

КОМП'ЮТЕР

у школі та сім'ї



XXV Міжнародна олімпіада з інформатики



IOI

2013

BRISBANE AUSTRALIA



5'2013

XXV МІЖНАРОДНА ОЛІМПІАДА З ІНФОРМАТИКИ (6–13 липня 2013 року, м. Брісбен, Австралія)



Команди України, Росії і Білорусі

КОМП'ЮТЕР

у школі та сім'ї

ISSN 2307-9851

НАУКОВО-МЕТОДИЧНИЙ ЖУРНАЛ

Виходить 8 разів на рік.
Видається з лютого 1998 року.

Засновники: Інститут педагогіки
НАПН України, Інститут інформаційних
технологій і засобів навчання НАПН
України, редакція журналу.

Журнал видається за сприяння
Міністерства освіти і науки України.

Свідоцтво про реєстрацію серія КВ
№12217-1101ПР від 17.01.2007.

Передплатний індекс 74248.

Журнал зареєстровано ВАК України
як фаховий у галузі педагогічних наук,
постанова Президії ВАК України
№1-05/2 від 27.05.2009.

Затверджено Вченою радою
Інституту педагогіки НАПН України,
протокол №9 від 27.06.2013 р.

Головний редактор
РУДЕНКО В.Д.

Заступник головного редактора
ЛАПІНСЬКИЙ В.В.

Редактор
КИРИЧКОВ Я.В.

Редакційна рада:
ГЛАДКОВСЬКИЙ Р.В.

ГУРЖІЙ А.М.

ЖУК Ю.О.

ЗГУРОВСЬКИЙ М.З.

КЛОКАР Н.І.

ЛИТВИНОВА С.Г.

ОЛІЙНИК В.В.

ПАНЬКОВ А.В.

Редакційна колегія:

БИКОВ В.Ю.

ГОЛОВКО М.В.

ГРИГОР'ЄВ С.Г.

ГРИНШКУН В.В.

ЖАЛДАК М.І.

ЖЕБРОВСЬКИЙ Б.М.

КАЛІНІНА Л.М.

ПУШКАРЬОВА Т.О.

СПІВАКОВСЬКИЙ О.В.

СПІРІН О.М.

ФОКІНА Т.М.

E-mail: csf221@rambler.ru,
www.csf.vashpartner.com

№5(109) ♦ 2013

ЗМІСТ

ПИТАННЯ ТЕОРІЇ

Пасіхов Ю.Я. Шкільна інформатика: програми, підручники, учителі 3

КОНКУРС «УЧИТЕЛЬ РОКУ 2013»

Костриба О.В. Урок на тему «Моделі та моделювання» 5

Шолом Г.І. Особливості уроку інформатики з розвитку критичного мислення 8

Романчук Т.М. Організація роботи з обдарованими дітьми на уроках інформатики 11

Гетманцев В.С. Формування умінь продуктивно-технологічної діяльності учнів на уроках інформатики 13

Кривко Н.В. Майстер-клас на тему «Інформаційна компетентність» 17

ІНФОРМАТИКА У 5-МУ КЛАСІ

Ігнатченко К.А., Ігнатченко Н.В. Викладання основ робототехніки у 5-8 класах 19

Казанцева О.П. Методика викладання теми «Графічний редактор» в 5-му класі 21

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ НАВЧАННЯ

Литвинова С.Г., Тебенко О.В. Хмарні технології. Соціальне середовище програмування Touchdevelop 26

ОЛІМПІАДИ З ІНФОРМАТИКИ

Бондаренко В.В., Ягіяєв Ш.І. Задачі XXVI Всеукраїнської олімпіади з інформатики та рекомендації щодо їх розв'язування 31

Кузічев М.М. Друга Всеукраїнська олімпіада з інформаційних технологій 38

Мельник В.І., Мотурнак Є.В. Задачі XI Всеукраїнського турніру юних інформатиків 41

КОМП'ЮТЕР В УПРАВЛІНСЬКІЙ ДІЯЛЬНОСТІ

Моцик Б.В. Інформаційне управління якістю освіти в багато-профільному ліцеї 45

НА ДОПОМОГУ ВЧИТЕЛЮ ІНФОРМАТИКИ

Жугастрова О.В. Кодування звукових даних 49

НОРМАТИВНО-МЕТОДИЧНІ ДОКУМЕНТИ

Про використання Інструктивно-методичних матеріалів з питань створення безпечних умов для роботи у кабінетах інформатики та інформаційно-комунікаційних технологій загальноосвітніх навчальних закладів 52

Про надання грифа на навчальну програму «Інформатика. 5-9 класи. Для навчальних закладів з поглибленим вивченням предметів природничо-математичного циклу» 55

