

УДК 371:004

**Л. Л. Філіппова**

### **МЕТОДИКА НАВЧАННЯ ЛОГІЧНОМУ ПРОГРАМУВАННЮ**

**Постановка проблеми в загальному вигляді.** Сучасне суспільство, у якому інформація набуває ролі соціально-значущого ресурсу потребує висококваліфікованих фахівців, які б вільно володіли інформаційними технологіями (ІТ) і ефективно використовували б їх у своїй майбутній професійній діяльності. Стрімкий перехід суспільства в інформаційну епоху, комп'ютеризація всіх ланок освіти вимагають від бакалаврів з економічної кібернетики нових знань і умінь для задоволення ринкового попиту виробництва.

В Законі України «Про основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки» відмічено, що Україна готує і має значну кількість висококваліфікованих фахівців з ІТ, математики, кібернетики; у країні постійно зростає та поновлюється парк комп'ютерної техніки, сучасних систем та засобів телекомунікації, зв'язку; високою є ступінь інформатизації банківської сфери. Ці та інші передумови дають підстави вважати, що вітчизняний ринок ІТ перебуває у стані активного становлення та за певних умов може стати фундаментом розвитку інформаційного суспільства в Україні [1].

**Аналіз останніх досліджень і публікацій.** Аналіз психолого-педагогічної літератури дозволив з'ясувати різне бачення науковців на процес навчання студентів.

Як показують наукові дослідження удосконаленню підготовки фахівців в умовах нових інформаційних технологій присвячено ряд робіт (А.Гуржій, О.Євдокимов, А.Єршов, М.Жалдак, Ю.Жук, Н.Тверезовська), які орієнтують на використання у навчальному процесі передових педагогічних технологій, що передбачають урахування індивідуальних особливостей студентів, закладають фундамент їхньої освіти та саморозвитку.

**Мета статті.** Показати методику навчання логічному програмуванню.

**Виклад основного матеріалу.** Щоб матеріал сприймався і «відтворювався» студентами на високому рівні і після закінчення курсу, слід підключити «внутрішню мотивацію». У статті [4] автор описав свою точку зору щодо того, як цього можна досягти. Вона базується на досвіді викладання психології з використанням «Інтерактивного підходу», що робить методику універсальною щодо застосування до будь-яких дисциплін, де головною особливістю інтерактивного підходу є його діалогічний характер.

Якісне засвоєння навчального матеріалу можливе лише за умови співвідношення його змісту з особистістю того, хто навчається. Студент повинен «бачити» матеріал не як щось абстрактне по відношенню до нього, а так, якби це безпосередньо його стосувалося і торкалося, пов'язане з ним і його життям. Це дозволяє студенту привласнити і засвоїти навчальний матеріал як щось суттєво близьке і споріднене. Якщо цього не відбувається, то спостерігається так зване відторгнення навчального матеріалу, як чогось стороннього, і студент благополучно забуває його після закінчення вивчення дисципліни.

Виходячи з таких міркувань, потрібно будувати лекційні та лабораторні (практичні) заняття таким чином, щоб студенти відчували себе безпосередніми учасниками того, що відбувається.

Головне, що потрібно від студентів, це давати аргументований, обґрунтований аналіз матеріалу, висувати версії можливого рішення. Як правило, при такій формі проведення заняття тих що нудькують і байдужих не спостерігається. Студентам особливо подобається те, що викладач просить висловити саме їхні власні думки. Таким чином, встановлюється контакт між студентами і викладачем, відбувається визначення точок діалогу, що є необхідною умовою майбутнього співробітництва.

Викладач виявляє себе саме як партнер навчального спілкування. Він відкрито повідомляє про свої параметри оцінок і знань студентів, тобто оцінює вміння опрацьовувати літературні джерела, читати конспект, відповідаючи на поставлені питання, їх власні думки і думки щодо нового матеріалу.

Логічне програмування (ЛП) в якості об'єкта вивчення прийшло до ВНЗ набагато раніше об'єктного. У період з 1987 по 1995 р. С.Григор'євим, О.Єрохіною, В.Кайміним, Н.Угріновичем, А.Щеголевым та іншими авторами були розроблені численні методичні матеріали по ЛП. Але спеціального посібника з організації курсу «Технологія створення програмних та інтелектуальних систем» або нових методичних розробок, поки що немає.

