. ведется с помощью диалога в терминах выбранной предметной области (ПО). В ходе игры накапливаются знания, организованные в виде независимых модулей.

Хотя осн. целью Э.и. является накопление в базе знаний сведений о ПО, эта задача не ставится перед экспертом в явном виде. Наоборот, у него создается нацеленность на решение чисто *игровых задач* типа: максимизировать или минимизировать требуемые параметры, набрать наибольшее количество очков, рационально истратить выделенные ресурсы и т.п. Следовательно, цели эксперта в процессе игрового взаимодействия не совпадают с осн. целью Э.и. — извлечением экспертных знаний. В области образования Э.и. можно использовать для извлечения знаний по диагностике ошибок *обучаемого* и формированию сценария обучения в зависимости от состояния модели обучаемого.

Дальнейшую обработку экспертных знаний, извлеченных в процессе игрового взаимодействия, можно проводить автоматически, что представляется весьма затруднительным и малоэффективным, или привлечь на этом этапе специалиста по инженерии знаний. Второй путь дает возможность более эффективно проанализировать и структурировать полученную *информацию*. Задача инженера по знаниям состоит в том, чтобы правильно оценить действия эксперта в ходе игры и окончательно сформировать конкретные модули базы знаний.

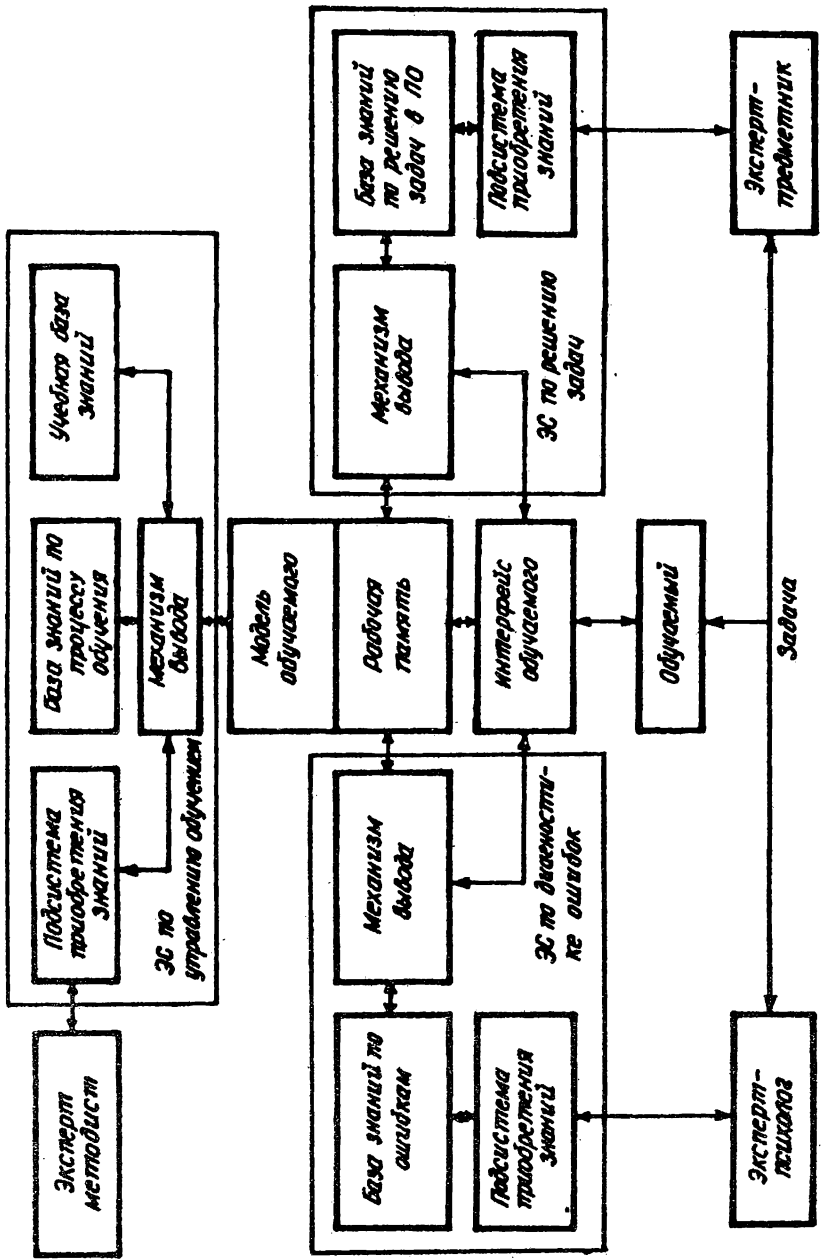
В.В.Колос.

ЭКСПЕРТНАЯ ОБУЧАЮЩАЯ СИСТЕМА (ЭОС) — *обучающая система*, реализующая функции управления обучением, диагностики ошибок обучаемого и решения задач в некоторой предметной области (ПО) на основе знаний экспертов и показывающая результаты на уровне экспертных. В основе архитектуры ЭОС лежит следующая осн. модель процесса обучения. Имеется некоторая цель обучения, выраженная в терминах текущих характеристик обучаемого. Пока цель не достигнута, повторяется такая последовательность действий: на основании текущего состояния обучаемого и методики обучения генерируется очередная задача (здесь задача понимается широко, как любая информация, требующая ответных действий обучаемого); ответ обучаемого сравнивается с эталонным решением и на основании различий производится диагностика ошибок обучаемого; по результатам диагностики корректируются текущие характеристики обучаемого. В соответствии с этой моделью ЭОС представляет собой (рис.) совокупность трех взаимодействующих экспертных систем (ЭС): по решению задач в изучаемой ПО; по диагностике ошибок обучаемого; по планированию процесса управления обучением. ЭС по решению задач (ЭС РЗ) предназначена для выработки эталонного решения задачи. Она может представлять собой: собственно ЭС, построенную на знаниях эксперта по решению задач в данной ПО, если ПО плохо формализуема; решатель задач в некоторой хорошо формализуемой ПО; программы поиска или выбора решения данной задачи из конечного (возможно большого)

набора решений, если обучение ведется на заранее заготовленной совокупности задач. ЭС по диагностике ошибок обучаемого (ЭС ДО) предназначена для выявления неправильных представлений обучаемого об изучаемой ПО на основе сравнения его ответа с эталонным. Парадигма ЭС ДО — диагностика; симптомами служат различия в ответах обучаемого и ЭС РЗ, а предписаниями — изменение модели обучаемого (МО). Она формализует знания эксперта-психолога. ЭС по управлению процессом учения (ЭС УУ) представляет собой ЭС по планированию в ограничениях, накладываемых имеющимся учебным материалом. Она формализует знания эксперта-методиста о методиках обучения.

Взаимодействие ЭОС с обучаемым происходит следующим образом. ЭС УУ формирует в соответствии с текущей целью очередное задание для обучаемого, которое передается одновременно в ЭС РЗ. Далее ЭС ДО сравнивает решение обучаемого с решением, полученным ЭС РЗ, и на основании различий пытается установить, какие неправильные представления обучаемого о ПО могли привести к расхождениям. В результате диагностики меняется представление ЭОС об обучаемом, отраженное в МО, и управление снова получает ЭС УУ, которая уточняет текущую цель и формирует новое задание. Взаимодействие с обучаемым происходит через *интерфейс*, который может содержать средства текстового, графического или речевого ввода—вывода, *лингвистический процессор* и т.п.

В структуре ЭОС большое значение имеют *базы знаний (БЗ)*: учебная БЗ для данной ПО; модель обучаемого; БЗ о возможных ошибках обучаемого; БЗ о процессе обучения. Учебная база знаний (УБЗ) описывает не только осн. *понятия* и методы решения задач в ПО, но и содержит определения понятий, описания методов, примеры, упражнения и задачи. В отличие от БЗ ЭС по решению задач, УБЗ должна явно отражать структуру ПО и стратегические знания о методах решения задач. С другой стороны, УБЗ можно рассматривать как представление ограничений, в рамках которых ЭС УУ планирует обучение. МО содержит информацию о состоянии знаний обучаемого: как общие, интегрированные характеристики, так и те, которые отражают усвоение им текущего учебного материала. Первоначально МО формируется при предварительном тестировании обучаемого. В терминах МО выражается цель обучения. БЗ об ошибках обучаемого (БЗО) содержит каталог возможных ошибок обучаемого и правила выдвижения и проверки гипотез о неправильных представлениях обучаемого, приведших к данной ошибке, на основе различий между решениями, предложенными обучаемым и ЭС РЗ, а также текущего состояния МО. БЗ о процессе обучения (БЗПО) содержит знания о планировании и организации процесса обучения, общих и частных методиках обучения. Для БЗПО предложена иерархическая структура, в которой выделен уровень педагогических, стратегических и тактических знаний. Знания



представлены в виде правил, определяющих тип обучающего воздействия. Выбор очередного правила определяется специальными метаправилами, зависящими от текущей цели обучения и МО. В целом БЗПО можно рассматривать как *метазнания* по отношению к остальным БЗ.

В соответствии с целями функционирования различают консультационные, диагностирующие, управляющие и сопровождающие ЭОС. Консультационная система предназначена для оказания помощи обучаемому в виде выдачи информации по его *запросу* или решению предложенной им задачи с последующим объяснением (если обучаемый потребует), как была получена представленная информация или решение. Консультационная ЭОС состоит из *учебной среды* информационно-справочного или решающего типа и подсистемы объяснения, а также, возможно, МО. Диагностирующая система предназначена для указания обучаемому на его неправильные представления о ПО, вследствие чего он допускает ошибки при решении определенного типа задач. Диагностирующая ЭОС состоит из интерфейса, ЭСРЗ, ЭСДО и МО. Часто диагностирующие системы наз. интеллектуальными тренирующими или *экспертными тренирующими системами*, т.к. их применяют для тренировки решения задач, когда последовательность задач предлагается *преподавателем* или генерируется системой на основании параметров, задаваемых преподавателем или самим обучаемым. Управляющая система предназначена для управления познавательной деятельностью обучаемого. Она является расширением диагностирующей ЭОС знаниями о целях функционирования системы, имеющемся учебном материале и *стратегиях* обучения. Различают обучение понятиям и умениям (навыкам); соответствующие программы наз. обучающими и тренирующими. При всей относительности этого различия осн. отличием *обучающих программ* от тренирующих является наличие в первых цели ознакомления обучаемых с новым материалом. Сопровождающая система предназначена для слежения за деятельностью *пользователя* при работе в некоторой (инструментальной) системе и оказания ему помощи при обнаружении ошибочных или нерациональных действий.

В.А.Петрушин.

ЭКПЕРТНАЯ СИСТЕМА (ЭС) — система *искусственного интеллекта*, решающая сложные практически значимые задачи в некоторой, как правило, плохо формализуемой предметной области (ПО) на базе *знаний*, полученных от человека-эксперта. Демонстрирует при этом качество решения, не уступающее решениям экспертов, и обладает способностью объяснять (обосновывать) процесс получения своего решения. ЭС и инструментальные средства для их разработки классифицируются в зависимости от типа решаемых *задач*

задач (табл.). Имеется два типа *пользователей* ЭС: 1) эксперты, которые сами или при посредничестве специалистов по *представлению знаний* (инженеров-когнитологов) создают *базу знаний*; 2) конечные пользователи, применяющие ЭС при решении своих задач.

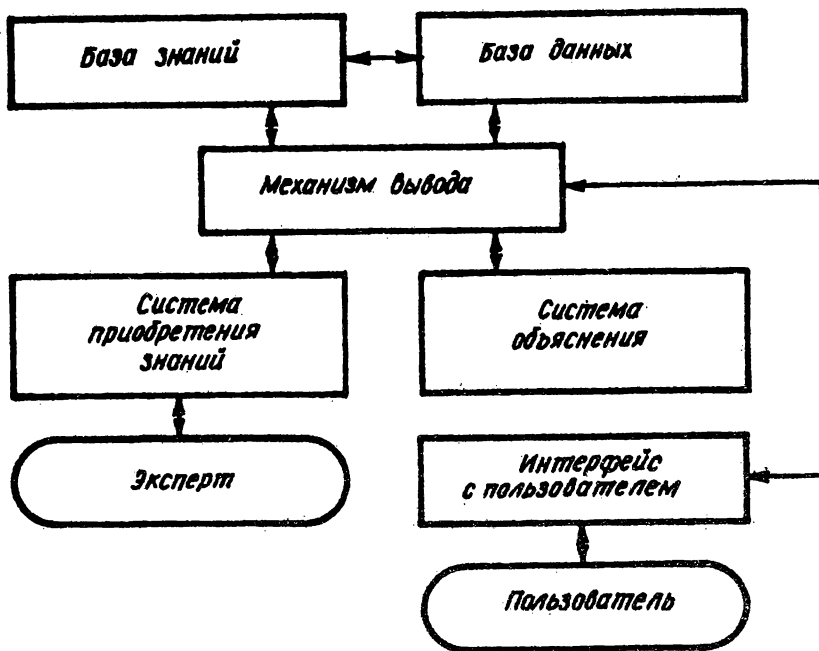
Основные типы задач, решаемых экспертными системами

Тип	Описание задачи	Применение
Идентификация	Построить описание ситуации по наблюдаемым данным	Распознавание речи, анализ сцен, анализ сигналов РЛС
Прогноз	Вывести вероятные следствия из заданных ситуаций	Предсказание погоды, демографические и коммерческие прогнозы
Диагностика	На основании наблюдаемых симптомов сделать заключение о вероятных нарушениях в системе	Диагностика болезней, технических устройств, знаний обучаемых
Проектирование	Построение сложного объекта из элементарных операций при соблюдении заданных ограничений	Компоновка блоков ЭВМ, проектирование зданий
Планирование	Построение описания действия из элементарных (проектирование плана действия)	Планирование эксперимента, поведения робота, учебной программы; автоматическое программирование для ЭВМ
Мониторинг	Слежение за поведением с целью предупреждения о критических ситуациях	Слежение за состоянием системы большого после операции, воздушного движения

Обобщенная структура ЭС приведена на рис. *База знаний* (БЗ) содержит *информацию* в виде фактов и правил, необходимую для решения задач требуемого типа. Рабочая память отражает текущее состояние системы при решении задачи (*данные* о задаче и процессе её решения). Механизм вывода представляет собой общий *алгоритм* решения задач данного класса, реализуемый, как правило, в виде *интерпретатора*. Применение его к БЗ о конкретной ПО, задаваемой экспертом, и к данным о текущей ситуации, задаваемой пользователем, дает решение требуемой задачи. *Интерфейс* с пользователем предназначен для взаимодействия с ним во время решения задачи и, в зависимости от последней, может использовать средства анализа фраз на естественном языке, выбора из *меню*,

графического ввода—вывода и др. Система объяснения служит для объяснения (обоснования) процесса получения системой решения текущей задачи. Наличие этого компонента позволяет пользователю проследить ход решения задачи и убедиться в обоснованности каждого шага, что весьма важно, т.к. ЭС построены на эмпирических знаниях экспертов. В силу тезиса о том, что разработка ЭС никогда не бывает завершена, т.к. ЭС всегда отражает текущее состояние динамически изменяемых знаний о заданной ПО, многие ЭС, и обязательно инструментальные экспертные системы, содержат систему *приобретения знаний*, посредством которой в *диалоге* с экспертом и/или инженером-когнитологом создается и модифицируется БЗ.

Методология построения ЭС учитывает эволюционный характер процесса её разработки и предполагает, что на каждом уровне эволюции система совершает виток жизненного цикла разработки, включающего этапы идентификации, концептуализации, формализации, реализации и тестирования ЭС. На этапе идентификации осн. действующими лицами являются эксперты и инженеры-когнитологи. В их задачи входит: определение участников процесса проектирования и распределение ролей между ними, неформальное



описание задачи и оценка целесообразности разработки ЭС, определение целей разработки и необходимых для нее ресурсов (время, оборудование и т.п.). На этапе концептуализации действующие лица те же, но теперь они решают задачу выявления всех знаний, относящихся к процессу решения поставленной задачи (объекты, данные, гипотезы, отношения, подзадачи, правила, *стратегии, вопросы, типы ответов, объяснения* и т.п.). Этап формализации осуществляется инженером-когнитологом, который выражает ранее выявленные знания формально на некотором языке, определяет пригодность доступных *инструментальных средств* для решения данной задачи и принимает решение о выборе подходящего средства. Результатом данного этапа является формальная *модель* решения задачи. На этапе реализации в действие вступают программисты, которые, используя выбранные инструментальные средства, воплощают формальную модель в *программе* для ЭВМ. При этом они тесно взаимодействуют с инженерами-когнитологами, т.к. в процессе реализации проявляются неувязки в спецификациях данных, правил и схем управления. Этап тестирования состоит в оценке качества работы прототипа ЭС и разработке рекомендаций по его модификации. Оценка производится экспертами, которые затем помогают инженерам-когнитологам сформулировать требования к её модификации. В зависимости от несоответствия поведения прототипа ЭС желаемому поведению модификация может заключаться в уточнении элементов БЗ либо в пересмотре выбранных методов представления знаний (переконструирование), либо в поиске других моделей решения задачи (переформулирование).