Про надання грифа на підручник «Інформатика. 5 клас. Для навчальних закладів з поглибленим вивченням природничо-математичних дисциплін» 55

ІНФОРМАЦІЯ

Успіхи загальноосвітніх навчальних закладів України у Всеукраїнських та Міжнародних олімпіадах з інформатики 56



XXV Міжнародна олімпіада з інформатики

У м. Брісбен (Австралія) з 6 по 13 липня 2013 року відбулася ювілейна XXV олімпіада з інформатики — IOI–2013. У олімпіаді взяли участь 299 учнів із 81 країни. Команда України виступала у такому складі:

Рубаненко Роман Сергійович — учень 11-го класу Полтавського обласного ліцею-інтернату для обдарованих дітей із сільської місцевості при Кременчуцькому педагогічному училищі ім. А.С. Макаренка, (учитель інформатики заслужений вчитель України Мельник Валентин Іванович);

Федоряка Дмитро Сергійович — учень 11-го класу Комунального закладу освіти «Дніпропетровський ліцей інформаційних технологій» при Дніпропетровському національному університеті ім. Олеся Гончара (учитель інформатики Буланій Олександр Павлович);

Фурко Роман Володимирович — учень 11-го класу Полтавського обласного ліцею-інтернату для обдарованих дітей із сільської місцевості при Кременчуцькому педагогічному училищі ім. А. С. Макаренка (учитель інформатики заслужений учитель України Мельник Валентин Іванович);

Шевченко Ілля Олегович — учень 9-го класу Харківського фізико-математичного ліцею №27 Харківської міської ради Харківської області (учитель інформатики Олійникова Ніна В'ячеславівна).

Керівники команди: дійсний член НАПН України, доктор технічних наук, професор **Гуржій Андрій Миколайович** та асистент Київського національного університету імені Тараса Шевченка **Бондаренко Віталій Вікторович**.

Олімпіада IOI–2013 проводилася в два тури тривалістю 5 годин кожний. У кожному турі учасникам пропонувалося для розв'язування три задачі.

Дозволялося використовувати мови програмування Pascal, C, C++.

Усі учасники нашої команди вибороли медалі: **Фурко Роман** — срібну медаль, **Рубаненко Роман**, **Шевченко Ілля** і **Федоряка Дмитро** — бронзові медалі.

Редакція журналу щиро вітає команду школярів України з високим досягненням та зичить їм добра і подальших успіхів!

ШКІЛЬНА ІНФОРМАТИКА: ПРОГРАМИ, ПІДРУЧНИКИ, УЧИТЕЛІ

Пасіхов Юрій Якович,

заступник директора, завідувач лабораторії інформаційно-комунікаційних технологій ФМГ №17 м. Вінниці, заслужений учитель України.



Інформатика — одна з наймолодших шкільних дисциплін. Абсолютно очевидно, що її місце в шкільній освіті, освітні стандарти, методичні прийоми, навчальна література — усе це формується буквально на наших очах за безпосередньої нашої участі. У цьому процесі особливу роль відіграють такі чинники, як **виняткова стратегічна важливість інформаційних технологій для сучасного життя і постійне прискорення темпу їх змін. Спробуємо в комплексі розглянути вимоги до основних функціональних складових вивчення інформатики — програми, як похідної від стандартів освіти, підручника та вчителів.**

Постановка проблеми і її аналіз. Інформатика змінюється значно швидше й динамічніше, ніж інші шкільні предмети. Автори стандартів і програм вимушені постійно «бігти навздогін» за досягненнями ІКТ, що бурхливо розвиваються. З одного боку, це приводить до швидких, а отже, не завжди виважених змін як у змістовій частині курсу, так і взагалі у його парадигмі, а, з другого боку, **забезпечує постійне відставання від реалій сьогодення, особливо коли мова йде про підручники.**

Спробуємо прослідкувати динаміку зміни парадигми шкільного курсу і напрями вектора цих змін.

За більш як двадцятип'ятирічний час розвитку шкільної інформатики ми активно рухалися від переважного вивчення алгоритміки (включаючи її «безмашинний» варіант) до «тотального» освоєння комп'ютерних технологій. Це було пов'язано як з удосконаленням цифрових технологій, розширенням сфери їх застосування, а, отже, суспільним замовленням, потребою у «навчених» користувачах, так із поліпшенням оснащення шкіл комп'ютерною технікою.