ЛП і створені на його основі системи програмування знаходять все більш широке застосування як інструментальний засіб для вирішення різних економічних завдань. Застосування ЛП при вирішенні ряду завдань [2; 3; 4] дозволяє в десятки разів скоротити довжину програми в порівнянні з процедурним програмуванням і уникнути безпосередньої реалізації такої трудомісткої процедури як перебір з поверненням. В якості прикладів завдань, які ефективно вирішуються засобами логічного програмування, можна вирішувати наступні завдання:

- обробка списків, у тому числі сортування, об'єднання, перетин та ін.;
- отримання різних перестановок;
- робота з деревами (можливість створення і обробки рекурсивних типів даних);

- аналіз тексту;
- розробка експертних систем та ін.

Це досягається за рахунок забезпечення можливості роботи з масивами (бінарні терми і вбудовані предикати для роботи з ними в Visual Prolog), включення потужних бібліотек предикатів різного призначення тощо.

На сучасному етапі розвитку програмного забезпечення (ПО) велика роль відводиться візуальному стилю проектування та розробки програм – «візуальному програмуванню» (ВП).

«ВП (від лат. visualis – зоровий) – передбачає створення додатків за допомогою наочних засобів. При цьому програміст не створює текст програми, а показує, що повинно вийти у кінці її відпрацювання. Текст програми генерується автоматично за допомогою візуального прототипу. ВП ґрунтується на об'єктно-орієнтованому програмуванні і OLE-технології або подібних їй технологіях» [5; 6].

Це дозволяє професіоналам проектувати і розробляти великі програмні комплекси швидше. Для непрофесіоналів ВП дозволяє зосередитися на суті завдання предметної області (її об'єктах, відносинах між ними, поведінці об'єктів або їхньому стані), абстрагуючись від особливостей реалізації в кожному конкретному діалекті мови програмування.

ВП, безперечно, набагато краще відповідає природі людського сприйняття, ніж методи традиційного, текстового програмування. Однак практично всі візуальні засоби потребують доповнення функціями, які не можуть бути представлені у вигляді графічних конструкцій і вимагають текстового вираження та доповнюються спеціальними скриптами, написаними на різних мовах програмування.

Для мов ЛП існує певний досвід створення систем з візуальним проектуванням інтерфейсу програм (сучасні транслятори мови Prolog). Однак системи візуального проектування логічних програм практично не відомі, хоча вивчення і практичне освоєння непрофесіоналами логічного стилю програмування є актуальним завданням сьогодення.

Професійне навчання програмуванню за напрямом підготовки «Економічна кібернетика» передбачає освоєння мови ЛП як інструмента створення інтелектуальних систем. Вивчення основ ЛП виконується в дисципліні «Технологія створення програмних та інтелектуальних систем».

Мова програмування Prolog має свої переваги перед іншими процедурними мовами програмування. Одними з них є:

- для певних завдань програма на Prolog вимагає тільки одну десятину рядків коду в порівнянні з аналогічною програмою на іншій процедурній мові;
- завдяки декларативному підходу, такі добре відомі джерела помилок, як зациклення, усуваються з самого початку;

- Prolog «змушує» програміста починати з добре структурованого опису завдання, тому він може використовуватися і як засіб створення специфікації, і як засіб реалізації продукту [2; 3; 7].

Для кращого сприйняття дисципліни «Технологія створення програмних та інтелектуальних систем» розглянемо методику навчання ЛП.

Системи програмування, з якими доводиться працювати студентам значно відрізняються один від одного. Тому, у будь-якому випадку виклад матеріалу рекомендується вести в рамках стандарту, підкреслюючи особливості і відмінності конкретної системи програмування.

При плануванні курсу викладач повинен виходити з рекомендацій щодо вивчення окремих тем (в тій послідовності, в якій зазвичай вивчають Prolog, хоча деякі теми і допускають перестановку).

У темі «Введення в Prolog» доречно нагадати студентам основні принципи обробки даних, а також наголосити, що неможливо ефективно обробляти інформацію, не маючи розгалужених структур даних, бо ефективність рішення задачі багато в чому пов'язане з тим, наскільки вміло вони структуровані. Однак класичні мови програмування вимагають від програміста повністю описати, у вигляді ланцюжка операторів, порядок обробки інформації. Але це не єдиний шлях. Вже давно реалізуються спроби доручити відтворення логічних міркувань системній програмі, а за людиною залишити лише формулювання завдання. Також доречно підвести студентів до думки, що головна особливість мови Prolog полягає в тому, що вона є не стільки мовою для запису алгоритмів, скільки мовою для опису даних та логіки їх обробки.

Матеріали для цієї теми можна знайти в передмові більшості підручників з Prolog, а також в журнальних статтях тощо. Якщо студенти вже вивчали раніше введення в Prolog, то бесіда ця може бути зведена до нагадування основних принципів позицій і показу декількох програм на Prolog. Якщо ні, то після загальної розмови можна навести елементарні програми на Prolog, не згадуючи ні про які формальні правила їх побудови. Такі програми включають два-три факти і одне питання, відповідь на яке, з точки зору студентів, очевидна. При цьому потрібно підкреслити, що очевидність відповіді для людини означає, що її мозок виконав якусь (хай і дуже просту) програму.