Для разработки ЭС применяются различные инструментальные средства. По степени проблемной ориентированности и, следовательно, по уменьшению затрат, необходимых для построения ЭС, их можно расположить в следующем порядке: 1) *универсальные языки программирования (Фортран, Паскаль, Си)*; 2) *языки искусственного интеллекта (Лисп, Пролог, Смолток-80)*; 3) *универсальные языки или языки искусственного интеллекта, расширенные набором специальных типов данных, функций и процедур, позволяющих строить ЭС или их компоненты (PIE)*; 4) *языки представления знаний (ROSIE, OPS5)*; 5) *специализированные программные системы, которые, опрашивая и предоставляя средства редактирования пользователю, позволяют наполнять БЗ и параметризовать встроенные в систему механизмы вывода и вытчи объяснений (KAS, TEIRESIAS)*. Представителей последнего класса инструментальных средств называют инструментальными экспертными системами (ИЭС) или оболочками экспертных систем. Применение ЭС в образовании связано с построением интеллектуальных обучающих систем или экспертных обучающих систем (SCHOLAR, SOPHIE, EXCHECK, САКИО, BETS).

С.П.Кудрявцева, В.А.Петрушин.

ЭКСПЕРТНАЯ ТРЕНИРУЮЩАЯ СИСТЕМА — разновидность *экспертной обучающей системы*, предназначенная для оказания консультационной помощи и *диагностики знаний* обучаемого при решении задач. Как правило, позволяет сформировать навыки и умения решения определенного класса задач, но не содержит объяснение нового учебного материала. Э.т.с. предлагает *обучаемому* задания, анализирует результаты их выполнения и строит гипотезы о наличии у него ошибочных представлений. Часто Э.т.с. имеют два осн. режима функционирования: консультационный, при котором задание генерируется системой или вводится обучаемым, но решается всегда системой с одновременной выдачей объяснения решения; тренирующий, при котором задание обязательно решается обучаемым, а решение диагностируется системой. Осн. компонентами Э.т.с. являются *экспертная система* (ЭС) по решению задач, ЭС по диагностике ошибок, *интерфейс* и *управляющая программа*. В Э.т.с. можно выделить базовую подсистему, не зависящую от конкретной *предметной области*, и предметно-зависимую часть. В базовую подсистему входят *управляющая программа*, средства организации интерфейса, средства ведения *модели обучаемого* и механизм поиска ошибок. В качестве модели обучаемого можно использовать, напр., *продукционную модель* Слимана, в которой поведение обучаемого при решении задачи моделируется набором правил, содержащим как “хорошие”, так и “плохие” (ошибочные) правила. В предметно-зависимую часть входят: *решатель* задач, включающий механизм *генерации задач* и набор “хороших” правил, сочетание которых обеспечивает решение задач; набор “плохих” правил, отражающих типичные ошибки обучаемых при решении задач данного типа; сообщения системы, включающие *тексты* условий задач, приглашения или подсказки для ввода ответов обучаемых, объяснения способов решения задач, а также текстовые эквиваленты “хороших” и “плохих” правил.

М.Прието Мендес.

ЭКСПЕРТНЫЕ ЗНАНИЯ — *знания эксперта по предметной области* (ПО). Необходимы для описания ПО, её анализа, преобразования и выбора. Знания о ПО в силу экспертного характера во многих случаях бывают противоречивы, приближительны, многозначны. Это приводит к необходимости использовать не одну, а несколько *моделей предметной области*, а также несколько источников знаний. Анализ ПО связан с её доопределением и переформулированием текущего состояния. Преобразование ПО связано с переходом одного её состояния в другое. В этом случае *данные* как добавляются в ПО, так и удаляются из неё. Выбор ПО связан с переходом *экспертной системы* (ЭС) из одной альтернативной области в

другую. При выборе ПО ЭС осуществляет поиск той из альтернативных или взаимодополняющих областей, которая адекватна решаемой задаче. Э.з. должны отражать точность или противоречивость знаний о ПО, полноту или неполноту знаний о ней, статичность или динамичность ПО. Форма представления Э.з. зависит от конкретной ПО и специфики ЭС. Одной из комбинированных форм представления знаний о ПО, сочетающих в себе возможность описания отношений между понятиями ПО и процедурами выполнения некоторых действий, являются сети фреймов (см. Методы представления знаний). Если рассматриваются ЭС в области образования, то к знаниям эксперта по ПО добавляются знания эксперта-психолога по диагностике ошибок и эксперта-методиста по управлению обучением (см. Экспертная обучающая система).

В.В.Колос, С.П.Кудрявцева.

ЭКСТЕРН (ЭКспериментальная Система ТЕстирования и Регулируемого Научения) — автоматизированная обучающая система для создания обучающе-контролирующих учебных программ на компьютерах серии ЕС. Разработана в 1980 на факультете вычислительной математики и кибернетики МГУ им. М.В.Ломоносова. Рассчитана на работу с четырьмя группами пользователей: авторами (пишут и отлаживают обучающие программы); обучаемыми (изучают учебный материал, подготовленный авторами); преподавателями (проводят занятия с обучаемыми); сопроводителем (обеспечивает правильность функционирования Э.). Взаимодействие пользователей с системой осуществляется с помощью пяти типов директив. Директивы общего назначения предназначаются для входа в систему, отправки сообщений, использования кадрами обучающей информации и выхода из системы. Директивы сопроводителя применяются для санкционирования доступа пользователей к системе. Директивы обучаемого позволяют управлять дидактическим материалом, выбирая режимы работы с обучающей программой. Директивы автора предназначаются для написания и отладки обучающей программы с использованием редактирования, тестирования, трассировки и т.д. на авторском языке *Дидакт*.

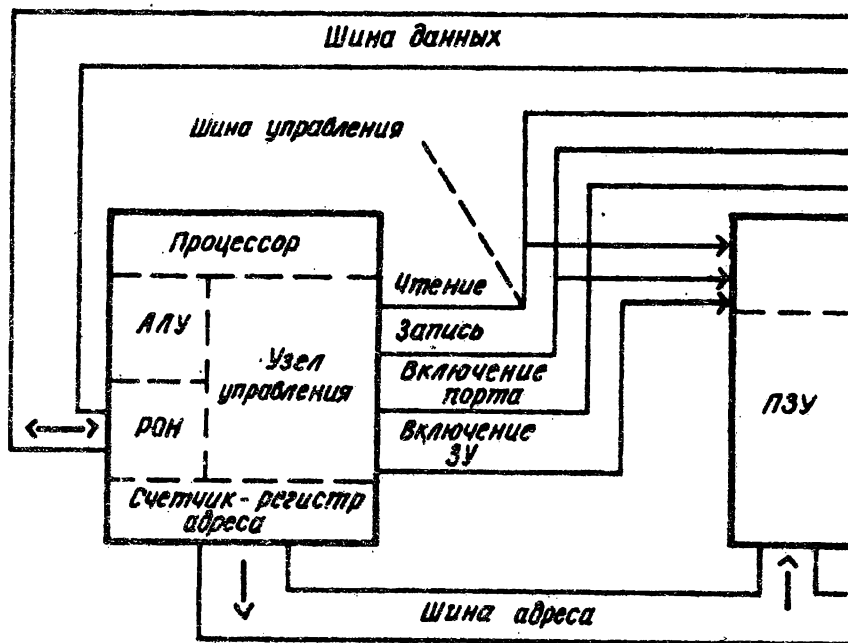
Направление процессом обучения в Э. основывается на использовании факта правильности ответа обучаемого и простой модели обучаемого, представляющей собой совокупность счётчиков правильных, неправильных, непредвиденных и неуложившихся по времени ответов обучаемого, а также его путь следования по блокам учебного материала. Автор обучающей программы имеет возможность средствами авторского языка программирования проверить состояние модели и в зависимости от результатов проверки предпринять то или иное управляющее воздействие.

В.А.Третьяк.

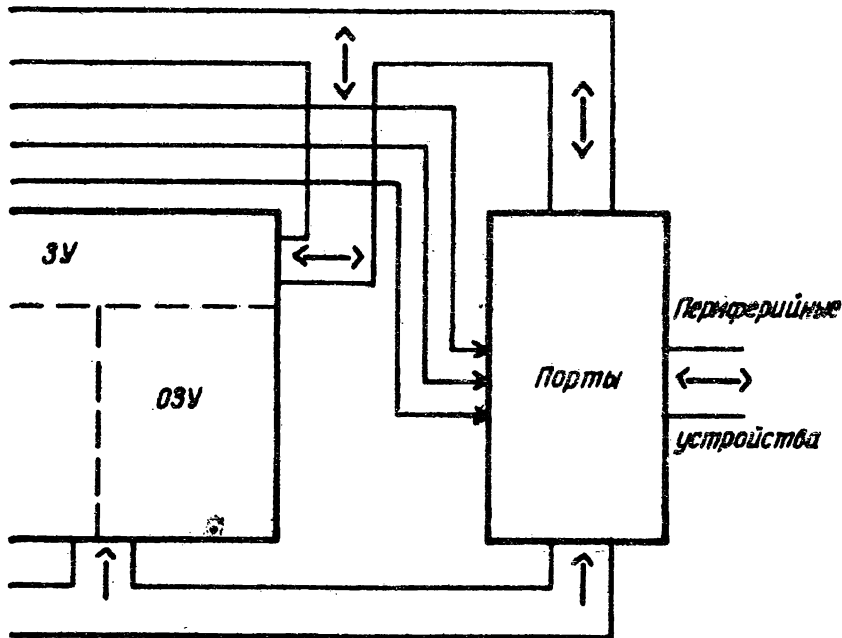
ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ЭВМ), компьютер — вычислительная машина, основными элементами которой являются электронные приборы. Характеризуется высокой скоростью обработки информации, универсальностью, простотой обслуживания. Различают ЭВМ аналоговые и цифровые. Аналоговые ЭВМ (АВМ) оперируют информацией, представленной в виде электрических напряжений и /или токов, зависимости между которыми и изменения которых аналогичны соответствующим зависимостям и изменениям реальных величин в решаемой задаче. Цифровые ЭВМ (ЦВМ) оперируют информацией, представленной в цифровом виде, т.е. реальные величины в решаемой задаче, их зависимость и изменения, алгоритм решения задачи представляются в виде таблиц чисел. Вследствие того, что для решения задачи на АВМ не требуется детально разработанный алгоритм, результат решения матем. задач и задач управления может быть получен быстрее, чем на ЦВМ. Однако ЦВМ обладают большей универсальностью, более высокой точностью вычислений и процесс решения на них более автоматизирован. Существуют комбинированные (гибридные) ЭВМ, которые совмещают положительные качества АВМ и ЦВМ. Они используются при решении задач, требующих больших объемов матем. операций и сложных или неясных алгоритмов.

Для решения задач обучения наиболее часто используют ЦВМ (рис.). В состав упрощенной структурной схемы ЦВМ входят процессор, постоянное и оперативное запоминающие устройства (ПЗУ и ОЗУ), порты ввода и порты вывода информации, шины адреса, данных и управления. Шины адреса и данных представляют собой совокупность проводов, каждый из которых связывает между собой все блоки ЭВМ. Количество проводов в шине адреса равно количеству разрядов (разрядности) кода адреса; то же и для шины данных. Шина управления — совокупность проводов, соединяющих процессор с управляемыми им блоками ЭВМ. Процессор состоит из узла управления, арифметико-логического устройства (АЛУ), регистров общего назначения (РОН) (внутреннее запоминающее устройство (ЗУ) для временного хранения обрабатываемой или управляющей информации) и счетчика-регистра адреса. Каждый порт ввода позволяет процессору получать информацию из периферийных устройств, а порт вывода — передавать информацию в периферийные устройства. В ЗУ хранятся программы работы ЭВМ, в ПЗУ — программы, которые всегда необходимы для работы ЭВМ, а также программы часто встречающихся задач. В ПЗУ, в частности, хранится программа “загрузчик”. Выполняя эту программу, процессор загружает через порты ввода из периферийных устройств в ОЗУ программы пользователей.

Выполнение программы в ЭВМ происходит следующим образом. Процессор выставляет на шине адреса код адреса ячейки ЗУ, содержащей код команды, и на управляющей шине — сигналы



“включение ЗУ” и “чтение”. Первый из этих сигналов подключает ЗУ к шине данных, а второй заставляет ЗУ выдать на шину данных код, содержащийся в ячейке с заданным адресом. Процессор принимает код команды, записывая его в соответствующий внутренний регистр (один из РОН). Узел управления, получая код из этого регистра, выясняет, нужна ли для выполнения этой команды дополнительная информация. Если не нужна, АЛУ выполняет получаемую команду. Если нужна, в счетчике-регистре прибавляется единица и читается следующая по порядку ячейка ЗУ. Код из этой ячейки записывается в другой внутренний регистр. Чтение из ЗУ подобным образом производится в одной команде столько раз, сколько необходимо для получения всей информации, относящейся к данной команде. После этого АЛУ выполняет команду, а результат узел управления записывает либо в свободный внутренний регистр процессора (или освободившийся в процессе выполнения команды), либо в ЗУ. В последнем случае код адреса должен содержаться в информации, относящейся к данной или одной из предыдущих команд. При записи в ЗУ к шине адреса подключается внутренний регистр, содержащий код адреса, а к шине данных — внутренний регистр, содержащий код результата. Таким образом, процессор выставляет нужный адрес, код данных и сигналы “запись”



и “включение ЦУ”. Сигнал “запись” записывает в ячейку с данным адресом код с шины данных. Подобным же образом производится чтение из порта и запись в порт, однако в этом случае процессор, кроме сигналов “чтение” или “запись”, подает сигнал “включение порта”. После выполнения команды в счетчике-регистре прибавляется единица, и процессор переходит к чтению следующей команды. Существуют команды перехода, т.е. команды, указывающие, что после каждой из них должна выполняться не следующая по порядку команда, а команда, адрес которой указан в коде команды перехода или в информации, относящейся к команде перехода. При выполнении команд перехода код адреса следующей команды переписывается из соответствующего внутреннего регистра процессора в счетчик-регистр адреса, и в дальнейшем выполнение программы происходит обычным образом, по порядку возрастания адресов, пока не встретится код следующей команды перехода. При включении ЭВМ начинает работу всегда с одного и того же адреса, обычно нулевого. Этот адрес относится к ПЗУ.

Цифровые ЭВМ бывают универсальными, проблемно-ориентированными и специализированными. Универсальные ЭВМ могут решать задачи разнообразных типов. Но такие машины сложны и дорогостоящи. Проблемно-ориентированные ЭВМ обладают высоким

быстродействием только при решении определенного типа задач (напр., вычислительных и задач управления) и менее высоким при решении задач других типов. Благодаря этому они имеют существенно более простую структуру и значительно меньшую стоимость. Специализированные ЭВМ решают задачи только одного типа, благодаря чему они имеют наименьшую стоимость. Вычислительные проблемно-ориентированные ЭВМ ориентированы на решение матем. задач. Управляющие проблемно-ориентированные ЭВМ предназначены для управления реальными процессами или объектами. Они перерабатывают информацию, получаемую непосредственно с управляемого объекта, и вырабатывают управляющие воздействия с такой скоростью, которую требуют характеристики этого объекта.

Первая ЭВМ была создана в середине 40-х гг. 20 в. Характерные черты ЭВМ первого поколения (МЭСМ, БЭСМ, БЭСМ-2, М-2, "Стрела", "Урал", "Минск-1"): использование электронных ламп в основных и вспомогательных схемах; наличие параллельного арифм. устройства; разделение *памяти ЭВМ* на быстродействующую оперативную ограниченного объема (выполненную на электронно-лучевой трубке или на ферритовых сердечниках) и медленную внешнюю большого объема (*накопители* на магнитных барабанах и лентах); применение полупроводниковых диодов и магнитных сердечников в логических элементах машин, перфолент и перфокарт как *носителей информации*. В ЭВМ второго поколения (напр., БЭСМ-6) основу технической базы составили полупроводниковые диоды и транзисторы. Эти машины отличались более высокой надежностью, меньшим потреблением энергии, более высоким быстродействием. Технической базой машин третьего поколения (напр., ЕС ЭВМ) стали малые интегральные схемы и микросхемы средней степени интеграции. Переход на интегральные схемы способствовал повышению надежности ЭВМ, уменьшению их габаритов и потребляемой ими энергии. В основе ЭВМ третьего поколения заложен принцип параллельной работы различных блоков и периферийных устройств. Технической базой машин четвертого поколения стали большие интегральные схемы. В одной такой схеме объемом всего лишь в доли кубического сантиметра умещается блок, занимавший в ЭВМ первого поколения целый шкаф. Производительность машин четвертого поколения достигает нескольких десятков миллионов *операций* в секунду. Вырос объем памяти. Наряду с усовершенствованием традиционных устройств памяти на *магнитных дисках* и лентах разрабатывается память без движущихся частей. Общий объем внешней памяти в крупнейших машинах четвертого поколения превышает 10^{14} байт. Отличительной чертой вычислительных машин пятого поколения будет применение параллельных структур для обработки смысловой информации. У систем пятого поколения будет характерная элементная база — не только сверхбольшие интегральные схемы, но и созданные на их основе ЭВМ, имеющие некоторое сходство по

структуре с *искусственным интеллектом*. В ЭВМ пятого поколения будет широко использоваться оптоэлектроника. Поскольку скорость света значительно выше скорости электронов, повысится как быстродействие машины, так и пропускная способность линий связи, по которым информация должна поступать в ЭВМ.

В зависимости от быстродействия и объема памяти различают *микро-ЭВМ*, *мини-ЭВМ*, средние ЭВМ, большие ЭВМ и супер-ЭВМ. Микро-ЭВМ имеет наименьшее быстродействие и малый объем памяти. У супер-ЭВМ эти параметры наибольшие. Развитие микроэлектроники способствует улучшению параметров ЭВМ. Так, у многих микро-ЭВМ четвертого поколения быстродействие и объем памяти значительно выше, чем у супер-ЭВМ второго поколения или больших ЭВМ третьего поколения.