Відповідно змінювалися й підручники. Текст сучасного підручника з інформатики все більше і більше став нагадувати гарний (чи не дуже) «help» до того чи іншого продукту компанії MICROSOFT. Практично усі «користувачькі теми» (за більшістю діючих чи запланованих до впровадження в 2013–2014 н. р. програм на вивчення цих тем відведено біля 70% навчального часу, а пропонувалось — усі 100%!!!), у всіх відомих автору підручниках викладено за однією і тією ж схемою:

- автор обирає з якихось власних міркувань ту чи іншу версію операційної системи Windows й офісного пакета;
- далі, згідно теми, що вивчається — набір скріншотів з послідовними вказівками типу «натиснути туди», «клацнути там»;
- інструкції для практичних робіт також відповідно повторюють згадану структуру (можливо, дещо спрощену в даному коментарі, але не змінену за суттю).

Такий підхід породжує безліч проблем. Частина з них «лежить на поверхні». Розглянемо «найвиднішу», але, на наш погляд, не найсуттєвішу.

У школах України на сьогоднішній день можна знайти великий широкий діапазон версій операційних систем (проблема ліцензійної чистоти зараз не обговорюється, повернемося до неї пізніше): від Windows 98 (саме з нею поставлялися ПК на початку 2000-их років, новіші версії працювати на такому застарілому «залізі» не можуть, а заміна таких ПК, як-

що й відбувається, то завершиться не скоро із зрозумілих причин, різноманітні варіації XP, Vista, Windows 7, а подекуди вже й Windows 8. Відповідно широкий діапазон версій пакета Microsoft Office, які суттєво відрізняються за можливостями й інтерфейсом. Немало шкіл (інколи цілими регіонами) використовують безкоштовне ПЗ, зокрема різні версії ОС Linux. До цього слід додати ще 30 тис. «гуманітарних» ПК з Китаю, що надійшли в школи. На них встановлено один із клонів Linux, між іншим, вельми специфічний. Нові програми, на щастя, не вимагають використовувати якусь конкретну версію ОС і додатків: теоретично вчитель може, виконуючи програму, використовувати все з перерахованого і ще багато чого іншого.

Але у будь-якому підручнику описана певна, і не завжди сама поширена версія, саме з неї зроблені скріншоти, і саме для цієї версії задано послідовність натискання і назви кнопок. Ось і виходить, що для значної частини учнів створено ситуацію: за підручником вивчаєш щось одне, на шкільному ПК встановлено щось інше, а на домашньому ще щось, як правило, піратське і «новеньке». Саме прикре, що маючи ліцензію на більш старі версії пакетів чи ОС, учитель, бажаючи спростити життя учням, й «актуалізувати» підручник, вимушений ставити на шкільні ПК те, на що ліцензії не має, стаючи злочинцем за означенням.

Ця проблема не має прямого вирішення й існуватиме завжди. Хоча б тому, що друкований підручник ніколи не зможе «встигнути» за процесами зміни поколінь ПЗ, які динамічно відбуваються, та й ніхто не буде купувати, наприклад, для школи ліцензію на Office 365, якщо є ліцензія на Office 2007, чи на значно старіший Office 2003...

Лише цей чинник (повторюю, не основний, і далеко не єдиний) ставить під сумнів доцільність підручника інформатики в такому вигляді, у якому він зараз існує.

Але це не самий суттєвий недолік нинішніх підручників з інформатики. Якби вдалося заволодіти «чарівною паличкою» і зробити так, що в актуальному нині підручнику описано самі сучасні операційна система й додатки, на які в усіх шкіл є ліцензія, а інших взагалі не існує, чи вони ніде не використовуються — написаний у «загальноприйнятому стилі» підручник кращим не стане.

Справа в тому, що в усіх відомих автору сучасних українських підручниках замість реалізації принципу **«Від усвідомлення — до виконання»**, реалізовано принцип **«Дізнаюся, куди натиснути/де поставити галочку — виконаю»**.

Найгірше в тому, що подібний підхід до підручника, на перший погляд, дозволяє досягти мети, задекларованої у стандарті освіти: «Формування і розвиток проектно-технологічної та інформаційно-комунікаційної компетентностей для реалізації творчого потенціалу учнів і їх соціалізації у суспільстві».

У випадку успішного навчання випускник школи досягає певної компетентності, як користувач, успішно «юзає» ПК, уміє користуватися соціальними мережами, навіть у бага-

твох випадках зможе використовувати комп'ютер у своїй виробничій діяльності (правда, частіше як друкарську машинку чи засіб спілкування, але і це вже не мало). Значно гірший вигляд має справа з реалізацією творчого потенціалу.