Ще необхідно ввести елементарні уявлення про базу знань, відштовхуючись від протиставлення з добре відомим студентам поняттям бази даних. База знань містить інформацію про зв'язки між об'єктами і складається з фактів і правил, у той час як база даних обмежується тільки фактами. Приклади елементарних баз знань легко знайти у літературі навчальних посібників з Prolog.

Тему «Факти. Предикатна форма представлення фактів. Бази даних. Прості запити» доречно почати з вивчення основ логіки, на якій базується Prolog. На прикладах потрібно показати, що означає

висловлювання (судження) і те, що всяке висловлювання може бути істинним або хибним. Слід дати поняття «твердження» – судження, яке потрібно довести або спростувати.

При роботі над поняттям «факт» слід звернути увагу на те, що в програмах на Prolog під фактами розуміються твердження, істинність яких розуміється. Прийняту в мові предикатну форму запису слід відпрацювати на прямих і зворотних прикладах, починаючи перекладом з мови Prolog на природну мову. Рекомендується на прикладі певного сюжету побудувати базу даних (БД) і задати відповідні питання (запити). Залежно від поставленого питання в сюжеті необхідно навчитися визначати імена фактів, кількість і призначення його аргументів. Вважається, що БД з різними іменами предикатів у фактах є кращими на первинному етапі вивчення матеріалу. Також потрібно ввести поняття змінної.

Паралельно з вивченням мови відбувається знайомство з системою програмування. Розглядаються її особливості, правила запису текстів програм, прийоми їх редагування, способи запису та зчитування на зовнішні носії організації запитів.

Вивчення теми «Складні запити. Правила. Бази знань» можна розпочати з наведення таблиці логічної операції І (одночасна істинність декількох висловлювань). При організації запитів до БД на Prolog відповідно можна перевірити присутність або відсутність декількох фактів одночасно (складний запит). При роботі з конкретною БД від складного запиту можна перейти до такого важливого поняття, як правило. При відпрацюванні правила рекомендується спочатку правило записане на Prolog, перевести на природну мову, а потім з природної мови на Prolog.

Бази знань, що складаються з фактів і правил, дозволяють розглянути більш широкий клас завдань. При їх вирішенні слід звернути увагу на синоніми: однак, але. У деяких випадках при складанні правил може з'явитися необхідність скористатися логічною операцією НІ (заперечення). Тому слід описати і показати на прикладах роботу відповідного вбудованого предиката.

Якщо в попередніх темах розглядалися питання змістовного характеру, то тема «Терми Prolog (дані): константи, змінні, складові терми (структури). Робота Prolog: зіставлення, пошук у базі знань, механізм повернення. Управління роботою Prolog. Вбудовані предикати» має формалізований характер. У ній докладно розглядаються механізми пошуку відповіді на запит і його управління. При розгляді механізму зіставлення слід звернути увагу на використання змінних (область дії, конкретизація, зв'язаність). Так як із змінною пов'язується не область пам'яті, а об'єкт (терм), не слід використовувати термінологію з процедурних мов програмування, а саме: присвоювання, розгалуження, повторення. Серед вбудованих предикатів бажано розглядати лише ті, які необхідні для управління роботою з організації пошуку (зіставлення,

заперечення, відсікання). Щодо предиката зіставлення слід зазначити, що символ рівності (=) в деяких системах програмування (Турбо Пролог) використовується також в якості порівняння та арифметичного оператора.

У темі «Рішення логічних завдань на Prolog» потрібно розглянути два відомих методи розв'язання логічних задач: на встановлення відповідності між кількома множинами та на впорядкування між об'єктами і показати на практичних прикладах, як ці методи реалізуються на Prolog. Велику кількість прикладів логічних завдань можна знайти на форумах в Інтернеті.

У темі «Оператори порівняння. Арифметичні оператори. Предикати введення-виведення. Організація діалогових програм. Рішення задач на пошук в базах знань з використанням операторів порівняння і арифметичних операторів» дається розширення мови введенням перерахованих в заголовку теми операторів та предикатів. При розгляді арифметичних операторів і операторів порівняння слід звернути увагу на відмінності в їх написанні та роботі між стандартом мови і конкретною системою програмування.