В.П.Козлова, В.Л.Леонтьев, З.Ш.Примакова.

ЭЛЕКТРОННАЯ ИГРА — игра, сценарий которой реализован на электронных схемах. Как правило, имеет простой сценарий и не предполагает управления со стороны обучающей. Э.и. успешно применяются для развития и *обучения* детей дошкольного и младшего школьного возраста, выступая в качестве средств *мотивации* познавательной деятельности.

Э.Л.Ивахненко, Е.Д.Маргулис.

ЭЛЕКТРОСТАТИЧЕСКОЕ ПОЛЕ — электрическое поле неподвижных электрических зарядов, осуществляющее взаимодействие между ними. Источниками Э.п. являются, напр., видеотерминалы с электронно-лучевой трубкой. Его возникновение и величина зависят от величины следового потенциала и от материала, из которого сделан экран видеотерминала, а также от уровня относительной влажности в помещении (см. *Микроклимат кабинета автоматизированного обучения*). Величина Э.п. может колебаться от 8 до 75 кВ/м и убывает пропорционально расстоянию от экрана до наблюдателя. *Обучаемый*, сидящий на расстоянии 0,3 м от экрана, имеет напряженность Э.п. на поверхности одежды около 0,6 кВ.

Для устранения Э.п. нужно регулярно проводить влажную уборку помещения кабинета информатики и вычислительной техники и протирать экран видеотерминала спиртом с целью снятия статического электричества, а также строго соблюдать правила техники безопасности (заземление видеотерминала и пр.).

А.С.Коваленко.

ЭЛЕМЕНТ ДАННЫХ — наименьшая единица наименованных данных. Может состоять из любого количества *байтов* или *битов*. Э.д. является логически неделимой структурой данных (представление *пользователя* о данных, не зависящее от способа их хранения в

базе данных (БД), из которых в итоге составляются структуры всех остальных типов данных (совокупность свойств, отличающая некоторый набор данных и определяющая множество значений, которое могут принимать эти данные). Э.д. не может подразделяться на меньшие типы данных, не теряя при этом смысла для пользователя.

Любая БД состоит из Э.д. и связей между ними. В БД может существовать много различных типов данных, поэтому необходимо формальное описание, позволяющее устанавливать связи между типами Э.д. Схема Э.д. может определять его имя, категорию его значения (число, строка *символов* и т.п.), особенности представления значения (двоичное представление, с фиксированной точкой, в символьном виде и т.п.) и другие правила проверки достоверности значения.

А.П.Ильяшенко, М.Е.Козлов.

ЭЛКОН (Электронный КОНструктор) — *графический редактор* табличного типа. Включает семейство композиционных таблиц (КОМТАБ) — специальный набор матриц различной размерности, ленточной или страничной памяти, объединяющей графические базовые элементы (ГБЭ) и библиотеки функций, реализующих теоретико-множественные и аффинные операции и преобразования (точечные преобразования плоскости (пространства) в себя, при которых прямые линии переходят в прямые и сохраняется их параллельность). *Операция* разметки элементов опорных понятий как совокупности точек интерпретируется цветовой раскраской. Отличительная черта Э. — рассмотрение каждого ГБЭ как отдельного множества с реализацией над ним всех множественных и аффинных операций.

Формирование графических *моделей* объектов *предметной области* (ПО) состоит из двух этапов. На первом этапе необходимо построить графические модели для каждого элемента *множества элементов опорных понятий*. На втором этапе, используя множества действий и отношений, следует построить набор графических композиций, необходимых для процесса решения *задачи* соответствующего типа. Для создания и коррекции графического элемента служат "линза" — таблица 128 x 128 и "карман" — таблица 32 x 32. Линза позволяет увеличить в четыре раза создаваемое и/или корректируемое изображение элемента. Карман обеспечивает параллельное представление изображения элемента, представленного в линзе, в формате библиотечного элемента. Управление процессом создания графического элемента в режиме линзы реализуется при помощи клавиш управления курсором.

КОМТАБ — множество от 1 до N табличек с размерностью ГБЭ является одним из осн. компонентов табличного редактора. Каждая табличка служит для размещения в КОМТАБ графического элемента.

Пользователю в сеансе взаимодействия доступна одна из возможных таблиц КОМТАБ. Для обеспечения процесса создания графических композиций в среду редактора включается библиотека экранных функций, которые определяются над каждым компонентом редактора.

В режимах работы с библиотекой ГБЭ *обучаемый* может сформировать и/или изучить графическую модель каждого элемента из множества ГБЭ, используемых для построения графического изображения (модели мира). При необходимости обучаемый может включить в библиотеку новую графическую модель на уровне описания её в виде ГБЭ. Библиотека ГБЭ представляет собой ленту, разбитую на конечное число ячеек (кадров). Каждая ячейка (кадр) определяет ГБЭ. В некотором смысле систему можно определить как модель машины Тьюринга с конечным числом состояний и динамически задаваемым алфавитом. Выделенным элементом алфавита может быть пустой ГБЭ. Просмотр ленты осуществляется последовательно в направлении вверх или вниз. Лента располагается на экране вертикально. Просмотр кадров ленты осуществляется как вверх, так и вниз. Кадр, содержащий ГБЭ, необходимый в данный момент деятельности, должен быть установлен между специальными стрелками-указателями. Каждый кадр содержит выпуклое замкнутое подмножество точек плоскости. Это подмножество и определяет ГБЭ. В процессе целенаправленного выделения ГБЭ обучаемый, используя функции системы, формирует множество конечного числа подмножеств точек плоскости.

В режиме электронного конструирования обучаемый формирует графические композиции объектов решаемой задачи в одной из таблиц КОМТАБ. Получаемое графическое изображение объектов изучаемой ПО представляет собой интерпретационную модель построенного множества подмножеств точек плоскости (ГБЭ).

Целенаправленность *учебной деятельности* обучаемого проявляется в последовательном применении экранных функций системы. Первоначально он может разместить в КОМТАБе одно подмножество ГБЭ. Далее, выбрав следующий набор ГБЭ, применить к нему определенную функцию КОМТАБа. Завершить процесс конструирования графической модели обучаемый может в любой момент взаимодействия. Каждому режиму соответствует свой набор экранных функций. В библиотеку экранных функций Э. включены функции, реализующие операции над аффинными множествами, обеспечивающие связь режимов между собой и ряд специальных преобразований ГБЭ и КОМТАБ как множеств.

Оси. правила конструирования графических моделей представляются при помощи аффинных преобразований: ПЕРЕНОС, МАСШТАБ, ПОВОРОТ, СИММЕТРИЯ и множественных операций: ОБЪЕДИНЕНИЕ, ПЕРЕСЕЧЕНИЕ, ИНВЕРСИЯ, СИММЕТРИЧНАЯ РАЗНОСТЬ. При помощи указанной группы преобразований реализуются функции: ОТПЕЧАТКА ГБЭ, ПОВОРОТЫ, СИММЕТРИЯ

(горизонтальная и вертикальная), ПОСТАВИТЬ И СТЕРЕТЬ ТОЧКУ, ЗАПИСЬ РАБОЧЕГО ГБЭ, ПРОСМОТР ВНИЗ, ПРОСМОТР ВВЕРХ, ПЕРЕНОС, ИЗМЕНИТЬ ТЕКУЩИЙ РАЗМЕР ТОЧКИ, РАБОТА С ПАМЯТЬЮ, СЖАТИЕ ЧАСТИ ПОЛЯ КОМТАБ, ВЕДЕНИЕ ФАЙЛОВОЙ СИСТЕМЫ, ПЕРЕМЕЩЕНИЕ ЧАСТИ КОМТАБ и т.п. Для получения композиции используются множественные операции: ОБЪЕДИНЕНИЕ, ПЕРЕСЕЧЕНИЕ, ИНВЕРСИЯ; логические функции: "XOR, OR, AND" и функции определения и замены цвета изображения ГБЭ и КОМТАБ. Дополнительно реализованы функции: поставить точку текущего размера, удалить точку текущего размера, изменить текущий размер точки, сменить рисунок, быстро перемещение, создание ГБЭ, работа с памятью, наложение ГБЭ на поле, сжатие части поля в ГБЭ, печать полей и библиотеки, ведение файловой системы, увеличение размера сжимаемого участка, уменьшение размера сжимаемого участка, перемещение части рисунка, печать КОМТАБ и библиотеки.

Последовательность применения экранных функций определяет целенаправленность деятельности обучаемого в среде системы, хотя их использование (применительно к выбранным ГБЭ) может носить произвольный характер. Однако более целесообразно, с точки зрения достижения учебного эффекта, связать применение этих функций со структурой конструируемой модели объекта или процесса. Описание динамических процессов в среде построенных моделей графических объектов задается при помощи группы аффинных преобразований, объединенных специальным протоколом. Обращение к каждой функции осуществляется при помощи функциональных клавиш, клавиш управления курсором и программируемой клавиатуры.

М.М.Дворина, Э.Л.Ивахненко, А.Е.Стрижак, А.В.Чигорев.

ЭПИСТЕМИЧЕСКАЯ ЗАДАЧА (от греч. *ἐπιστήμη* — знание) — то же, что и *задача усвершенствование знаний*.

ЭРГОНОМИКА КЛАВИАТУРЫ — требования к *клавиатуре* с учетом деятельности *пользователя* и особенностей конкретного компьютера. Формируются с целью определения соответствующих правил выбора блоков клавиш или их конструирования, обеспечивающих удобство в пользовании и способствующих увеличению надежности, точности и скорости работы пользователя. В основу Э.к. компьютера кладется учет антропометрических, биомеханических, психофизиологических характеристик его пользователя, а также психологических особенностей деятельности по вводу и редактированию данных в режиме *диалога*.

Особую роль при выборе и конструировании клавиатуры компьютера играют эргономические требования к клавиатуре как к

элементу *рабочего места* (в частности, размерные, цветовые и гигиенические характеристики). Клавиатура должна быть легкой. Угол наклона рабочей плоскости — в пределах от 0 до 15°, высота — не более 30 мм (измеряется по среднему ряду кнопок). Клавиатура должна быть изготовлена из материала с низким коэффициентом теплопроводности, не обладающего свойством накапливать электрический заряд. Поверхность её должна быть матовой, с коэффициентом отражения не более 40%. Материал, из которого изготовлена клавиатура, не должен быть источником выделения химических соединений. Цвет её корпуса должен совпадать с цветом корпуса монитора. Клавиатура должна отделяться от монитора, свободно перемещаться по рабочей поверхности, снабжаться опорными приспособлениями, изготовленными из материала с высоким коэффициентом трения и препятствующими неумышленному ее сдвигу. В конструкции клавиатуры с высотой передней кромки более 20 мм нужно предусматривать свободное пространство от нижнего ряда кнопок до передней кромки шириной 80—100 мм, предназначенное для опоры кистей рук.

При конструировании клавиатуры необходимо учитывать характеристики кнопок (форма и размер их рабочей поверхности, цвет, свободный ход кнопки, наличие тактильной обратной связи и усиление нажатия): рабочая поверхность приводного элемента кнопки должна иметь прямоугольную форму и быть вогнутой; миним. размер рабочей части приводного элемента кнопки — 12,5 мм; свободный ход кнопки — в пределах от 0,8 до 4,8 мм и одинаков для всех кнопок одной клавиатуры; усилие нажатия кнопки — в пределах от 0,25 до 1,5 Н. Кроме того, при конструировании клавиатуры должен быть решен комплекс программных, технических и психологических задач, в результате чего определяются виды функциональных блоков, их наполнение, компоновка и размещение в рабочей зоне. В клавиатуре должно быть обеспечено удобство работы двумя руками. Кнопочное поле должно быть расчленено на функциональные группы кнопок. При расположении функционально сгруппированных кнопок нужно соблюдать принцип, согласно которому часто используемые поля должны располагаться справа внизу, а редко используемые — слева вверху; кнопки в пределах функциональных групп — слева направо и сверху вниз с учетом содержания деятельности пользователя. Функциональные группы кнопок должны выделяться одним или несколькими из перечисленных способов: цветом; формой и/или размером кнопок; местом расположения. Расстояние между ближайшими краями приводных элементов соседних кнопок — не менее 1 мм. Устройства управления, указывающие режим работы, должны иметь световую индикацию. Индикацию таких органов управления необходимо размещать рядом с ними — сверху или сбоку.

Важнейшее место в организации продуктивной работы пользователя с клавиатурой занимают психолингвистические и психо-семиотические характеристики надписей на клавишах, принцип организации логограмм и буквенных аббревиатур. Надписи на кнопках должны быть читабельными (см. *Читабельность текста*), четкими и легко читаемыми (см. *Читаемость информации*), легко идентифицируемыми и устойчивыми к истиранию. Учет требований психологии и эргономики при выборе и проектировании клавиатуры компьютера начиная с ранних стадий разработки повышает их потребительские свойства и позволяет преодолеть ряд трудностей в организации эффективной работы пользователей. Использование эргономичных клавиатур при компьютеризации обучения снижает утомляемость.

С.А.Миронченко, В.С.Пашковский, Н.И.Повякель.

ЭСКИЗ — комплекс программных средств, представляющий собой *операционную среду для преподавателей и студентов*. Предназначен для ПЭВМ типа IBM PC и PS/2. Э. обеспечивает: целевое планирование учебных и вспомогательных работ; концептуальный доступ к программным средствам; многокритериальную экспертную ранжировку качества любых объектов; классификацию *понятий*, объектов и определяемых процессов; консультации; поиск и исполнение определяемых процессов; смену набора определений; просмотр активных определений; запись определений в *базу знаний*; редактирование текстовых *файлов*, изображений и звуковых эффектов; преобразование *текстов* с помощью универсального макрогенератора; вычислительные работы; разработку учебных матем. *моделей*; генерацию концептуальных и логических *задач*; подключение учебных средств; выбор и исполнение программных модулей; выход в стандартную *операционную систему* (DOS) и возврат к точке выбора; справочное обслуживание.

Ю.И.Лобанов.

ЯЗЫК ОПИСАНИЯ КУРСОВ (ЯОК) — *авторский язык программирования*, предназначенный для написания обучающе-контролирующих *программ учебного назначения*. Разработан в середине 70-х гг. в Институте кибернетики имени В.М.Глушкова АН Украины. Пробраз ЯОК — язык *Курсрайтер-III*, разработанный фирмой IBM. ЯОК является базовым для широко распространенных *автоматизированных обучающих систем* СПОК, СПОК-ВУЗ, АОС-ВУЗ, АОС-М, АОС-ВУЗ/микр и др.

Алфавит ЯОК, реализованого в системе АОС-М, составляют три группы *символов*: буквенные символы — все прописные буквы латинского и русского алфавитов; цифровые символы — 0,1,2,3,4,

5,6,7,8,9; специальные символы — “+” (плюс), “—” (минус); / (косая черта); () (круглые скобки); * (звездочка — любой символ в ответе обучаемого); (амперсанд — любая строка символов в ответе обучаемого); “ ’ ” (апостроф); “,” (запятая); “[]” (квадратные скобки); “<” (меньше); “=” (равно); “>” (больше). Операторы ЯОК образуются только из перечисленных выше символов. В строковых константах разрешается использование любых допустимых для конкретного компьютера символов. Данные ЯОК содержатся в счетчиках чисел, буферах, регистрах, счетчиках времени, переключателях, параметрах, дополнительной памяти ЭВМ. Счетчики чисел обозначаются Sn, где $n=0+31$, и предназначаются для хранения целых чисел. Буферы — Bn, где $n=0+5$, используются для хранения текстовой информации, причем в буфере B0 хранится последний ответ обучаемого. Регистры — Rn, где $n=0+5$, используются, как правило, для формирования и хранения меток курса. При написании программы автор может использовать три счетчика времени: T0 — текущее время дня, T1 — время работы с курсом с начала сеанса, T2 — текущая дата. Для управления работой обучаемого автор может использовать переключатели — Sn, где $n=0+31$, и параметры — Pn, где $n=0+31$. Переключатели и параметры могут находиться только в одном из двух состояний — “включен” или “выключен”. Дополнительная память (обозначение A, l, m, где l — номер позиции, m — количество символов) предназначается для хранения информации любого типа, допустимого в ЯОК. В обозначениях всех типов данных может использоваться косвенная адресация, т.е. номер ЯОК-переменной; позиция и длина адресуемой части дополнительной памяти могут определяться также через значение некоторого счетчика. Если, напр., значение счетчика C2 равно 5, то обозначение BС2 эквивалентно обозначению B5.

Осн. программная единица в ЯОК — *автоматизированный учебный курс*. Он состоит из последовательности фрагментов, которые выполняются друг за другом. Изменение порядка выполнения фрагментов осуществляется с помощью операторов передачи управления. Управление программой со стороны обучаемого осуществляется с помощью директив ЗАВЕРШИ, ПОМОГИ, КОПИЯ, НАУЧИ и ИДИ. Обучаемый, как правило, изучает учебный курс в течение нескольких сеансов. Сеанс работы начинается с вызова программы и заканчивается выдачей директивы ЗАВЕРШИ.

Различают три уровня операторов ЯОК. Операторы первого уровня используются для обозначения типа фрагмента и управления точкой перезапуска программы после выдачи директивы ЗАВЕРШИ; операторы второго уровня — для классификации и анализа ответа обучаемого; операторы третьего уровня — для обработки данных в ЯОК-программе. Фрагмент начинается с оператора первого уровня

и заканчивается следующим оператором первого уровня либо меткой END. Вложенность фрагментов не допускается. Фрагмент ЯОК-программы (рис.) состоит из операторов группы вопроса и операторов группы ответа. К операторам группы вопроса относятся оператор первого уровня (заголовок фрагмента) и все последующие операторы третьего уровня, которые располагаются перед первым оператором ввода ответа или первым оператором второго уровня. Первый оператор ввода ответа или первый оператор второго уровня и все последующие операторы до конца фрагмента образуют группу операторов обработки ответа. К ним относятся также операторы второго уровня. Перед любым ЯОК-оператором может находиться одна или несколько меток. Если оператор имеет два или больше *операндов*, то они разделяются косой чертой.