На сьогоднішній день постійно зростає потреба країни не лише в грамотних користувачах, а й у фахівцях-професіоналах АТ-галузі. Тобто, якщо ще 15 років тому дійсно найактуальнішою була задача «підготувати компетентного користувача», і зробити це треба у школі, бо тільки там учень може попрацювати хоч і на старенькому, але ПК, то нині перед курсом інформатики має стояти більш амбітна стратегічна задача. Спіраль еволюції зробила черговий виток, а ідеологія шкільного курсу інформатики залишилася на попередньому.

Нині, на новому витку спіралі, базовий курс шкільної інформатики повинен «розвернути» стандарти, програми, підручники й учителя(!) від користувацької інформатики до фундаментальної. І мета цього «розвороту» — створити передумови для всіх школярів України, завдяки яким вони зможуть (у випадку наявності здібностей та бажання рухатись у цьому напрямку) реалізувати себе як фахівця у АТ-галузі. Адже теорему Піфагора вивчають не лише майбутні учені-математики, а будову живої клітини — лише майбутні мікробіологи.

Останніми роками за даними Всеукраїнського громадського об'єднання керівників ІТ-компаній України «Рада з конкурентоспроможності індустрії інформаційно-комунікаційних технологій» різко знизилася конкурентоспроможність випускників ВНЗ, що мають дипломи ІТ-фахівців (а це 20 тис. випускників щороку, і ця цифра зростає). Значна частина їх просто є фахово непридатною. Викладачі з ВНЗ зазначають, що нинішній абітурієнт, який у своїй основній масі в школі не вивчав основи алгоритмізації та програмування, обирає цей фах лише за результатами ЗНО, а зовсім не на основі уподобань і природних можливостей оволодіти цією професією. Він у школі не мав можливості спробувати, вийде в нього це чи не вийде, говорячи простою мовою. А виходить це, як показує світова практика, далеко не в усіх, диплом тут допомогти не може.

Тому сформульована стандартом мета курсу є недостатньою, програми (у своїй більшості), не відповідають стратегічним вимогам часу, а підручники (у тому числі й ті, що вже створені під нову програму) — такі, що не дозволяють не тільки повноцінно вийти на «новий виток спіралі», а й досягти нинішньої, застарілої та спрощеної мети навчання.

У результаті вчителі приходять до неминучого риторичного запитання: «Чи потрібні взагалі підручники з такого предмету в наш час?»

Контингент учителів інформатики набагато більш різномірний, ніж з інших предметів, вони більшою мірою дезорієнтовані, ніж викладачі інших предметів — ще не склалася класична традиція викладання і вичерпна система фахових вимог до вчителя. Сучасний учитель інформатики, маючи запис у дипломі «учитель інформатики» як другий фах, і не маючи базової математичної освіти, у більшості своїй є «технологом», тобто тим, хто, наприклад, знає тільки «Офіс» і не знає навіть елементарного програмування. У значно меншій кількості можна зустріти вчителя-«програміста», який більше нічого не хоче з учнями вивчати, крім програмування у якомусь там програмному середовищі. Зрозуміло, що коректив потребують обидві позиції.

Шляхи розв'язання

1. Потрібна нова концепція шкільного курсу інформатики

Така концепція розробляється. Один із варіантів проекту був надрукований у журналі «Комп'ютер у школі та сім'ї» №1 за 2013 р. Є й інші проекти, зокрема розроблений Всеукраїнським громадським об'єднанням керівників ІТ-індустрії України «Рада з конкурентоспроможності індустрії інформаційно-комунікаційних технологій» за активної участі педагогічної громадськості із залученням міжнародних експе-

ртів. Раніше чи пізніше концепція буде прийнята. **У будь-якому випадку вона повинна врахувати необхідність зміни парадигми шкільного курсу інформатики на ту, що відповідає вимогам сьогодення і національним інтересам України. Відповідно з концепцією зміни повинні торкнутися стандарту і діючих програм.**

2. Потрібно створити нове покоління підручників

Навіть у рамках існуючих стандартів і програм потрібно кардинально змінити підхід до створення підручника з інформатики:

- підручник повинен містити лише фундаментальні поняття курсу згідно діючих програм і з урахуванням вікових особливостей учнів. Він не повинен опиратися а ні на конкретну ОС, а ні на якусь версію того чи іншого пакета. У своїй «користувацькій» частині матеріал має бути присвячений лише методологічним аспектам використання того чи іншого додатку й містити оглядову інформацію про найбільш уживані пакети і версії;
- до підручника обов'язково повинна додаватися низка додатків (найкраще — он-лайн з можливістю завантаження), з описом методів роботи, системи вправ і домашніх завдань для учнів. Такий додаток має бути створений згідно програми для відповідного класу з кожної з можливих ОС і прикладних пакетів. Наприклад, якщо у 6-му класі за новою програмою планується вивчати роботу з текстовими редакторами, то варто мати такі додатки з описом можливостей та інтерфейсу, системою вправ і практичних робіт, вказівками до виконання, тощо, окремо для WORD-2003, WORD-2007, WORD-2010, WORD-2013, WORD-365, Writer. І це не повний перелік. Зрозуміло, що такі додатки повинні бути загальнодоступні й безкоштовні.