Рекурсивний метод, який розглядається у темі «Рекурсія» є єдиним способом вирішення завдань за допомогою мінімальної кількості визначень, які повторюються в процесі пошуку тверджень. Сутність методу можна пояснити тим, що якщо рішення вихідної задачі може бути зведено до розв'язання іншої підзадачі, причому ця підзадача є зменшеним варіантом вихідної задачі і також зводиться до іншої підзадачі, то цей процес називається рекурсією. Треба зауважити, що спосіб розбиття і вирішення підзадачі ідентичний до вихідної задачі, що застосовується, а також наголосити, що для того щоб вирішити задачу рекурсивно, необхідно визначити зв'язок між вихідним і допоміжним завданням та визначити граничну умову.

Тему «Структури даних: списки» можна розпочати з того, що при вирішенні деякого класу задач побудова БД є занадто громіздким заняттям. Завдання буде записане у більш стислому вигляді, якщо скористатися структурою – списком. Після того потрібно вивести і записати деякі основні предикати. Для повного розуміння роботи зі списками рекомендується відлагоджувати програми в режимі ручного трасування.

При вирішенні завдань можуть зустрічатися труднощі з визначенням правил, за допомогою яких вони вирішуються. Це пов'язано з уже сформованим стереотипом вирішення аналогічних завдань алгоритмічно.

У темі «Структури даних: бінарні дерева» слід дати визначення бінарного дерева, показати на прикладах його графове уявлення, зв'язок з лінійними списками, дати визначення і проілюструвати на прикладах збалансовані, впорядковані бінарні дерева.

**Висновки.** Досвід роботи із студентами показав, що для інформаційних технологій такий варіант викладання є запорукою можливості самовизначення студентів і, більш-менш усвідомленого вибору саме такої позиції, яку він сам поділяє і приймає. Іншими словами, йому пропонується можливість вибору і тим самим значно знижується ризик формування шаблонних вузьких уявлень про предмет. У свою чергу задоволення від процесу стає найважливішим мотивом, що спонукає студентів, які знову й знову прагнуть його пережити, а тим самим і мислити.

**Актуальними напрямками подальшої розробки окресленої проблеми є** вдосконалення навчальних програм, вивчення особливостей використання активних методів навчання тощо.

#### **Список використаної літератури**

**1. Закон** України «Про Основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки» // <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=537-16>. **2. Адаменко А.** Логическое программирование и Visual Prolog / А. Адаменко, А. Кучуков. СПб. : БХВ-Петербург, 2003. – 992 с. **3. Братко И.** Алгоритмы искусственного интеллекта на языке PROLOG / И. Братко. М. : Вильямс, 2004. – 640 с. **4. Стерлинг Л.** Искусство программирования на языке Prolog / Л. Стерлинг, Э. Шапиро. М. : Мир, 1990. – 580 с. **5. Чернышов О.Г.** ПРОграммирование в ЛОГике: Учеб. пособие. – Омск: Изд-во ОмГУ, 2004. – 64 с. **6. Визуальное** программирование [Электронный ресурс] / Википедия – свободная энциклопедия, – [http://ru.wikipedia.org/wiki/Визуальное\\_программирование](http://ru.wikipedia.org/wiki/Визуальное_программирование). **7. Суслов А.В.,** Наумов Р.В. Введение в язык Prolog: основы синтаксиса и примеры использования (статья) [Электронный ресурс] / Computer technologies department, [http://rain.ifmo.ru/~suslov/prolog/introduction\\_to\\_prolog.htm](http://rain.ifmo.ru/~suslov/prolog/introduction_to_prolog.htm)

#### **Філіппова Л. Л. Методика навчання логічному програмуванню**

У статті розглядається проблема професійної підготовки фахівців з економічної кібернетики за допомогою логічного програмування. Підкреслюється необхідність засвоєння тем студентами з дисципліни «Технологія створення програмних та інтелектуальних систем».

*Ключові слова:* методика, логічне програмування, інформаційні технології, Prolog.

#### **Филиппова Л. Л. Методика обучения логическому программированию**

В статье рассматривается проблема профессиональной подготовки специалистов по экономической кибернетике с помощью логического программирования. Подчеркивается необходимость

усвоения тем студентами по дисциплине «Технология создания программных и интеллектуальных систем».

*Ключевые слова:* методика, логическое программирование, информационные технологии, Prolog.

**Filippova L. L. Methods of Teaching Logic Programming**

In the article the problem of professional training of specialists in economic cybernetics are viewed with help of logic programming. Emphasizes the need for learning by the students on the subject «Technology development of software and intelligent systems».

*Key words:* methodology, logic programming, information technology, Prolog.

Стаття надійшла до редакції 10.09.2012 р.

Прийнято до друку 28.09.2012 р.