Пример ЯОК-программы:

[ВОПРОС]

QU "Сколько месяцев в году"
"содержат по тридцать дней?"

LD 1/P2

CA "одиннадцать" "11" "все" "кроме февраля"

TY "Совершенно верно".

WA "четыре" "4"

TY "Неправда. Все месяцы, кроме февраля"

TY "содержит по 30 дней".

PA 5

DE

BR RT

UN "А если подумать?"

UN "Опять неверно. Ответьте еще раз".

UN "Правильный ответ — 11 месяцев".

BR PR

[END]

В приведенном примере ЯОК-программа начинается с метки ВОПРОС, заканчивается меткой END и состоит из одного фрагмента. Метки всегда заключаются в квадратные скобки. Фрагмент начинается оператором QU, который выводит на экран *текст* вопроса, причем вторая часть *вопроса* выводится оператором продолжения. "Включение" счетчика P2 (оператор LD) означает, что все латинские буквы в ответе обучаемого перед помещением в буфер ВО будут перекодированы в русские буквы. Поскольку во фрагменте отсутствует оператор ввода ответа, то операция ввода ответа в буфер ВО осуществится при передаче управления на оператор CA. Затем содержимое ВО последовательно сравнивается с каждым из образцов в операторе CA. При первом же совпадении на экран выдается сообщение "Совершенно верно" и выполняется переход на метку END, в противном случае управление передается на оператор WA.

Фрагмент

Операторы группы вопроса

Заголовок фрагмента

Операторы обработки данных

Операторы группы ответа

Точка ввода ответа

Анализ ответа

Классификация ответа

Операторы обработки данных

Если ответ обучаемого совпал с образцом WA, то на экране на 5 с (оператор PA) появляется соответствующее сообщение, после чего происходит очистка экрана (оператор DE) и выполняется оператор BR на начало фрагмента; иначе выполняется группа операторов неопределенного ответа UN. При первом неопределенном ответе выполняется первый оператор UN — выдается реплика "А если

подумать?”, и управление возвращается на ввод ответа. При втором неопределенном ответе выполняется аналогично первому, второй оператор UN и т.д. В данной программе в третьем операторе UN на экран выдается правильный ответ и выполняется оператор BR на следующий фрагмент. Поскольку программа состоит из одного фрагмента, то этот оператор завершает программу.

В.А.Третьяк.

ЯЗЫК ПРЕДСТАВЛЕНИЯ ЗНАНИЙ — *язык программирования* (ЯП), ориентированный на представление и обработку *знаний*. Я.п.з. можно разделить на следующие группы: языки обработки символьной информации (*Снобол-4*, *Рефал*, *Лисп*); языки, ориентированные на поиск решений в пространстве состояний и доказательство теорем (RLANNER, QLISP); языки, поддерживающие *фреймовые методы представления знаний* (КРЛ, ФРЛ, Ф-язык); языки, предлагающие объектно-ориентированный подход к реализации знаний (*Смолток*); языки логического программирования (*Пролог*). Каждый ЯП является и Я.п.з. При этом в качестве декларативных знаний выступают описания, а в качестве процедурных — последовательности *операторов*, реализующие тот или иной *алгоритм* (см. *Методы представления знаний*). Традиционные ЯП неудобны для *представления знаний*, т.к. построить такое представление — весьма трудоемкий и сложный процесс.

Базовыми структурами языков обработки символьной информации являются *символы*, строки символов и списки. В связи с тем, что в основе всех видов интеллектуальной деятельности лежит процесс обработки и преобразования символьной информации, языки символьной обработки полезны для представления знаний.

Для языков, ориентированных на поисковые *задачи*, характерны достаточно простые и хорошо определенные *модели предметных областей* и относительно сложные методы обработки, поэтому здесь осн. внимание уделено разработке средств управления, а не структурам *данных*.

В результате развития работ по структурному *программированию* стало преобладать представление, согласно которому центральное место в *программе* занимают структурированные данные, которые организуют процедурную часть программы. В это же время в связи с осуществлением работ по *искусственному интеллекту* были разработаны методы представления знаний в виде *фреймов*, что нашло свое воплощение в таких Я.п.з., как КРЛ, ФРЛ, OWL, Ф-язык. Характерные черты этих языков: представление данных в виде *фреймовых структур*; модель *предметной области* (ПО) как иерархически организованное множество *понятий* и конкретная *модель ситуации* как совокупность взаимосвязанных экземпляров

этих понятий; представление связей и закономерностей в виде *процедур*. Фреймовые языки удобно использовать при разработке модели ПО по предмету *обучения* в тренирующих и *автоматизированных обучающих системах* (АОС). Кроме того, фреймовые языки применяются при организации естественного языкового общения между *обучаемым* и АОС.

Широкое распространение получил новый подход к программированию — объектно-ориентированное программирование. В рамках этого подхода программа состоит из взаимосвязанных объектов, обменивающихся *запросами*; каждый объект является экземпляром некоторого класса, определяющего правила реакции на различные запросы. В системе программирования *Смолток-80* объектно-ориентированный подход получил наиболее полное и последовательное воплощение.

Ныне для представления знаний все чаще используются средства логического программирования, в частности язык Пролог. В основе логического программирования лежит идея описания задачи совокупностью утверждений в некотором формальном логическом языке и получение решения построением вывода в некоторой формальной дедуктивной системе. Классы формул, используемые для описания задач, методы определения их семантики, модели вычислений, основанные на тех или иных системах вывода, весьма разнообразны. Несложность расширения и модификации программ, написанных на Прологе, легкость их восприятия и доступность выразительных средств делают этот язык удобным для подготовки прототипов различных небольших систем и использования для создания *баз знаний*. Аксиомы Пролога легко воплощают декларативные знания благодаря удобному синтаксису и семантике предложений языка. ПРОЛОГ применялся для создания систем, ориентированных на знания и используемых для решения задач диагностики, архитектурного проектирования, логического проектирования аппаратного обеспечения. Для *обучающих систем* удобно использовать этот язык для представления изучаемых объектов со сложной логической структурой или процессов, требующих значительного количества логических выводов, напр., для определения состояния объектов, а также реализации моделей, представленных в логике предикатов первого порядка и естественного языкового *интерфейса* между *обучаемым* и АОС, напр. *САКИО*. Кроме того, на Прологе часто реализуют отдельные блоки интеллектуальных систем, системы управления эвристическим поиском и организацией управления учебным процессом.

В. В. Колов.

ЯЗЫК ПРОГРАММИРОВАНИЯ (ЯП) — алгоритмический язык, предназначенный для реализации программного обеспечения ЭВМ. Относятся к искусственным языкам, в которых синтаксис и семантика строго определены. Для ЯП характерно наличие на компьютере его реализации в виде транслятора или интерпретатора.

По способу представления и обработки данных ЯП делятся на императивные языки (Язык описания курсов (ЯОК), Фортран, Паскаль, Ада, Смолток-80 и др.), функциональные языки (Лисп, Рефал и др.) и декларативные языки (Пролог и др.). По уровню проблемной ориентированности ЯП подразделяют на проблемно-ориентированные языки (ЯОК, Кобол и др.), универсальные языки (Си, Паскаль, ПЛ/1 и др.) и машинно-ориентированные языки (Ассемблера язык, Лисп для ЛИСП-машины, Смолток для СМОЛТОК-машины и др.).

В.А.Третьяк.

ЯЗЫК ПРОЕКТИРОВАНИЯ — алгоритмический язык, предназначенный для разработки программного обеспечения. Имеет много общего с языком программирования, но больше ориентирован на структуру и содержание программ. Я.п., не рассчитанные на создание исполняемых программ, наз. псевдоязыками. Обычно формальный синтаксис Я.п. покрывает задачи определения данных и общей структуры программы. Разработано много разнообразных Я.п. программ. На практике выбирается тот из них, который наиболее соответствует выбранному языку программирования. В качестве Я.п. иногда используют языки спецификаций. По способу представления и обработки данных Я.п. делятся на императивные языки, функциональные языки и декларативные языки. По уровню проблемной ориентированности Я.п. подразделяют на проблемно-ориентированные языки, универсальные языки и машинно-ориентированные языки.

В.А.Третьяк.

ЯЗЫК СПЕЦИФИКАЦИЙ — алгоритмический язык, предназначенный для формального описания задачи в виде набора спецификаций и ориентированный на точное определение ее семантики. Исходные требования часто наз. внешней спецификацией задачи (они, как правило, задаются для внешнего пользователя — заказчика), а промежуточные требования (ориентированные на разработчика программы) — внутренней спецификацией. Если программирование решения задачи осуществляется на языке высокого уровня (Паскаль, Ада и т.п.), то Я.п. могут использоваться как

языки проектирования. Если Я.п. реализуются на компьютере, то они могут использоваться и как *языки программирования.*

По способу представления и обработки данных Я.с. подразделяются на *императивные языки, функциональные языки и декларативные языки,* по уровню проблемной ориентированности — на *проблемно-ориентированные языки, универсальные языки и машинно-ориентированные языки.*

В.А.Третьяк.

ЯЗЫКОВЫЙ ПРОЦЕССОР — программа, обрабатывающая тексты, составленные на некотором алгоритмическом языке. Имеет дело с одним или несколькими формальными языками. К Я.п. относятся компоненты программного обеспечения типа: *макропроцессор, синтаксически-ориентированный редактор, синтаксический анализатор, транслятор, интерпретатор, смешанный вычислитель, отладчик, система доказательства правильности программ* и др.

В.А.Петрушин.

ЯРКОСТЬ — интенсивность светового потока, падающего на единицу поверхности (в обозреваемом направлении). В эргономике Я. связывают с реальным ощущением света, полученным глазом. Численные значения Я. определяют по формуле

$$L = dE_v / d\omega \cos\theta,$$

где dE_v — освещенность малого участка поверхности в пределах малого телесного угла $d\omega$, θ — угол между рассматриваемым направлением и нормалью к участку поверхности. Официальная международная единица измерения Я. — кандела на квадратный метр ($\text{кд}/\text{м}^2$).

С.А.Миронченко, В.С.Пашковский.

КОМПЬЮТЕРНАЯ ТЕХНОЛОГИЯ ОБУЧЕНИЯ В СТРАНАХ ВОСТОЧНОЙ ЕВРОПЫ, В БЫВШЕМ СССР, НА КУБЕ

БОЛГАРИЯ. Работы по автоматизации процесса обучения в Болгарии относятся к началу 60-х гг., когда развернулось создание и внедрение сначала в вуз, а затем и в среднюю школу обучающих машин, разработанных на основе методологии программированного обучения, аудиовизуальных и технических средств обучения (ТСО). Первые ТСО созданы в Высшем машинно-электротехническом институте Софии под руководством проф. П.Дановского — основателя и руководителя первой в Болгарии проблемной научно-технической лаборатории ТСО, организованной в 1968. Затем был построен и введен в эксплуатацию завод имени М.В.Ломоносова в Плевене по выпуску ТСО.

Процесс внедрения ТСО в среднюю школу и вуз сопровождался комплексным применением аудиовизуальных средств (шрифтпроекторов, телевизионных систем, кинопроекторов, аппаратуры для озвучивания и др.), созданием групповых средств контроля знаний (т.наз. систем обратной связи) на лекциях и семинарских занятиях, конструированием индивидуальных контролирующих и обучающих машин типа МАК (машина для автоматизированного контроля), РЭМС (репетитор — экзаменатор Москва — София) и др. В дальнейшем началась разработка и применение сценарных автоматизированных обучающих и контролирующих систем на базе ЭВМ, напр. АОС-НРБ, в основу которых были положены аналогичные разработки в США (КУРСАЙТЕР) СССР (СПОК, СПОК-ВУЗ). Параллельно с внедрением АОС и ТСО в начале 70-х гг. разрабатывались теоретические основы компьютерной технологии обучения. В 1980 вышла книга «Кибернетика и педагогика» под редакцией проф. И.Марева. Базой развития и внедрения компьютеров в учебный процесс стал центр по инженерной педагогике в Высшем машинно-электротехническом институте (ныне Техническом университете) в Софии. Осуществление этих работ происходило в сравнительно благоприятных условиях, поскольку уже свыше 30 лет в стране выпускалась своя вычислительная техника (аналоги IBM-360 и РДР-11). С 80-х гг. начался серийный выпуск ПЭВМ серии «Правец-8» (аналог Apple-2) и на этой основе сложились предпосылки к массовому применению компьютерной техники в учебно-воспитательном процессе средней школы и вуза. Первый школьный компьютерный кабинет на базе этих ПЭВМ открылся (1982) в математической гимназии в Пловдиве.

Наряду с классическими психолого-педагогическими теориями методология КТО развивается в тесной связи с онтодидактикой (предполагает деятельностный подход к процессу обучения, органически сочетающий различные формы учебной, научной и практической работы обучаемых), изучающей по возможности все механизмы и средства ТСО, печатные дидактические материалы, с помощью которых новейшие знания отдельных наук трансформируются в содержание учебных предметов. Онтодидактический подход дает возможность оптимально представить современное научное знание в доступной для учащихся форме и преобразить его в средство развития их познавательных способностей.

В условиях организованного обучения работу на микрокомпьютере осваивают в 10-х и 11-х классах единой средней политехнической школы, на 2-м и 3-м курсах техникумов и средних профессиональных училищ, где введен новый общеобразова-

тельный предмет "Информатика". Некоторые элементы информатики введены в учебное содержание по математике в 8-х классах школ. В специализированных математических школах (существуют более 30 лет) с начала 70-х гг. изучаются такие языки, как Алгол, Фортран IV и учебный язык программирования Тутак, с 1983 — Бейсик и ПЛ/1. Это дало возможность подготовить учащихся в таких современных областях науки и техники, как компьютерная техника, робототехника, лазерная техника, оптоэлектроника и др. С 1988 в ряде средних школ введено экспериментальное обучение по учебной робототехнике. Оно направлено на усвоение практических умений и навыков для работы с учебными роботами и гибкими автоматизированными производствами, некоторых основных принципов конструирования роботов и роботизированных линий, способов применения компьютерных технологий для управления роботизированными системами разных типов.

В Болгарии выделяют три уровня компьютерной грамотности. Первый уровень ориентирован на подготовку пользователей, что составляет 80—90% общего количества студентов страны, изучающих естественные, технические, экономические, сельскохозяйственные, медицинские и др. науки. Второй уровень охватывает 10—20% студентов и предусматривает подготовку разработчиков прикладных средств информатики для определенных предметных областей. Третий уровень ориентирован на подготовку специалистов в области самой информатики и прежде всего — в области ее инструментальных средств.

КТО в Болгарии направлена в основном на обеспечение доступности компьютерных знаний на основе поддержки самостоятельной учебной деятельности обучаемых в определенной предметной области, повышение эффективности взаимодействия обучаемого и обучающего путем организации и осуществления педагогического управления. При этом КТО предусматривает: создание альтернативы книге (учебнику) на основе построения баз знаний учебного назначения; обеспечение перехода от информационно-поискового к активному обучению; выработку умений самостоятельного поиска знаний; требование применять новые подходы при структурировании содержания обучения; простоту изменений в методах и организационных формах обучения; создание условий для уменьшения информационной перегруженности учащихся и т.д.

Получает распространение видеокомпьютерная технология обучения, ведутся интенсивные исследования по применению средств искусственного интеллекта в учебных целях. Хорошие практические результаты достигнуты в создании и внедрении учебных роботов и гибких автоматизированных производств. Много сделано в теории и практике учебных игр, имитационных систем, а также в разработке санитарно-гигиенических и эргономических норм КТО. К перспективным техническим средствам КТО относятся: 32-разрядная супермикросистема FADATA-20, ориентированная на создание моделирующих (имитационных) систем и поддерживающая до 32 рабочих мест пользователей/обучаемых с двумя служебными рабочими местами, которые могут использоваться как рабочее место преподавателя; учебно-ориентированные локальные сети (напр., MICRONET-16) и др.

Разработка программных средств КТО идет по пути усложнения и интеграции или специализации поддерживаемых ими дидактических функций. Создаются программно-технические комплексы для проведения автоматизированных лабораторных работ и экспериментов (типа Микролаб), обучающие (Машинознание) и учебные (Система автоматизации инженерного труда) среды, интегрированные программы и системы учебного назначения (напр., диалоговая информационно-обучающая система, авторские системы для преподавания технических дисциплин) и др.

Дальнейшее развитие КТО в Болгарской Республике связано с созданием и применением экспертных и экспертных обучающих систем.

Лит.:

1. Атанасов К. Компьютерно обучение — проблемы, перспективы // Обучението по математика. — 1984. — № 2.
2. Бойко А. ПЭВМ в школе // Наука и жизнь. — 1985. — № 10.