Взагалі потреба в «паперовому» підручнику з інформатики нині має дуже архаїчний вигляд. Особливості предмету й умови його вивчення вже на сьогоднішній день дозволяють перейти на електронні версії підручників. Це значно зменшило б витрати і дозволило більш оперативно вносити зміни до підручників, не чекаючи 5-річного терміну їх фізичної придатності.

Інформатика найкраще з усіх шкільних дисциплін (через, хоча б те, що учень має доступ до ПК на кожному уроці) підходить для пілотного впровадження електронного навчання, учителі інформатики краще за інших для цього підготовлені, найбільше у мережі Інтернет розміщено ЕОР (електронних освітніх ресурсів) саме з інформатики. Це ще один аргумент на користь відмови від традиційного паперового підручника в тому вигляді, у якому він є зараз, або як перехідний етап, на повну зміну його ідеології.

3. Підготовка вчителя інформатики у ВНЗ повинна носити більш фундаментальний характер

Підготовка вчителя інформатики у ВНЗ обов'язково повинна вестися на основі базової математичної освіти. І якщо учитель фізкультури чи трудового навчання у рамках свого навчального плану такої освіти не отримує, навіряд чи він зуміє бути вчителем інформатики, здатним виконувати ті завдання, що стоять перед нею. Але, на сьогоднішній день, саме такі фахівці становлять ліву частку «дипломованих» учителів інформатики в країні.

Висновки. Якщо не рухатися вказаних напрямках, то від усіх прогресивних новацій, що зараз відбуваються в шкільній інформатиці (уведення вивчення предмету з 2-го класу, тотальне підключення закладів освіти до мережі Інтернет, нові програми, підручники, інше) ми матимемо результат, діаметрально протилежний очікуваному. Школярі України все більше «зависатимуть» ВКонтакті, реально живе спілкування замінять віртуальним, зможуть, можливо, зробити покупку в Інтернеті, але навіряд чи зможуть заробити собі на цю покупку інтелектуальною працею.

УРОК НА ТЕМУ «МОДЕЛІ ТА МОДЕЛЮВАННЯ»

Костриба Олександр Володимирович,

учитель Білогірського НВК «Середня загальноосвітня школа I–III ступенів ім. І.О. Ткачука, гімназія» Хмельницької області учитель вищої категорії, учитель-методист.



Таблиця 1

	Проект	Проектування	Модель	Моделювання
Визначення, пояснення				
Властивості (етапи)				
Типи				
Приклади				

Варіанти відповідей учнів

Проект — сукупність документів — розрахунків, креслень, макетів, моделей тощо, необхідних для зведення споруд, виготовлення машин, приладів.

Проектування — процес створення проекту, прототипу. У техніці — розробка проектної, конструкторської та іншої технічної документації.

Модель — зменшена копія об'єкта.

Моделювання — процес створення моделі.

Проблемні питання. Чим проект відрізняється від моделі? Що передує модель чи проект? Чому б не досліджувати сам об'єкт, для чого створювати модель?

II. Оголошення теми і мети уроку

III. Пояснення навчального матеріалу

Моделі і способи їх зображення

Людина в будь-якій діяльності постійно користується моделями. Діти граються іграшками — зменшеними копіями реальних об'єктів. Для гри використовують не тільки готові моделі, а й створені власними руками з пластиліну, деталей конструктора. У школі діти знайомляться з іншими моделями: аплікація, малюнок, креслення, глобус, моделі фізичних пристроїв тощо. У подальшому люди також використовують моделі — макет (проект) будинку, автомобіля, моделювання фізичних, хімічних явищ і процесів.

Спробуємо дати визначення поняттю «Модель». Закінчіть речення: «Модель — ...». Учні записують власні думки. Заслуховування й аналіз отриманих висловів.

Модель (від лат. «modulus» — міра, зразок, норма) — це прообраз, опис або зображення якогось об'єкта.

Моделлю можуть бути будь-який об'єкт, установка, явище або мислений образ, за допомогою яких вивчаються складніші об'єкти.

Моделі використовують тоді, коли безпосередньо дослідити відповідні об'єкти-оригінали важко або й неможливо. В іншому випадку моделі використовуються для дослідження ще й неіснуючих об'єктів.