3. Георгиев Г., Димитрова Т., Караманека Д. Эргономични фактори на компютризирано обучение. София: ВМЕИ "Ленин", 1988.
4. Дановски П., Довгялло А.М., Кирова К.Н., Панков И.П., Райко Д.М. Автоматизированные обучающие системы на базе СПОК // Современная высшая школа. —1983. —№ 1.
5. Larsen S. Computers in Education: A Critical View. В кн.: Children in an Information Age: Tomorrow's Problems Today. New York., 1986.
6. Ляудис В.Я. Психологические принципы конструирования диалоговых обучающих программ в ситуации компьютерного обучения. В кн.: Психолого-педагогические и психофизиологические проблемы компьютерного обучения: М., 1985.
7. Машбиц Е.И. Психолого-педагогические аспекты компьютеризации // Вестник высшей школы. —1986. —№ 4.
8. Pisarev A. The National Programme of the People's Republic of Bulgaria for the Introduction of Computers in Secondary Schools. В кн.: Children in an Information Age: Tomorrow's Problems Today. New York, 1986.
9. Papert S. Mindstorms: Children, Computer and Powerful Ideas. New York, 1980.
10. Sendov B. The Mathematical Computer Laboratory: Still Optimistic after Two Years' Experience. В кн.: Children in an Information Age. Preprints of the Third International Conference, Sofia, 1989.
11. Спиоров К. Индивидуализация обучения с помощью компьютеров. В кн.: Современные образовательные технологии. М., 1989.
12. Талызина Н.Ф. Технология обучения и ее место в педагогической теории // Современная высшая школа. —1977. —№ 1 (17).
13. Тихомиров О.К., Бабанин Л.Н. ЭВМ и новые проблемы психологии. М., 1986.
14. Цонева В., Павлов Д. Основные подходы в разработването и използването на компютърни програми за обучение // Психология. —1986. —№ 5.

К. Атанасов, А.Писарев, К.Спиоров, В.Цонева.

ЧЕХО-СЛОВАКИЯ. В 1981 было принято решение ускорить развитие микроэлектроники, разработать и планомерно реализовать долгосрочную программу электронизации всего народного хозяйства, в связи с чем возросли требования к системе подготовки кадров. Программа электронизации учебно-воспитательного процесса включает:

а) изучение электроники как научной дисциплины и учебной специальности в рамках самостоятельных курсов или в составе других предметов (изучение электронных деталей и схем, разработка, производство и проверка деталей, схем, сложных электронных систем);

б) изучение вычислительной техники и ее применение в рамках самостоятельных курсов или в составе других предметов (не только разработка и наладка ЭВМ, но и алгоритмизация задач, программирование и решение задач на компьютерах в различных областях деятельности, включая компьютерное управление технологическими процессами);

в) применение компьютеров и средств электроники непосредственно в учебно-воспитательном процессе, т.е. использование их для самостоятельной познавательной деятельности обучаемых, контроля знаний, обучения и тренировки.

Исходя из этих основных направлений, были сформулированы конкретные задачи для различных уровней системы образования Чехо-Словакии. Так, главная цель дошкольного воспитания — формирование у детей навыков контакта с электронными и вычислительными системами с помощью электронных игрушек, конструкторов и компьютерных игр. На первой ступени начальной или общеобразовательной школы — расширенный контакт учеников с электронными и вычислительными системами для выработки у них положительного отношения к технике, навыков использования ЭВМ в повседневной жизни. На второй ступени начальной школы ученики должны научиться пользоваться калькуляторами для решения простых задач, приобрести знания по основам алгоритмизации.

В средних школах электронизация учебно-воспитательного процесса развивается дифференцированно, в зависимости от типа школы. Основные цели электронизации в гимназиях — овладение физическими основами электроники и электронных систем, усвоение теоретической сущности работы вычислительных систем и основ их практического применения, включая алгоритмизацию задач и взаимодействие с ЭВМ. В средних профессионально-технических школах учащиеся должны освоить электронику и вычислительную технику на таком уровне, чтобы использовать затем электронные и вычислительные системы в своей профессиональной деятельности. В средних профессионально-технических училищах изучение вычислительной техники вводится для того, чтобы каждый учащийся знал основные функции компьютера, понимал электрические и электронные схемы, умел составлять простые программы для персональной ЭВМ, пользоваться микрокомпьютером в будущей профессиональной деятельности и в повседневной жизни, имел представление о применении вычислительной техники в народном хозяйстве.

В вузах методы и средства обучения должны обеспечивать прямой контакт обучаемых со средствами электроники и вычислительной техники, развивать активность обучаемых и самостоятельность при изучении новых предметов.

Выполнение программы электронизации зависит от оснащенности школ электронными конструкторами, электронными системами, калькуляторами, микрокомпьютерами (персональными и управляющими), учебными классами с микрокомпьютерными локальными сетями, дисплейными учебными классами и современной дидактической техникой, а также от качества подготовки и степени повышения квалификации преподавателей (к основной переподготовке привлекаются преподаватели школ всех типов). Повышение квалификации преподавателей вузов также осуществляется дифференцированно, в зависимости от типа вуза и специальности. Для этой цели широко используются собственные силы и средства вузов.

Реализация программы электронизации в учебно-воспитательном процессе предусматривает не только организованное обучение в школах, но и внеклассную работу учащихся, напр. в Чехословацком научно-техническом обществе и Добровольном обществе содействия армии. Для внеклассной работы учащихся в любительских кружках школ, научно-технического творчества учащихся средних и высших учебных заведений привлекается оборудование школ. Чтобы избежать дробления сил и средств, деятельность инструкторов, лекторов и руководителей, а также использование оборудования координируются.

Из-за недостатка учебных программ в школах рекомендуется заменить стандартные телевизоры мониторами, унифицировать типы используемых микро-ЭВМ, разработать проект учебного компьютерного класса и лаборатории электротехники (такая разработка осуществлена в Северо-Моравской области), подготовить единую методику по составлению программ. Особое внимание уделяется подготовке преподавателей информатики и вычислительной техники. Ныне в школах, как правило, есть преподаватели — так наз. гаранты по электронизации, которые работают под руководством заведующих кабинетов электронизации областных педагогических институтов или районных педцентров. Областной педагогический институт в Брно издает справочник, содержащий информацию по электронизации и обмену опытом реализации долгосрочной комплексной программы электронизации в области просвещения Южно-Моравской области. Процесс электронизации наиболее интенсивно происходит в высших технических учебных заведениях и на факультетах вузов, осуществляющих подготовку преподавателей. В вузах используются 16-битовые персональные компьютеры типа PC, совместимые с IBM PC, создаются учебные классы с графическими системами для конструкторских работ.

Чтобы создать теоретические предпосылки для широкого применения электроники в народном образовании, была разработана научно-исследовательская программа "Электронизация в воспитании и образовании". Цели этой программы: создание типовых лабораторий и учебных компьютерных классов с локальной сетью для основных, средних и высших учебных заведений; подготовка методических материалов для преподавателей. Дальнейшее развитие программы электронизации обучения связано с информатизацией образования. Среди программ учебного назначения

преобладают программы контроля знаний, игры. Актуальны проблемы использования сетей ЭВМ, включая большие (национальные и интернациональные) информационно-справочные системы. Задача овладения элементарной компьютерной грамотностью распространяется на все подрастающее поколение. Предусматривается переход от языков программирования к проблемно-ориентированному пакету прикладных программ, системам управления базами данных, графическим редакторам, средствам обработки знаний. Предполагается радикальное изменение содержания обучения, что повлечет за собой и изменение методов обучения.

З.Цага.

ВЕНГРИЯ. Применение ЭВМ в обучении началось в конце 60-х гг. и осуществлялось по двум направлениям: изучение основ компьютерной грамотности, повышение квалификации специалистов по информатике и вычислительной технике.

Основы компьютерной грамотности изучаются в начальной школе — общеобразовательной школе для учащихся от 6 до 14 лет; средней школе — общеобразовательной гимназии для учащихся от 14 до 18 лет; техникуме, профтехучилище — учебном заведении для учащихся от 14 до 17 лет; в университетах и колледжах. В программу венгерских средних учебных заведений не входит такой предмет, как «Вычислительная техника», поэтому в некоторых ПТУ и техникумах, специализирующихся на подготовке специалистов по вычислительной технике, изучение данного предмета включено в рамки таких, напр., дисциплин, как техника, математика, физика, а также осуществляется в кружках, иногда на факультативных уроках. Для наиболее способных учеников ежегодно организуется конкурс, лауреаты которого (10 человек) получают право поступать в вуз без вступительных экзаменов. Изучая основы компьютерной грамотности, учащиеся школ, ПТУ, техникумов знакомятся с основами информатики, программирования, электроники; совершенствуют алгоритмическое мышление, способности решать задачи; изучают области применения вычислительной техники. В университетах и колледжах готовят специалистов по обслуживанию вычислительной техники, преподавателей по вычислительной технике и программированию.

В 1965 был основан учебный центр по информатике и вычислительной технике САМОК (с 1981 является частью предприятия по применению и внедрению вычислительной техники САМАЛК) для подготовки разработчиков информационно-справочных систем, программистов высокой квалификации, руководителей вычислительных центров, а также повышения квалификации специалистов, работающих в этих центрах. С 70-х гг. организуются и международные курсы. С начала 80-х гг. функционируют курсы для пользователей технического и программного обеспечения ЭВМ. Обучение на курсах по подготовке кадров проводится, как правило, в заочной форме, а на курсах повышения квалификации — стационарно, с отрывом от производства, в течение пяти дней. По окончании курсов слушатели пишут и защищают дипломную работу и сдают экзамен. Слушатели, успешно закончившие курсы, получают диплом, приравненный к диплому о высшем профессиональном образовании. В 1986 САМОК открыл курсы с дневной формой обучения по подготовке специалистов в области вычислительной техники для выпускников средних школ, не поступивших в вуз. Они учатся в течение двух лет и получают аттестат оператора ЭВМ, кодировщика, младшего программиста, оператора информационно-справочных систем. С 1989 обучение на курсах проводится также на английском, немецком, французском языках. Такие курсы организованы не только в Будапеште, но и в других больших городах. В Венгрии существует много любительских кружков, общества по пропаганде, обучению и внедрению средств информатики. В 1980 Общество по вычислительной технике имени Дж. Неймана разработало национальную программу информатизации венгерского общества, включающую программу изучения основ компьютерной грамотности, организацию массового изучения вычислительной техники, поддержку любительских кружков по изучению информатики. Обществу имени Дж. Неймана принадлежит идея внедрения технологии обучения на расстоянии. Первой

попыткой реализации такого обучения был телевизионный курс "Обучение Бейсику" для университетов и колледжей, затем курс "Обучение для предпринимателей". Общество имени Дж.Неймана организует частные и государственные клубы по изучению компьютеров, а также открывает различные курсы по вычислительной технике, устраивает выставки по истории компьютеров, летние лагеря по вычислительной технике. В Венгрии несколько сот тысяч персональных компьютеров находятся у населения. В связи с этим возникла проблема обучения населения применению ЭВМ на профессиональном уровне, которая успешно разрешается в первую очередь в кружках при школах. Созданы различные курсы в обществах распространения научных знаний, в домах культуры, при небольших предприятиях, а также частными лицами.

В школах получили распространение венгерские компьютеры HT-1080 (на базе TRS-80), TV компьютер (Videoton), американский компьютер Commodore модификаций C-64, C-16 и др. Еще в конце 60-х гг. в вузах были созданы вычислительные центры для коллективного пользования главным образом на базе ЕС ЭВМ с целью приобретения специальности в области вычислительной техники, оказания вычислительных услуг, разработки программ, в т. ч. и обучающих (примером может служить система автоматизированного производства Chemius для химической промышленности). Планировалось создать сеть из этих центров, однако с появлением доступных для любого учреждения мини- и микрокомпьютеров необходимость в ней отпала. До 1980 большую часть компьютеров вузы приобретали за счет государственного бюджета, а с 1980 — в основном благодаря собственным денежным ресурсам.

В области программного обеспечения венгерские разработки всегда находили спрос на мировом рынке. Наиболее результативными и эффективными были инструментальные средства для разработки прикладных программ (хотя авторских систем для разработки учебных курсов все еще недостаточно), системы CAD/CAM, языки программирования высокого уровня, разработки в области искусственного интеллекта, системы баз данных и др. Внедрение КТО в образование — сложная проблема, поэтому необходимо, чтобы небольшие государства кооперировались для ее решения. Исходя из этого, учебный центр САМОК начал контактировать с различными зарубежными университетами и колледжами, с руководством ЮНЕСКО для организации международного колледжа по обучению вычислительной технике с использованием технологии обучения на расстоянии. В таком колледже можно было бы изучать вопросы применения средств вычислительной техники, получать диплом более высокого уровня (магистра, доктора) при условии, если учащийся успешно сдаст экзамены по различным дополнительным предметам.

А.Бардош, А.Киш, Д.Ковач

ГЕРМАНИЯ (восточные земли). Разработка отдельных элементов компьютерной технологии обучения и внедрение их в практику осуществляется с 1985. Особое внимание при этом уделяется теоретическим вопросам применения вычислительной техники в образовании, созданию научных основ разработки технологических приемов и построения эффективных педагогических процессов с помощью компьютера. Одним из результатов исследований в этой области, проводимых в Дрезденском университете, явилась классификация функций компьютера в учебной деятельности, на основании которой можно судить об эффективности использования его в процессе обучения.

Моделью процесса обучения служит процесс взаимодействия субъекта (обучаемого) с предметами (объектами, процессами) и /или средствами (инструментами, машинами) с целью приобретения знаний, умений, навыков. Получение материальных ценностей при этом — лишь средство для достижения цели. Первый вариант такой модели — совместная или кооперативная деятельность нескольких обучаемых, главная цель которой — формирование способностей к разделению труда, к руководству кооперативными процессами, к коммуникации. Второй вариант — взаимодействие обучаемого с моделями реальных объектов, процессов и др. для получения знаний об оригиналах или навыков работы с ними. Помимо предметов и средств деятельности

сама деятельность также является предметом изучения. При этом обучаемый должен усвоить приемы, методы, правила решения задач и оценки результатов. Различные виды деятельности связаны в реальном процессе обучения, их последовательность зависит от целей обучения, методического плана и т.д. Основная задача преподавателя заключается в том, чтобы создавать необходимые условия деятельности обучаемых, стимулировать, ориентировать, оценивать и управлять ею. Таким образом, в учебной деятельности можно выделить следующие функции компьютера:

- а) предмета — компьютер как часть реальности сам является предметом усвоения;
- б) рабочего средства — компьютер используется для накопления и обработки данных, для расчетов, поиска информации, управления, оптимизации;
- в) моделирования — компьютер представляет модель предмета, напр. экономического процесса, структуры или функций машины, поведения организмов;
- г) отображения — компьютер передает отображения изучаемого предмета в языковом, графическом или численном виде;
- д) управления — с помощью компьютера ориентируется, управляется, оценивается познавательная деятельность обучаемых;
- е) организации — компьютер выступает как рабочее средство преподавателя, напр. при анализе успеваемости, генерации задач, разработке программ учебного назначения;
- ж) руководства и планирования — компьютер поддерживает планирование, управление и оптимизацию учебной деятельности в целом.

Выбор и сочетание различных функций компьютера определяется дидактическими целями и характером решаемых задач. Обучаемый может самостоятельно использовать компьютер в функциях б, в, г и д. Целесообразно предусмотреть возможность сочетания нескольких функций компьютера в процессе обучения, напр. при использовании его в качестве рабочего средства (б) для расчета конструкций моста обучаемому может быть представлена информация о методах вычисления (г), а также оказана помощь в применении конкретного метода (д). В том случае, когда основной целью взаимодействия обучаемого с компьютером служит приобретение знаний, функция управления (д) может преобладать. Программа представляет обучаемому необходимую информацию (г), поддерживает работу с имитационными моделями (в), осуществляет проведение вычислений, поиск информации (б) и т.п.

В общеобразовательных школах проводится поэтапное введение элементов информатики в рамки других учебных предметов (математики, физики, трудового обучения), факультативных курсов и специального предмета в зависимости от типа школы (специальные школы математического, естественнонаучного и технического профиля, 12- и 10-летние школы). Обязательное изучение курса информатики введено в специальных школах с 1985, в 12-летних — с 1990. В 10- и 12-летних школах, оснащенных компьютерами, введены два факультативных курса по информатике. В рамках трудового обучения учащиеся ряда школ знакомятся с техническими аспектами информатики в компьютерных кабинетах школ или политехнических центров.

В системе профессионального образования разрабатываются новые учебные планы, элементы КТО в которых имеют различное значение в зависимости от профиля будущей профессии. Обучение проводится в компьютерных кабинетах профтехучилищ и на специальных тренажерах на предприятиях с использованием специального программного обеспечения для учебных целей (напр., для имитации и тренировки), а также программного обеспечения, применяемого на будущем рабочем месте.

В высшей школе компьютер рассматривается как решающее средство для новой качественной подготовки студентов. Особое внимание при этом должно уделяться: расширению активных видов усвоения, использованию имитации, моделированию изучаемых объектов, проведению расчетов, оптимизации поиска информации без прямого участия преподавателя; расширению возможностей самостоятельного доступа обучаемого к актуальному фонду знаний вуза, общества, человечества (банки данных, методов, знаний); помощи преподавателю в определении и структурировании целей и содержания обучения; методической организации учебного процесса (обучающие и авторские системы, экспертные системы учебного назначения).

Развивается разработка программного обеспечения для обучения, в т.ч. автоматизированных деловых игр в области экономических наук, тренажеров для работы с обрабатывающими и управляющими устройствами, различных моделирующих программ для практических и лабораторных занятий по физике, химии и автоматизированному производству, а также обучающе-контролирующих программ. Создана модульная система методических указаний по разработке обучающе-контролирующих программ. Ведутся исследования в области разработки теоретических основ КТО (в частности, диагностики и моделирования процессов учения и обучения), а также в области конструирования баз знаний и использования методов искусственного интеллекта.

Общедоступной стала возможность самостоятельного повышения квалификации в области информатики и вычислительной техники без отрыва от производства. Созданы и функционируют курсы повышения квалификации по телевидению и радио, многочисленные компьютерные клубы, вечерние школы, где изучают компьютеры, а также академии повышения квалификации на предприятиях. Определенный вклад в распространение знаний в области информатики вносят различные общественные организации.

Подготовка и переподготовка преподавателей информатики проводится на курсах повышения квалификации (для выпускников вузов), на вечерних курсах и интенсивных курсах во время каникул (для студентов). Предполагается обеспечить последовательное применение элементов КТО в школах, профтехучилищах, вузах по единой сквозной концепции.

Лит.:

1. Inbe W., Hartwig W.-H. Computergestützte Vermittlung und Aneignung (Anleitung für Programmautoren). —Т.1—2. Berlin, 1988.

2. Schneider G. Stand und Entwicklung von Informatikbildung als Bestandteil der Allgemeinbildung // Pädagogik. —1986. —43, 6.

В. Ибе, В.-Х. Хартвиг.

ПОЛЬША. Компьютеризация обучения в вузах началась в 60-х гг. В начале 70-х гг. в общеобразовательных лицеях, в некоторых профтехучилищах и в одной из средних школ Варшавы был введен предмет "Информатика". В связи с низкой степенью насыщенности рынка вычислительным оборудованием, его высокой стоимостью и большими габаритами лаборатории информатики не были оснащены компьютерами. Занятия проводились в вычислительных центрах, где обучаемые знакомились с компьютерами, программированием. В 1974 курс информатики был введен в институте подготовки учителей. В 1985 организован Консультационный комитет по информатике. Быстро увеличивается количество персональных ЭВМ, приобретаемых за счет местных ресурсов. Организуются студенческие микрокомпьютерные клубы. Персональные ЭВМ начинают использовать для обучения не только информатике, но и другими предметам. С 1986—87 информатика вводится как факультативный предмет в средних школах. Особое внимание уделяется: внедрению КТО, другим новым информационным технологиям в процесс обучения; обеспечению учебных заведений компьютерами, соответствующими целям и задачам образования; разработке программ учебного назначения и обеспечению реализации в этих программах лучших методов обучения; подготовке преподавателей в области КТО и другими информационными технологиями в образовании.

Компьютерная грамотность в Польше предусматривает общую и профессиональную грамотность. Общая грамотность — это знания об архитектуре ЭВМ, ориентация в возможностях компьютеров, а также знания о базовом программном обеспечении. Обучаемый должен получить знания, которые дают возможность запуска готового программного обеспечения, и таким образом использовать возможности компьютера. Профессиональная грамотность тесно связана с профессиональной деятельностью и

включает знания о таких основных операциях обработки данных, как накопление данных, поиск, сортировка и т.д. Основной для изучения вопросов общей технологии работы с компьютером служат распространенные в Польше текстовые редакторы, базы данных и др. В профессиональной компьютерной грамотности выделяют три подуровня подготовки. Первый подуровень охватывает основные знания об аппаратуре ЭВМ и программном обеспечении. Обучаемый знакомится с устройством компьютера и важнейшими типами внешних устройств, получает знания о применениях ЭВМ, приобретает умения использования готовых программ и основных возможностей операционной системы. После проработки материала обучаемый должен уметь свободно пользоваться микрокомпьютером, оценивать полезность микрокомпьютерных устройств и понимать их описания. Второй подуровень охватывает изучение процесса обработки данных на примерах таких практических систем, как текстовые редакторы, базы данных, электронные таблицы. Обучаемый также знакомится с основами применения компьютерной графики, сведениями об алгоритмах и программировании (на проблемно-ориентированных языках). На третьем подуровне обучаемый должен научиться использовать в рамках профессиональных предметов соответствующие компьютерные системы и пакеты прикладных программ, напр: настольную издательскую систему, средства проектирования печатных плат, элементов машин, образцов тканей, средства программирования работы станков с числовым программным управлением. На первом и втором подуровнях профессиональной грамотности предусмотрена возможность факультативных занятий с теми обучаемыми, которые желают расширить свои знания и умения. Цель КТО в средней школе — создание условий, позволяющих учащимся овладеть методами и средствами современной информатики в такой степени, чтобы в дальнейшей профессиональной деятельности компьютер стал для них обычным производственным инструментом. Во время занятий учащиеся должны ознакомиться с основами программирования и приобрести практические навыки в работе со школьным вычислительным оборудованием и программным обеспечением.

В школах применяются компьютеры фирмы Spectrum, совместимые с ZX/Spectrum (напр., Timex, Unipolbit) и IBM PC/XT, польский компьютер Meritum нескольких модификаций (выпуск начат в 1984), а также компьютеры Amstrad, MСХ, Atari. Вузы в связи с проведением научно-исследовательских и других работ наиболее часто используют компьютеры, совместимые с IBM PC/XT/AT, из польских — компьютер Мазовия.

Программное обеспечение КТО в ПР находится на начальном этапе развития. Если общее программное обеспечение и некоторые инструментальные программы учебного назначения (ПУН) поставляются организациями, выпускающими вычислительную технику, то разработки прикладных ПУН пока еще находятся на стадии экспериментов и любительских работ. Одна их важнейших работ — систематизация и определение типа прикладных ПУН (в ПР их называют дидактическими программами) в зависимости от того, кто является основным пользователем программы: учитель, обучаемый под руководством преподавателя, или самостоятельно работающий обучаемый.

Для изготовления прикладных ПУН применяются штатные языки программирования Фортран, Бейсик, Паскаль, Турбо-Паскаль, а также стандартные пакеты прикладных программ типа dBASEIII, Word star. Специализированные инструментальные средства — авторские языки и системы более предпочтительны, но пока мало доступны. Большинство создаваемых программ соответствует лишь любительскому уровню либо представляет собой адаптацию более или менее удачных зарубежных программ. Имеется небольшое число полностью оригинальных разработок, выполненных на профессиональном оборудовании.

Несмотря на то, что количество компьютерной техники постепенно увеличивается и она становится более доступной, многие вопросы КТО находятся в фазе эксперимента. Перестройка системы просвещения в области компьютеризации требует очень серьезных изменений прежде всего в дидактике и методике обучения отдельным предметам с учетом возможности индивидуализации работы обучаемых.

Лит.:

1. Качор С. Сюпитус — развитие целей воспитания в профессиональных школах и внешкольных формах профессионального образования: Состояние и перспективы профессионального образования в Польше — ИКЗ. Варшава, 1987.
2. Informatics and Education. A first survey of the state of the art in 43 countries. Paris, 1986.
3. Pinchas Tamir — Current and Potential uses of Microcomputers in Science Education Journal of Computers in Mathematics and Science Teaching. Winter 1985/1986.
4. Сольва Я. Условия развития современного профессионального образования: Состояние и перспективы профессионального образования в Польше — ИКЗ. Варшава, 1987.

В.Бабченко, С.Бонкович-Ситтауэр, А.Рафф.

КУБА. Работы в области компьютерной технологии обучения были начаты в 1978 в университетах городов Санта-Клары и Гаваны. Направление этих работ — исследование и создание компьютерных средств обучения (управления обучением), включая начальные варианты авторских языков. Техническую базу составили ЕС ЭВМ и кубинские компьютеры CID-201-A, CID-201-B, плохо приспособленные к работе в диалоговом режиме. Программное обеспечение: расширенный компилятор с Бейсика, автоматизированные обучающие системы INSTRUCTOR и CED в операционной системе ФОБОС. Указанные АОС обладали авторскими языками с возможностями для программирования и выполнения диалоговых обучающих программ, редактирования текстов и статистического контроля результатов работы этих программ. Кроме того, системы INSTRUCTOR и CED обеспечивали выдачу последовательности экранных страниц и анализ ответов обучаемых с помощью эталонов, используемых в традиционных АОС КУРСАЙТЕР, ТЬЮТОР и др. Исследование проводили в основном математики и инженеры из преподавательского состава вузов, пользующиеся компьютерами и техническими средствами обучения.

Внедрение в вузах микрокомпьютеров ускорило развитие обучающих систем. Уже в 1981 — 82 были приобретены и внедрены 8- и 16-битовые микрокомпьютеры с графическими возможностями и операционными системами CP/M и MS/DOS. Это позволило реализовать на новом технологическом уровне ранее разработанные АОС (напр., MicroCED) и одновременно развить новые виды обучающих программ, учебных игр, системы проектирования и имитационного моделирования. Были внедрены в эксплуатацию системы MicroCED, Suestionario и Test. Первая из них — расширенная версия системы CED, которая и ныне применяется при изучении различных предметов. Две другие представляют собой автоматизированные генераторы экзаменационных вопросов. В эти же годы предприняты первые попытки планирования и интеграции работ в области КТО. Министерством высшего образования разработана программа развития вычислительной техники, направленная на достижение следующих целей: приобретение студентами необходимых знаний, умений и навыков для использования компьютеров в своей профессиональной деятельности; рациональное применение научно-педагогическими и исследовательскими кадрами вузов вычислительной техники в научных исследованиях и в преподавании; повышение эффективности и производительности труда управленческих кадров, технического и административного персонала; развитие тематики научных исследований в области вычислительной техники на уровне мировых стандартов; совершенствование системы управления высшим образованием путем автоматизации информационных процессов и процесса принятия решения.

В плане развития средств КТО приоритетными являются автоматизированные обучающие системы, основанные на методах искусственного интеллекта; локальные сети учебного назначения; учебные системы автоматизированного проектирования; системы обучения иностранным языкам; программное обеспечение для профессионального образования инженеров и экономистов.

Кроме указанных работ, Министерство высшего образования разработало программу развития вычислительной техники, в которой предусматривается как массовое внедрение микрокомпьютеров в школах, так и формирование и переподготовка преподавательского состава, включая обеспечение необходимыми программными и техническими средствами (импортными и отечественными).

В последние годы первоочередное внимание уделяется разработке теоретических основ, освоению и созданию инструментальных средств для интеллектуальных обучающих систем. Координацию работ осуществляет Гаванский университет. Создана методология таких систем, в частности определены характеристики авторских средств для формирования инженерных знаний. Разработаны методы преодоления ограничений традиционных сценарных АОС в части индивидуализации обучения, диагноза и исправления ошибок обучаемых; определены характеристики взаимодействия человека с машиной в рамках интеллектуальных АОС; исследуются проблемы общения с интеллектуальной АОС на ограниченном естественном языке; разрабатываются инструментальные учебно-технологические модули, предназначенные для создания систем обработки знаний и экспертно-тренирующих систем. К числу таких модулей, в частности для создания перспективных программных средств КТО, относятся: обобщенные средства программирования на Прологе; расширения ARITY/PROLOG для графической работы; модули проектирования и реализация базы данных о студентах; модули реализации обобщенных дидактических метаправил; изучение проблем развития элементов общения на естественном или близком ему языке; развитие процессора двухразмерной математической нотации; внедрение тренировки по вычислению производных; внедрение тренировки для создания навыков программирования.

М.Прието Мендес.

КОМПЬЮТЕРНАЯ ТЕХНОЛОГИЯ ОБУЧЕНИЯ В РЕСПУБЛИКАХ БЫВШЕГО СССР. С начала 60-х велись поиски способов модернизации системы образования. *Программированное обучение, стандартизированный контроль знаний, обучающие машины* — первые шаги в области автоматизации и электронизации учебного процесса с целью повышения качества обучения, интенсификации труда преподавателей, управления работой учебных заведений. К началу 60-х гг. относятся и первые попытки применения компьютеров в обучении. Практически одновременно в СССР и в США были разработаны и реализованы автоматизированные системы для обучения и контроля знаний: в СССР — система РАДОН, разработанная Киевским высшим инженерным радиотехническим училищем на базе многотерминальной ЭВМ "Днепр", созданный в Институте кибернетики АН Украины; в США — одностерминальная система PLATO совместной разработки Иллинойского университета и фирмы CDC и др.

Применение компьютеров в учебном процессе началось с 1962 на первой Всесоюзной конференции по программированному обучению (Киев) академик АН СССР и АН УССР В.М.Глушков поставил задачу разработки компьютерных методов и средств для автоматизации обучения. В то время компьютеры были громоздкими и дорогими устройствами. Единственный возможный способ взаимодействия с компьютером состоял в подготовке колоды или набора перфокарт, которые затем передавались на вычислительный центр, после чего через 1—2 дня пользователь получал распечатку (листинг) с ответом. Компьютеры, применяемые в учебных целях, оснащались специальными (зачастую кустарными) терминалами, включающими кнопку-цифровую панель или клавиатуру для ввода сообщений обучаемого и слайдпроекторы, цифровые индикаторные устройства и т.п. для вывода ответов компьютера. В конце 60-х ситуация несколько улучшилась благодаря появлению мини-ЭВМ и режима *разделения времени*, когда десятки пользователей, каждый со своего терминала — электрической пишущей машинки, телетайпа — получили возможность одновременно работать с одним относительно мощным компьютером. Были созданы автоматизированные обучающие системы первого поколения (ПЕДАГОГ, САДКО, КОНТАКТ).

В начале 70-х гг. разработаны первые технологические *автоматизированные обучающие системы*, базирующиеся на специальных инструментальных средствах — *пакетах прикладных программ* типа *КУРСРАЙТЕР, СПОК, СПОК-ВУЗ* и др. Появление микрокомпьютеров и персональных ЭВМ (ПЭВМ) в конце 70-х гг., их относительно низкая стоимость и быстрый рост их применения в промышленности, коммерческих операциях и административном управлении обусловили социальный заказ на внедрение компьютеров в учебных заведениях, а также подготовили постановку задачи о многоуровневом компьютерном образовании всех членов общества. К началу — середине 80-х г. относятся правительственные постановления о компьютеризации системы образования, что связано с дальнейшим развитием вычислительной техники. Тем не менее о некоторых результатах внедрения КТО в СССР можно говорить лишь в отношении вузов и отраслевых систем повышения квалификации.

Для создания необходимой технической базы КТО в вузах и техникумах предусматривается оборудовать *рабочие места* обучаемых терминалами или персональными компьютерами. Определение места поступающей техники в учебном процессе, выбор форм и методов ее использования, поиск и адаптация средств математического обеспечения, разработка методов оценки и повышения эффективности вычислительной техники — практически все вопросы КТО каждый педагогический коллектив должен сегодня решать самостоятельно, опираясь лишь на опыт подготовки специалистов по вычислительной технике. При этом имеющийся опыт и средства подготовки специалистов по вычислительной технике могут быть лишь ограничено использованы при подготовке специалистов другого профиля. Использование ЭВМ в качестве «программированного пособия» и средства проверки знаний, несмотря на очевидные преимущества перед безмашинным обучением, не удовлетворяет полностью потребности преподавания дисциплин общепрофессионального и специального циклов и не соответствует современным требованиям к качеству обучения.

Одним из важнейших этапов подготовки к осуществлению КТО является организация в каждом учебном заведении кабинетов информатики общего назначения, оборудованных современными средствами вычислительной техники и предназначенных для подготовки специалистов в области *информатики* и вычислительной техники (в этих кабинетах могут проводиться занятия и по др. учебным дисциплинам), а также специализированных кабинетов — лабораторий (уже есть в таких вузах как московские авиационный, инженерно-физический и энергетический институты, Ленинградский электротехнический институт) для изучения микропроцессорной техники, систем автоматизации проектирования, робототехнических систем. В вузах для таких кабинетов разработаны собственные учебные микро-ЭВМ нескольких типов, напр., «*НЕРИНГА*» (Клайпедский факультет Каунасского политехнического института). УМПК-80 и МП-589 (Московский институт электронной техники).

Массовое и эффективное внедрение средств вычислительной техники в учебный процесс возможно при удовлетворении требований повышения их надежности, сравнимости с др. образцами лабораторного и экспериментального оборудования по стоимости и габаритам, простоте перестройки с целью проблемной ориентации, простоте эксплуатации в условиях учебного заведения. Этим требованиям в значительной степени соответствуют ПЭВМ типа IBM PC, которыми и оснащаются в настоящее время вузы и институты повышения квалификации.

В начале 80-х гг. совершенствовались инструментальные программные средства КТО, с помощью которых создаются прикладные программы в виде обучающих и контролирующих курсов, системы автоматизации проектирования и расчета конкретных объектов и т.д. Они должны обеспечить общение *пользователя* с ЭВМ на естественном профессиональном языке *предметной области*. С развитием инструментальных пакетов предусматривается возможность создания интегрированных систем различной степени комплексирования, в состав которых входят типовые компоненты, развитый графический интерфейс, *системы управления базами данных*, пакеты статистического анализа и др. К наиболее распространенным инструментальным авторским средствам обучения относятся автоматизированные обучающие системы, развивающиеся на базе пакетов прикладных программ РАКУРС, АДОНИС.

Технология разработки программ учебного назначения быстро видоизменяется в направлении снижения ее трудоемкости и повышения качества конечного продукта. Работа с современными инструментальными средствами не требует от авторов знания языков программирования, предлагая ему разработать и ввести в ЭВМ сценарий обучения; ряд средств со встроенным набором сценариев обучения предполагают лишь ввод понятий конкретной предметной области. Дальнейшее развитие авторских средств связано с применением *экспертных систем в обучении*.

Наиболее перспективными средствами для изучения специальных дисциплин, а также для научно-исследовательской и конструкторской работы *обучаемых* являются *автоматизированная система научных исследований (АСНИ)* и *автоматизированная система проектирования (САПР)*.