Тема уроку: Поняття моделі. Типи моделей. Моделювання як метод дослідження об'єктів.

Мета: Ввести поняття:

- модель, моделювання;
- інформаційна модель;
- комп'ютерне моделювання.

Сформулювати:

- на основі життєвого досвіду учнів поняття про різні різновиди моделей і їх відображення;
- класифікацію моделей;
- уявлення про моделювання як метод пізнання;
- загальні основи термінології теорії моделювання.

Навчити:

- будувати інформаційну модель задачі (об'єкта).

Формувати:

- уміння чітко і лаконічно висловлювати думки.

Виховувати:

- уважність, дисциплінованість у роботі на ПК.

Базові поняття й терміни: Модель, моделювання, проект, проектування.

Часова модель уроку

1	Організаційний етап	3–5 хв.
2	Оголошення теми уроку	1–2 хв.
3	Вивчення нового матеріалу • Моделі та способи їх зображення. • Інформаційні моделі. • Комп'ютерне моделювання. • Етапи побудови комп'ютерної моделі	15–20 хв.
4	Відновлення навичок роботи з клавіатурою • Виконання комплексу вправ для зняття зорової втоми. • Відновлення навичок набору тексту. • Проведення комплексу вправ для зняття м'язового напруження. • Перевірка швидкості набору	10–12 хв.
5	Узагальнення й осмислення набутих знань	10–12 хв.
6	Підсумок уроку. Оголошення домашнього завдання	2–3 хв.

Хід уроку

I. Організаційний момент. Привітання. Підготовка класу до занять.

Ви починаєте вивчати новий розділ інформатики — «Комп'ютерне моделювання. Основи алгоритмізації». Тема розрахована на 6 годин. Під час вивчення теми ви навчитесь створювати різноманітні алгоритми і програми, виконаєте низку практичних завдань.

Ознайомлення з критеріями оцінювання рівня навчальних досягнень.

Сьогодні на уроці ми виконаємо практичну вправу «Пошук та аналіз» з використанням мережі Інтернет, розглянемо теоретичний матеріал, будемо працювати в середовищі текстового редактора над створенням електронного конспекту.

Вправа «Пошук та аналіз». Учні об'єднуються в 4 групи.

Крім *матеріальних* моделей (іграшки, глобуса, макета будинку), існують *абстрактні* моделі: описи, формули, зображення, схеми, креслення, графіки тощо. За допомогою математичних формул описуються, скажімо, арифметичні операції, співвідношення в геометрії, закони руху і взаємодії тіл, формули ($S=V \cdot t$, C_3H_8 , $c^2=a^2+b^2$, ...). Користуючись таблицями, графіками, діаграмами, можна відобразити різноманітні явища, процеси, закономірності і залежності реального світу.

Усі абстрактні моделі, які можна подати за допомогою набору знаків (геометричних фігур, символів, фрагментів тексту), — це *знакові моделі*. Для побудови знакової моделі потрібно знати значення знаків, що використовуються.

Абстрактні моделі, отримані внаслідок висновків, висловів називаються *вербальними* (від лат. *verbalis* — усний).

Мозковий штурм

Спробуємо провести класифікацію моделей: іграшка, макет будинку, правила для учнів, твір на тему «Як би я був президентом», запис структури молекули води, відповідь на запитання «Чи є життя у всесвіті?»

Наведена класифікація моделей (рис. 1) — це одна з найпростіших класифікацій за способом створення. Можливі й інші класифікації, наприклад, за предметною ознакою: фізичні, хімічні, моделі будівельних конструкцій, різних механізмів тощо.



Рис. 1

Інформаційні моделі

Під час вивчення інформатики нас цікавитимуть *інформаційні моделі*, тобто такі, що стосуються інформаційних процесів. До якої із вказаних вище множин можуть належати ці моделі? По-перше, інформаційні моделі мають бути абстрактними, оскільки, як відомо, інформація — це нематеріальна категорія. По-друге, інформаційні моделі мають бути знаковими, тому що повідомлення зображуються у вигляді знаків. Знакові моделі прийнято поділяти на *математичні* й *інформаційні* моделі.

Математична модель — це модель, що описує об'єкт, явище чи процес мовою математики.

Інформаційна модель — це модель, що описує інформаційні процеси або містить інформацію про властивості і стан об'єктів, процесів, явищ.

До інформаційних моделей можна віднести тексти довідкових видань, енциклопедій.

Якщо модель формулюється так, що її можна обробити на комп'ютері, вона називається комп'ютерною.