Учебная нагрузка по мере компьютеризации учебных заведений переносится с лекционных занятий на практические, лабораторные, семинарские занятия и самоподготовку, причем в условиях дефицита машинного времени расписание занятий выходит за рамки 6 ч дневного времени. Лекция как форма обучения не исчезает, однако ее содержание быстро изменяется. Вследствие минимизации описания объектов и явлений, составляющих предметную область, лекция принимает все более установочный характер, служит средством мотивации, формой постановки научно-технических проблем и формирования научного мировоззрения. Изменение содержания труда преподавателей предопределяет новые требования к его подготовке. Наряду со знанием предметной области по специальности, преподаватель должен быть соответствующим образом подготовлен в области КТО и общей педагогики, владеть методами системного анализа содержания труда специалистов и др. Часть преподавателей должна получить расширенную подготовку в области КТО, необходимую для методической работы по созданию средств КТО. Это направление подготовки включает изучение технологии производства программ учебного назначения, создание баз знаний, баз данных, их актуализация, освоение инструментальных средств автора обучающих диалоговых программ и др. средств КТО в зависимости от задач методической работы.

Чтобы решить поставленные задачи по компьютеризации образования, необходимо прежде всего преодолеть существующий еще у части преподавательского состава психологический барьер при использовании вычислительной техники, осуществить пересмотр содержания учебных программ по различным дисциплинам для непрерывного использования ЭВМ в учебном процессе при подготовке специалистов, разработать программные и методические средства обучения на основе новой технологии в кабинетах информатики и вычислительной техники.

Развитие КТО предусматривает обеспечение компьютерной грамотности студентов технических вузов уже в 1990; к 1995 планируется обеспечить всеобщую компьютерную грамотность студентов, преподавателей и научных работников всей системы народного образования. К 2000 намечено осуществить практическое внедрение в народное образование новых информационных технологий, включая КТО.

А.М.Довгялло, Н.М.Когдов, А.Я.Савельев.

СПИСОК ЛИТЕРАТУРЫ

- Абов Н.В.* Чань-буддизм и культурно-психологические традиции в средневековом Китае. — Новосибирск, 1989.
- Абрамов В.В., Розенталь Ю.Д.* Внешние запоминающие устройства для персональных ЭВМ // ЭВМ массового применения. —М., 1987.
- Автоматизация управления вузом.* —М., 1984.
- Автоматизированные обучающие системы профессиональной подготовки операторов летательных аппаратов.* —М., 1986.
- Агафонов В.Н.* Спецификация программ: понятийные средства и их организация. — Новосибирск, 1987.
- Акинфеев А.Б., Мironцев В.И., Софинский Г.Д. и др.* Полупроводниковые запоминающие устройства и их применение. —М., 1981.
- Алексюк А.М., Чорний В.М.* Методи навчання в сучасній буржуазній педагогіці США. —К., 1983.
- Альтшуллер Г.С.* Найти идею. Введение в теорию решения изобретательских задач. — Новосибирск, 1986.
- Аткинсон Р., Бауэр Г., Кротерс Э.* Введение в математическую теорию обучения. —М., 1969.
- Бабанский Ю.К.* Теоретические вопросы оптимизации процесса обучения // Методика оптимального планирования процесса обучения. —М., 1976.
- Балл Г.А.* О понятиях “воздействие”, “действие” и “операция” // Вопр. психологии. —1974. —№ 4.
- Безбородов Ю.М.* Индивидуальная отладка программ. —М., 1982.
- Безруков Н.Н.* Классификация компьютерных вирусов MS-DOS и методы защиты от них. —М., 1990.
- Белая Н., Стилл Т.* Логика вопросов и ответов. —М., 1981.
- Белов В.Н., Брановицкий В.И., Вершинин К.П. и др.* ПРОЛОГ — язык логического программирования // Прикл. информатика. —1986. —В.1.
- Берг А.И.* Кибернетика — наука об оптимальном управлении. —М.—Л., 1964.
- Беспалько В.П.* Программированное обучение. Дидактические основы. —М., 1970.
- Боднер В.А., Закиров Р.А., Смирнова И.И.* Авиационные тренажеры. —М., 1978.
- Бондаровская В.М., Мартиросова В.Г., Навакитян А.О.* Влияние работы с видеотерминалами на зрительную систему пользователей // Измерения, контроль, автоматизация. —1989. —№ 3.
- Бондаровская В.М., Миронченко С.А., Появкель Н.И. и др.* Повышение потребительских свойств технических средств СМ ЭВМ в процессе их разработки // Приборы и системы упр. —1985. —№ 4.
- Бондаровская В.М., Миронченко С.А., Появкель Н.И.* Эргономический проект учебных ПЭВМ // Информатика и образование. —1988. —№ 6.
- Бондаровская В.М., Появкель Н.И.* Психология и программное обеспечение ЭВМ. Теория и практика. —Киев, 1988.

- Бондаровская В.М., Субботин Ю.А., Чачко С.А. Эргономические аспекты проектирования и эксплуатации видеотерминалов // Измерения, контроль, автоматизация. —1986. —В.2(58).
- Бондарчук О.І. Психологічні особливості застосування комп'ютерної гри у позакласній роботі з молодшими школярами // Психологія. —1989. —В.33.
- Бончев С. Еще раз о компьютерных вирусах // Компьютер за вас. —1989. —№ 3—4.
- Браун П.Дж. Макропроцессоры и мобильность программного обеспечения. —М., 1977.
- Браун П. Обзор макропроцессоров. —М., 1975.
- Брукс Ф. Как проектируются и создаются программные комплексы. —М., 1979.
- Брусенцова Т.Н. О психолого-педагогических принципах компьютерной системы обучения Лого // Вопр. психологии. —1986. —№ 2.
- Брушлинский А.В. Психология мышления и проблемное обучение. —М., 1983.
- Брябрин В.М. Программное обеспечение персональных ЭВМ. —М., 1986.
- Буш Г.Я. Диалогика и творчество. —Рига, 1985.
- Буш Г.Я. Рождение изобретательских идей. —Рига, 1976.
- Буш Р., Мостеллер Ф. Сравнение восьми моделей // Математические методы в социальных науках. —М., 1973.
- Буш Р., Мостеллер Ф. Стохастические модели обучаемости. —М., 1962.
- Быков В.Е., Мартынов А.Н. Экономико-математические модели управления в просвещении. —Томск, 1988.
- Вадюшин В.А., Пальчевский В.В., Фридман Л.С. Технические средства обучения. —Минск, 1987.
- Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. —М., 1985.
- Варсановцев Д.В., Куширенко А.Г., Лебедев Г.В. Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники // Микропроцессорные средства и системы. —1985. —№ 3.
- Василевский Ю.А. Носители магнитной записи. —М., 1989.
- Васильев Ю.В. К вопросу о теории педагогического управления // Новые исследования в педагогических науках. —1983. —№ 2.
- Вегнер П. Программирование на языке Ада. —М., 1983.
- Вельбицкий И.В. Алгебра конструирования алгоритмов и программ // Управляющие системы и машины. —1987. —№ 6.
- Вельбицкий И.В. Технология программирования. —Киев, 1984.
- Веселов Е.Н. Интегрированная система МАСТЕР для ПЭВМ. —М., 1989.
- Вирт Н. Программирование на языке Модула-2. —М., 1987.
- Вирчев Л. О целях обучения и дидактических решениях // Современ. высш. школа. —1975. —№ 4.
- Возрастная и педагогическая психология. —М., 1984.
- Войсхунский А.Е. О двусторонней коммуникации в системах "человек—ЭВМ" // Искусственный интеллект и психология. —М., 1976.
- Габай Т.В. Учебная деятельность и ее средства. —М., 1988.
- Гаспарский В. Праксеологический анализ проектно-конструкторских разработок. —М., 1978.
- Гастев Ю.А. Гомоморфизмы и модели. —М., 1975.
- Гершунский Б.С. Компьютеризация в сфере образования: проблемы и перспективы. —М., 1987.
- Глухов А.А., Заславский Д.П. Вопросы прогнозирования высшего образования. —Воронеж, 1973.
- Глушков В.М. Введение в АСУ. —Киев, 1974.
- Глушков В.М. Макроэкономические модели и принципы построения ОГАС. —М., 1975.
- Глушков В.М. Основы безбумажной информатики. —М., 1987.
- Глушков В.М., Вельбицкий И.В. Технология программирования и проблемы ее автоматизации // Управляющие системы и машины. —1976. —№ 6.
- Глушков В.М. Фундаментальные исследования и технология программирования // Программирование. —1980. —№ 2.

- Гордон Д. Вычислительные аспекты имитационного моделирования // Исследование операций. Т.1. Методологические основы и математические методы. — М., 1981.
- Гребень И.И., Догьялло А.М. Автоматические устройства для обучения. (Обучающие машины). — Киев, 1965.
- Грис Д. Конструирование компиляторов для цифровых вычислительных машин. — М., 1975.
- Грисуолд Р., Поудж Дж., Полонски И. Язык программирования СНОБОЛ-4. — М., 1980.
- Давыдов В.В. Проблемы развивающегося обучения. — М., 1986.
- Деисс У. Операционные системы. — М., 1980.
- Демин М.В. Игра как специфический вид человеческой деятельности // Филос. науки. — 1983. — № 2.
- Денек К. Дидактическое программирование в высшей школе // Современ. высш. школа. — 1982. — № 3(39).
- Денинг В., Эссиг Г., Маас С. Диалоговые системы "человек—ЭВМ". Адаптация к требованиям пользователя. — М., 1984.
- Джеймс Д.Фолл. Человеко-машинные интерфейсы // В мире науки. — 1987. — № 12.
- Джонс Дж.К. Инженерное конструирование. Современные методы проектного анализа. — М., 1976.
- Джонсон У.Л., Солоуэй Э. PROUST (автоматический отладчик для программ на языке Паскаль) // Реальность и прогнозы искусственного интеллекта. — М., 1987.
- Джонсон Х.Д. Драматизация как форма воспитания // Свободное воспитание. — 1914/1915. — № 7.
- Диалектика познания. — Л., 1983.
- Диалоговые системы в АСУ. — М., 1983.
- Диалоговые системы. Современное состояние и перспективы развития. — Киев, 1987.
- Догьялло А.М. Диалог пользователя и ЭВМ. Основы проектирования и реализации. — Киев, 1981.
- Догьялло А.М., Никулин В.Н., Петрушин В.А. Средства обработки знаний и учебная информатика // Теоретические и учебные аспекты информатики. — К., 1987.
- Дорониш В.Б., Рылова Т.Н. Методология профессионального обучения технологии программирования в учебной информатике // Управляющие системы и машины. — 1988. — № 6.
- Дракин В.И., Попов Э.В., Преображенский А.Б. Общение конечных пользователей с системами обработки данных. — М., 1988.
- Дрынков А.В. Математические модели процесса научения // Математическая психология: методология, теория модели. — М., 1985.
- Дудников Е.Е. Персональные компьютеры. — М., 1988.
- Дюи Д., Дюи Э. Школы будущего. V. Игра // Свободное воспитание. — 1916/1917. — № 2.
- Дьяченко М.И., Кандыбович Л.А. Психологические проблемы готовности к деятельности. — Минск, 1976.
- Ершов А.П. Об объектно-ориентированном взаимодействии с ЭВМ // Микропроцессорные средства и системы. — 1983. — № 3.
- Естественный язык, искусственные языки и информационные процессы в современном обществе. — М., 1988.
- Ефимов В.М., Комаров В.Ф. Введение в управленческие имитационные игры. — М., 1980.
- Жимерин Д.Г., Маснишов В.А. Автоматизированные и автоматические системы управления. — М., 1979.
- Журавлев А.П. Языковые игры на компьютере. — М., 1988.
- Задачи и методы ОАСУ просвещением. — М., 1981.
- Звенигородский Г.А. Первые уроки программирования. — М., 1985.
- Зелкович М., Шоу А., Гэннон Дж. Принципы разработки программного обеспечения. — М., 1982.
- Золотова Г.А. Синтаксический словарь. Репертуар элементарных единиц русского синтаксиса. — М., 1988.

- Ибрагимов О.В.* Инструментальная экспертная система, основанная на байесовском методе принятия решений // Разработка и применение экспертно-обучающих систем. —М., 1989.
- Ибрагимов О.В.* Экспертно-обучающая система по байесовскому методу принятия оптимального статистического решения / 4.1. Методы и системы технической диагностики // Экспертные обучающие системы. —1989. —В.12.
- Игровое моделирование. Методология и практика.* —Новосибирск, 1987.
- Ильцов И.И.* Структура процесса учения. —М., 1986.
- Информатика и компьютерная грамотность.* —М., 1988.
- Искусственный интеллект / Кн.1. Системы общения и экспертные системы.* —М., 1990.
- Каймин В., Угринович Н.* О преподавании курса ОИВТ по машинному варианту // Информатика и образование. —1989. —№ 2.
- Калиниченко Л.А.* Методы и средства интеграции неоднородных баз данных. —М., 1983.
- Калиниченко Л.А., Чабан И.А.* Системы интеграции неоднородных баз данных // Управляющие системы и машины. —1981. —№ 4.
- Капица П.Л.* Эксперимент, теория, практика. —М., 1974.
- Каптелинин В.Н.* Психологические проблемы формирования компьютерной грамотности школьников // Вопр. психологии. —1986. —№ 5.
- Катица Г.* Язык Фортран 77. —М., 1982.
- Кейтер Д.П.* Компьютеры-синтезаторы речи. —М., 1985.
- Кернцан Б., Ритчи Д., Фаюэр А.* Язык программирования Си. —М., 1985. КОБОЛ. —Киев, 1978.
- Коддингтон Л.* Ускоренный курс КОБОЛа. —М., 1974.
- Койчибаев Б.А.* Имитационные методы управления учебными процессами. —Алма-Ата, 1981.
- Коллекция. Личность. Общение. Словарь социально-социологических понятий.* —Л., 1987.
- Компьютер в обучении: психолого-педагогические проблемы (Круглый стол) // Вопр. психологии.* —1986. —№ 5.
- Концепция информатизации образования // Информатика и образование.* —1988. —№ 6.
- Корнева Л.А.* Эволюция языка ПЛИН // Прикл. информатика. —1982. —В.1.
- Корнеев М.А.* Использование компьютера в формировании у учащихся психологической готовности к труду // Вопр. психологии. —1986. —№ 6.
- Корнейчук В.И., Сороко В.Н., Захаревич К.Г.* Машинная генерация учебных контрольных заданий в АОС // Управляющие системы и машины. —1985. —№ 2.
- Короткова Н.А.* Современные исследования детской игры // Вопр. психологии. —1985. —№ 2.
- Кочетков Г.В.* Могущество и бессилие компьютера. —М., 1988.
- Краснов М.А., Лыских А.Ф., Мазурук А.А. и др.* Макросредства управления отладкой учебных программ в ПИ АОС-ВУЗ/ПРИМУС // Теория и практика разработки программных комплексов. Математические исследования. —1986. —В.86.
- Краснов М.А., Рынгаз В.Д.* Диалог и многофункциональные автоматизированные учебные курсы для обучения программированию // Диалоговые системы. Современное состояние и перспективы развития. —Киев, 1987.
- Краткий психологический словарь.* —М., 1985.
- Кратки... словарь по социологии.* —М., 1988.
- Кузин Е.С.* Представление знаний в системе интеллектуального интерфейса // Принципиальные вопросы теории знаний. —Тарту, 1984.
- Купицевич Ч.* Основы общей дидактики. —М., 1986.
- Кучинский Г.М.* Диалог и мышление. —Минск, 1983.
- Кучинский Г.М.* Психология внутреннего диалога. —Минск, 1988.
- Лауров С.С.* Представление и использование знаний в автоматизированных системах // Микропроцессорные средства и системы. —1986. —№ 3.