Комп'ютерна модель — це модель (математична чи інформаційна), реалізована за допомогою програмних засобів.

Комп'ютерне моделювання

Моделювання на комп'ютері має набагато більше можливостей, ніж просто моделювання за допомогою реальних предметів або матеріалів.

Спробуємо це довести.

Учні об'єднуються в групи і заповнюють таблицю 2. Заслуховуються виступи представників груп. Дискусія.

Етапи побудови комп'ютерної моделі

Спробуємо уявити, з яких етапів складається процес створення комп'ютерної моделі. Взагалі, моделювання — це творчий процес, і розділити його на будь-які етапи і кроки дуже складно. Багато моделей і теорій народжуються внаслідок поєднання досвіду й інтуїції вченого або фахівця. Однак розв'язування більшості конкретних задач все ж таки можна уявити поетапно.

Процес створення комп'ютерної моделі можна уявити як шлях від постановки задачі, тобто від інформаційної моделі, і до її втілення на комп'ютері (рис. 2). Реалізація моделі на комп'ютері відбувається за допомогою програм.

Практична частина. Побудова моделей.

Створити моделі засобами Microsoft Office:

Об'єкти: куб, будинок, ялинка.

Приклад

Куб — правильний багатогранник, кожна грань якого є квадратом (рис. 3).



Рис. 2

Таблиця 2

Об'єкт, явище чи процес	Матеріальна модель (зменшена копія)	Інформаційна модель (опис)	Комп'ютерна модель (програмна реалізація)
Будинок	Великі затрати, не інформаційна, важко вносити зміни		
Ракета	Великі затрати, не інформаційна, важко вносити зміни		
Графік функції	Для кожної функції створюється власний графік		
Дія землетрусу	Практично неможливо реалізувати		
Траєкторія руху планети	Примітивна		

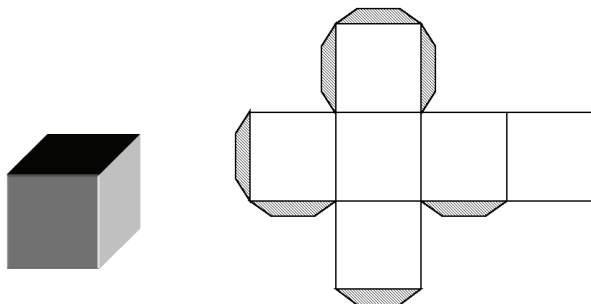


Рис. 3

Процесів: хімічна реакція (HCl+Zn), передавання інформації.

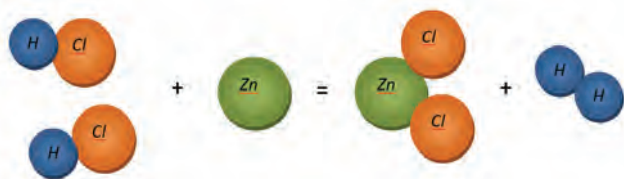
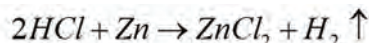


Рис. 4

Приклад

Явищ: Заломлення світла.

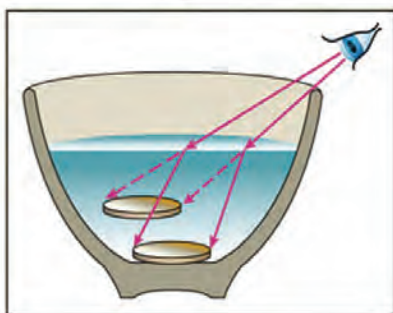
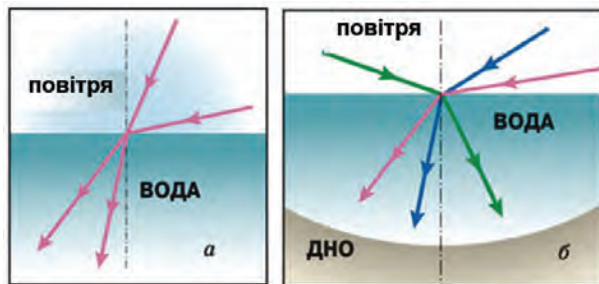


Рис. 5

IV. Відновлення навичок роботи з клавіатурою

Виконання комплексу вправ для зняття зорової втоми (Варіант 1 або 2). Вправи проводить староста класу (групи) або його заступник.

Вправи виконуються сидячи в зручній позі, хребет прямий, очі відкриті, погляд — прямо, відвернувшись від комп'ютера.

Варіант 1

1. Погляд спрямовувати вліво-вправо, вправо-прямо, вгору-прямо, дотолу-прямо без затримки в кожному положенні.