- Лебедев Г.В.* Разработка интерактивных программ на основе принципа непосредственного редактирования информации // Микропроцессорные средства и системы. —1986. —№ 1.
- Левада Ю.А.* Игровые структуры в системах социального действия // Системные исследования. —М., 1984.
- Лернер И.Я.* Проблемное обучение. —М., 1974.
- Лимантов Ф.С.* Лекции по логике вопросов. —Л., 1975.
- Лингарт И.* Процесс и структура человеческого учения. —М., 1970.
- Лингвистические* вопросы алгоритмической обработки сообщений. —М., 1983.
- Лильят А.* Архитектура малых вычислительных систем. —М., 1981.
- Лобанов Ю.И.* Задача дидактического программирования в АОС // Математические модели и вычислительная техника в управлении учебным процессом высшей школы. —Рига, 1986. —Ч.1.
- Лобанов Ю.И., Савельев А.Я. и др.* Подготовка информации для автоматизированных обучающих систем. —М., 1986.
- Лобанов Ю.И.* Телеологический подход к оценке эффективности компьютерного обучения // Создание автоматизированных систем и комплексов для высшей школы на базе персональных ЭВМ. —Казань, 1988. —Т.1.
- Логико-психологические* основы использования компьютерных учебных средств в процессе обучения // Информатика и образование. —1989. —№ 3.
- Логическое программирование.* —М., 1988.
- Лозовский В.С.* Семантические сети // Представление знаний в человеко-машинных и робототехнических системах. —М., 1984.
- Ломов Б.Ф.* Методологические и теоретические проблемы психологии. — М., 1984.
- Майерс Г.* Архитектура современных ЭВМ. —М., 1985. — Кн.1.
- Майерс Г.Дж.* Искусство тестирования программ. —М., 1982.
- Малашин И.И., Сидорова И.И.* Тренажеры для операторов АЭС. — М., 1979.
- Малютин Э.А., Малютин Л.В.* ПЛ/1 для начинающих. —М., 1985.
- Маргулис Е.Д.* Компьютерная игра в учебном процессе // Сов. педагогика. — 1989. —№ 4.
- Маргулис Е.Д.* Психологические особенности учебной игры с использованием компьютера // Всп. психологии. —1988. —№ 2.
- Маргулис Е.Д., Иващенко Е.Л., Стрижак О.Е.* Психологічні особливості використання гри за допомогою комп'ютера у навчальному процесі // Психологія. —1987. —В.28.
- Мартин Дж.* Организация баз данных в вычислительных системах. — М., 1980.
- Марчук Ю.Н.* Проблемы машинного перевода. —М., 1983.
- Маслов С.Ю.* Теория дедуктивных систем и ее применение. —М., 1986.
- Математический энциклопедический словарь.* —М., 1988.
- Матюшкин А.М., Петросян А.Г.* Психологические предпосылки групповых форм проблемного обучения. —М., 1981.
- Матюшкин А.М.* Проблемные ситуации в мышлении и обучении. —М., 1972.
- Маурер У.* Введение в программирование на языке ЛИСП. —М., 1976.
- Махмутов М.И.* Проблемное обучение. — М., 1975.
- Майшиц Е.И., Андриеская В.В., Коллисарева Е.Ю.* Диалог в обучающей системе —Киев, 1989.
- Майшиц Е.И., Каптелини В.Н., Маргулис Е.Д.* Введение в язык Лого. —Киев, 1989.
- Майшиц Е.И.* Основы управления учебной деятельности. —Киев, 1989.
- Майшиц Е.И.* Психологические основы управления учебной деятельности. —Киев, 1987.
- Майшиц Е.И.* Психолого-педагогические проблемы компьютеризации обучения. —М., 1988.
- Мельников И.А., Монкус В.В., Тамм Б.Г.* Обзор и анализ зарубежных компьютерных обучающих систем в области программирования // Прикл. информатика. —1989. —В.15.
- Мендельсон Э.* Введение в математическую логику. —М., 1984.
- Методы автоматического анализа и синтеза текста.* —Минск, 1985.

- Минский М. Фреймы для представления знаний. —М., 1979.
- Митин А.И., Пасхин Е.Н. Программно-методическое обеспечение подготовки учебного материала для автоматизированных обучающих систем // Сборник научно-методических статей по математике. —М., 1988. —В.15.
- Михалевич В.С., Довгялло А.М., Савельев А.Я. и др. Экспертно-обучающие системы в комплексе компьютерных средств обучения // Соврем. высш. школа. — 1988. —№ 1(61).
- Мобильность программного обеспечения. —М., 1980.
- Моляко В.А. Психология конструкторской деятельности. —М., 1983.
- Монахов В.М. Информационная технология обучения с точки зрения методических задач реформы школы // Вопр. психологии. —1988. —№ 2.
- Монахов В.М. Психолого-педагогические проблемы обеспечения компьютерной грамотности учащихся // Там же. —1985. —№ 3.
- Нариньяни А.С. Проект ЗАПСИБ — серия лингвистических процессов для взаимодействия с базами данных // Вопросы разработки прикладных систем. — Новосибирск, 1979.
- Нефедов А.В., Савченко А.М., Феоктистов Ю.Ф. Зарубежные интегральные микросхемы для промышленной электронной аппаратуры. —М., 1989.
- Никитина С.Е. Тезаус по теоретической и прикладной лингвистике. —М., 1978.
- Нильсон Н. Принципы искусственного интеллекта. —М., 1985.
- Обучающие машины, системы и комплексы. Справочник. — Киев, 1986.
- Общесистемное проектирование АСУ реального времени. —М., 1984.
- Ожогин В.Я. Технические средства в учебном процессе. Информационные свойства и эргономические особенности применения. —Киев, 1984.
- Основные направления перестройки высшего и среднего специального образования в стране. —Киев, 1987.
- Основы компьютерной грамотности. —Киев, 1988.
- Основы создания больших АСУ. —М., 1979.
- Пажитнов А.Л. Логическая структура компьютерной игры // Микропроцессорные средства и системы. —1987. —№ 3.
- Пайл Э. Ада — язык встроенных систем. —М., 1984.
- Парр Э. Знакомство с микро-ЭВМ. —М., 1989.
- Педагогика. —М., 1983.
- Пейган Ф. Практическое руководство по АЛГОЛУ-68. —М., 1979.
- Пересмотренное сообщение об Алголе-68. —М., 1979.
- Персональные компьютеры Единой системы ЭВМ. —М., 1988.
- Петрушин В.А., Отенко В.И., Усенко Р.Д. Мобильная реализация автоматизированной обучающей системы на мини- и микро-ЭВМ // Методы и средства системного программирования. —Киев, 1984.
- Петухов И.А., Андреев Ю.Д. Вы собираетесь приобрести персональную ЭВМ?! -Л., 1989.
- Пильщикова В.Н. Язык плэнэр. —М., 1983.
- Питровская К.Р. Хелп-система для работы с иноязычным текстом на персональном компьютере // Терминологическое обеспечение научно-технического прогресса. —Омск, 1988.
- Поля Дж. Математика и правдоподобные рассуждения. —М., 1975.
- Попов Э.В. Общение с ЭВМ на естественном языке. —М., 1982.
- Попов Э.В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ. —М., 1987.
- Поспелов Г.С. Искусственный интеллект — основа новой информационной технологии. —М., 1988.
- Поспелов Г.С., Поспелов Д.А. Искусственный интеллект — прикладные системы. —М., 1985.
- Поспелов Д.А. Ситуационное управление. Теория и практика. —М., 1986.
- Пратт Т. Языки программирования. Разработка и реализация. —М., 1979.
- Представление знаний в человеко-машинных и робототехнических системах. / Т.А. Фундаментальные исследования в области представления знаний. Т.В. ИнSTRU-

- ментальные средства разработки систем, ориентированных на знания. Т.С. Прикладные человеко-машинные системы, ориентированные на знания. —М., 1984.
- Представление и использование знаний.* —М., 1989.
- Применение ЭВМ в учебном процессе.* —М., 1969.
- Психологические особенности обучения групповой деятельности по решению задач с помощью ЭВМ // Психологическая наука и педагогическая практика.* —Киев, 1983.
- Психологические проблемы автоматизации научно-исследовательских работ.* —М., 1987.
- Психологический словарь.* —М., 1983.
- Психология. Словарь.* —М., 1990.
- Психолого-педагогические основы использования ЭВМ в вузовском обучении.* —М., 1987.
- Психология программного навчання.* —К., 1973.
- Пяринуу А.А. Программирование на Алголе и Фортране.* —М., 1978.
- Ревзин И.И. Современная структурная лингвистика. Проблемы и методы.* —М., 1977.
- Решение задач обработки данных с помощью ЭВМ.* —Киев, 1978.
- Розин В.М. Методологический анализ деловой игры как новой области научно-технической деятельности и знания // Вопр. философии.* —1986. —№ 6.
- Рубинштейн С.Л. Принципы и пути развития психологии.* —М., 1959.
- Рубцов В.В. Компьютер в школе (опыт, проблемы и перспективы) // Вильямс Р., Маклин К. Компьютеры в школе.* —М., 1988.
- Рынгаз В.Д., Краснов М.А., Лысков А.Ф. и др. Автоматизированные системы для обучения программированию.* —Кишинев, 1989.
- Рынгаз В.Д., Краснов М.А., Лысков А.Ф. и др. Концепции и возможности многофункциональных АОС для обучения программированию // Математическое обеспечение вычислительных машин и систем. Математические исследования.* —Кишинев, 1985. —В.81.
- Рынгаз В.Д. О проблемном подходе к автоматизированному обучению программированию с помощью ЭВМ // Технические средства обучения в вузах.* —Кишинев, 1979.
- Рынгаз В.Д., Алексеев Е.А., Копычко С.Н. и др. Обучающе-программирующая система ДИПРОФОР и некоторые результаты ее экспериментального исследования // Управляющие системы и машины.* —1975. —№ 6.
- Савельев А.Я., Новиков В.А. Внедряется СПОК/ВУЗ // Вестн. высш. школы.* —1982. —№ 10.
- Савельев А.Я., Новиков В.А., Лобанов Ю.И. Подготовка информации для автоматизированных обучающих систем.* —М., 1986.
- Самофалов К.Г., Корнейчук В.И., Тодоров Г.С. Использование методов прогнозирования для повышения эффективности АСУ высшей школой // Механизация и автоматизация управления.* —1977. —№ 1.
- Свиридов А.П. Основы статистической теории обучения и контроля знаний.* —М., 1981.
- Семантика языков программирования.* —М., 1980.
- Симонс Дж. ЭВМ пятого поколения: компьютеры 90-х годов.* — М., 1985.
- Синяк В.С. Основы создания ОАСУ.* —М., 1978.
- Система программирования обучающихся курсов.* —Киев, 1981.
- Система разделения времени ЕС ЭВМ. Справочное пособие.* —М., 1982.
- Системное проектирование АСУ хозяйством области.* —М., 1977.
- Скуржихин В.И., Шифрин В.Б., Дубровский В.В. Математическое моделирование.* —Киев, 1983.
- Словарь по кибернетике.* —Киев, 1989.
- Словарь терминов по информатике на русском и английском языках.* —М., 1971.
- Слюком Дж. Обзор разработок по машинному переводу: история вопроса, современное состояние и перспективы развития // Новое в зарубежной лингвистике / В.24. Компьютерная лингвистика.* —М., 1989.
- Смирнов И.Н. Социально-философские проблемы информатики // Вопр. философии.* —1986. —№ 10.

- Сороко В.Н., Захаревич К.Г. Методологические аспекты создания информационно-методического обеспечения АОС // Вестн. Киев. политехн. ин-та. Серия научно-методическая. —1988. —В.12.
- Социальные и методологические проблемы информатики, вычислительной техники и средств автоматизации (Материалы "Круглого стола") // Вопр. философии. —1986. —№ 9—11.
- Средства формирования учебно-игровых средств для младших школьников. —Киев, 1988.
- Стефанюк В.Л., Алексеева Е.Ф. К созданию интеллектуальной операционной системы // Алгоритмы обработки экспериментальных данных. — М., 1986.
- Стрижак А.Е. АОС для средней школы. Вопросы реализации и развития // Развитие методов и средств автоматизированного обучения. —М., 1987.
- Стрижак А.Е. Применение многофункциональных АУК при обучении программированию учащихся средних школ // Диалоговые системы. Современное состояние и перспективы развития. —Киев, 1987.
- Талызина Н.Ф., Габай Т.В. Пути и возможности автоматизации учебного процесса. —М., 1977.
- Талызина Н.Ф. Теоретические проблемы программированного обучения. — М., 1969.
- Талызина Н.Ф. Технология обучения и ее место в педагогической теории // Современ. высш. школа. —1977. —№ 1(17).
- Талызина Н.Ф. Управление процессом усвоения знаний. —М., 1984.
- Текстова перевірка знань учнів. —Киев, 1973.
- Тельнов Н.И. Технические средства пропаганды и обучения. —М., 1985.
- Технические устройства обучения. —Киев, 1985.
- Тихомиров О.К. Информатика и новые проблемы психологической науки // Вопр. философии. —1986. —№ 7.
- Тихомиров О.К., Бабачин Л.Н. ЭВМ и новые проблемы психологии. —М., 1986.
- Тихомиров О.К., Гурьева Л.П. Психологический анализ трудовой деятельности, опосредствованной компьютерами // Психологический журнал. —1986. —№ 5.
- Толковый словарь по вычислительным системам. —М., 1989.
- Тренажерные системы. —М., 1981.
- Трудовая подготовка учащихся в межшкольных комбинатах. —Киев, 1988.
- Турский В.М. Методология программирования. —М., 1981.
- Турчин В.Ф. Алгоритмический язык рекурсивных функций (рефал). Описание языка и приемы программирования. —М., 1968.
- Тыгуев Э.Х. Концептуальное программирование. —М., 1984.
- Ульман Дж. Основы систем баз данных. —М., 1983.
- Уотерман Д. Руководство по экспертным системам. —М., 1989.
- Уэйт М., Прата С., Мартин Д. Язык Си. Руководство для начинающих. —М., 1988.
- Фараопова Э.А. Воспитание психологической готовности к труду // Вопр. психологии. —1985. —№ 6.
- Физиология речи. Восприятие речи человеком. —Л., 1976.
- Флорес И. Структуры и управление данными. —М., 1982.
- Фокс Дж. Программное обеспечение и его разработка. —М., 1985.
- Форсайт Р. Паскаль для всех. —М., 1986.
- Фойш Дж., ван Дэм А. Основы интерактивной машинной графики / Кн.1—2. —М., 1986.
- Фрилинг Г., Ауэр К. Человек-цвет-пространство. Прикладная цветопсихология. —М., 1973.
- Фути К., Судзуки Н. Языки программирования и схемотехника СБИС. —М., 1988.
- Халилов А.И., Ющенко А.А. АЛГОЛ-60. —Киев, 1979.
- Хоггер К. Введение в логическое программирование. —М., 1988.
- Хротко Г. Языки программирования высокого уровня для микро-ЭВМ. —М., 1985.
- Цаленко М.Ш. Семантические и математические модели баз данных. —М., 1985.
- Человеко-машинные системы. —М., 1977.

- Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. — М., 1983.
- Четаериков В.Н., Резунков Г.И., Самохвалов Э. Н. Базы и банки данных. — М., 1987.
- Чижов А.А. Некоторые соображения по поводу компьютерных вирусов // В мире персональных компьютеров. — 1988. — № 1.
- Чубук Ю.Ф. и др. Принципы научной организации высшего строительного образования. — Киев, 1970.
- Шеннон Р. Имитационное моделирование систем — искусство и наука. — М., 1978.
- Шнейдерман Б. Психология программирования. Человеческие факторы в вычислительных и информационных системах. — М., 1984.
- Шоу А. Логическое программирование операционных систем. — М., 1981.
- Экспертные системы. Принципы работы и примеры. — М., 1987.
- Эллингтон Г., Эддинг П., Персивал Ф. Игры, имитации и социальная значимость науки // Импакт. Наука и общество. — 1984. — № 2.
- Элти Д., Кумбс М. Экспертные системы. Концепции и примеры. — М., 1987.
- Эльконин Д.Б. Психология игры. — М., 1978.
- Эстес В.К. Статистические модели вспоминания и опознавания возбуждающих образов человеком // Самоорганизующиеся системы. — М., 1964.
- Язык компьютера. — М., 1989.
- Язык ПРОЛОГ в пятом поколении ЭВМ. — М., 1988.
- Ясин Е. Теоретические проблемы развития информационных систем // Модели данных и системы баз данных. — М., 1979.
- Anderson C, Dewolf M. Computers in the Curriculum, ed. by London, 1984.
- Baker F.T. Chief Programmer Meam. Management of Production Programming // IBM Syst. J. — 1972. — V.11. — № 1.
- Clancy S. Viruses, Trojan horses and other badware. Information and implications for online searchers. // Database. — 1988. — V.11. — № 4.
- Clancy W. J.GUIDON // J.of Computer-Based Instr. — 1983. — V.10. — № 182.
- Cohen F. On the implications of Computer viruses and methods of defence // Computers Security. — 1988. — V.7. — № 2.
- Denning Peter J. The science of computing: Computer viruses. // Amer. Scient. — 1988. — V. 76. — N 3.
- Dewdney A.K. A Core war // Scient. Amer. — 1984. — V. 250. — № 5.
- Dewdney A.K. A Core war bestiary of viruses, worms and other threats to computer memories // Scient. Amer. — 1985. — V.252. — № 3.
- Koffman E.B., Blount S.E. Artificial intelligence and automatic programming in CAI // Artif. intelligence. — 1975. — V. 6. — № 3.
- Marshall E. The scourge of computer viruses // Science. — 1988. — V. 240. — N 4849.
- Merrill M.D. Computer Guided Instructional Design // J. of Computer-Based Instr. Spring. — 1984. — V. 11. — № 2.
- Merrill M.D. Prescriptions for an Authoring System // J. of Computer-Based Instr. — 1987. — V. 14. — № 1.
- Mudrick D., Stone D. An Adaptive Authoring System for Computer-Based Instruction // J. of Computer-Based Instr. Sum. — 1984. — V. 11. — № 3.
- Newell A., Simon H.A. Human problem Solving-Englewood cliffs. — Newjersey, 1972.
- Papert S. Mindstorms: Children, computers and powerful ideals. — New York, 1980.
- Reigelath C.M., Merrill M.D., Wilson B.G. & Spiller P.T. The Elaboration Theory of Instruction. A Model for Sequencing and Synthesizing Instruction // Instruct. Science. — 1990. — № 9.
- Schmidt G., Stronlein T. Timetable construction — an annotated bibliography // Comput. J. — 1980. — V. 23. — № 4.
- Sleeman D., Brown I.S. Intelligent Tutoring Systems. — Cambridge, 1982.
- Solomon A. A Trojan war. Pers. Comput // World. — 1988.
- Soloway E., Rubin E., Woolf B. et al. MENO-II: an AI-based programming tutor // J. of Computer Based Instr. — 1983. — V. 10 — № 1.

- Thomas R. Gruber.* Acquiring strategic knowledge from experts. — Int. J.Man-Machine Studies. — 1988. — 29.
- Univerwood J.H.* Linguistics, Computers and the Language teacher. — London — Tokyo, 1984.
- Weinberg G.M.* The Psychology of Computer Programming. Van Nostrand Reinhold. — New York, 1971.
- Wobis B., McDonald D.D.* Building a Computer Tutor: designissues // Computer. — 1984. —№ 9.