Повторити 5 разів і 5 разів у зворотному напрямі.

2. Закрити очі на рахунок «раз-два», відкрити очі і подивитися на кінчик носа на рахунок «три-чотири».

3. Кругові рухи очей: до 5 кругів вліво і вправо.

Варіант 2

1. Швидко кліпати очима протягом 15 с.

2. Заплющити очі. Не відкриваючи очей, начебто подивитися ліворуч на рахунок раз-чотири», повернутися у вихідне положення. Так само подивитися праворуч на рахунок «п'ять-вісім» і повернутися у вихідне положення. Повторити 5 разів.

3. Спокійно посидіти із закритими очима, розслабившись протягом 5 с.

Інструктаж з правил техніки безпеки

Відновлення навичок набору тексту

Завантажити текстовий процесор Word. Набрати і відредагувати текст:

Модель (від лат. *modulus* — міра, зразок, норма) — це прообраз, опис або зображення якогось об'єкта.

Моделлю може бути будь-який об'єкт, установка, явище або мислений образ, за допомогою яких вивчаються складніші об'єкти.

Крім матеріальних моделей (іграшки, глобуса, макета будинку), існують абстрактні моделі: описи, формули, зображення, схеми, креслення, графіки тощо.

Абстрактні моделі, отримані внаслідок висновків, висловів називаються вербальними.

Математична модель — це модель, що описує об'єкт, явище чи процес мовою математики.

Інформаційна модель — це модель, що описує інформаційні процеси або містить інформацію про властивості і стан об'єктів, процесів, явищ.

Знакові моделі прийнято поділяти на математичні й інформаційні моделі.

Комп'ютерна модель — це модель (математична чи інформаційна), реалізована за допомогою програмних засобів.

Проведення комплексу вправ для зняття м'язового напруження

Вихідне положення — сидячи на стільці.

1. Витягнути і розчепірити пальці так, щоб відчувати напруження. У такому положенні затримати протягом 5 с. Розслабити, а потім зігнути пальці. Повторити вправу 5 разів.

2. Повільно і плавно опустити підборіддя, залишатися у такому положенні 2–3 с, і розслабитися.

3. Сидячи на стільці, піднести руки якомога вище, потім плавно опустити їх дотолу, розслабити. Вправу повторити 5 разів.

4. Переплести пальці рук і покласти їх за голову. Звести лопатки, залишатися у такому положенні 5 с, а потім розслабитися. Повторити вправу 5 разів.

Перевірка швидкості набору

Використовується клавіатурний тренажер **Stamina** (можна скачати із сайту: stamina.ru) або текстовий редактор Microsoft Word. Текст набирається протягом 10-ти хвилин. Для визначення швидкості набору кількість набраних символів (Сервіс\Статистика) ділимо на 10. Завдання для набору тексту — можна використати текст навчального матеріалу за курс 10-го класу.

V. Узагальнення й осмислення набутих знань

Бесіда з елементами опитування.

1. Що таке модель? Наведіть приклади моделей.
2. Що розуміється під знаковою моделлю?
3. Що таке математична й інформаційна модель? У чому різниця між ними?
4. Що таке комп'ютерна модель?
5. Які унікальні можливості дає комп'ютерне моделювання?
6. Назвіть основні етапи створення комп'ютерної моделі.
7. Що таке комп'ютерні експерименти?
8. Назвіть тип програмного забезпечення, яке використовується для реалізації комп'ютерних моделей.
9. На які типи умовно можна поділити моделі?
10. Назвіть недоліки комп'ютерних моделей.

VII. Підсумок уроку. Оголошення домашнього завдання

ОСОБЛИВОСТІ УРОКУ ІНФОРМАТИКИ З РОЗВИТКУ КРИТИЧНОГО МИСЛЕННЯ

Шолом Ганна Іванівна,

*учитель інформатики ЗОШ I–III ступенів №20 м. Чернігова,
учитель вищої категорії, учитель-методист.*



У сучасних умовах, коли обсяг необхідних для людини знань швидко збільшується, уже неможливо робити головну ставку на засвоєння певної суми фактів. Важливо навчити самостійно поповнювати свої знання, орієнтуватися в стрімкому потоці даних. ХХІ століття пред'являє до будь-якого фахівця вимоги швидко перебудовуватися з одного виду діяльності на інший, ефективно самонавчатися, творчо підходити до справи. Отже, якість освіти визначається здатністю людини стати успішним учасником розвитку суспільства.

Аналіз змісту навчального матеріалу, який передбачається вивчати за чинними програмами, свідчить про те, що учні на уроках мають незначну кількість часу для опанування сучасної системи знань з інформатики. Зрозуміло, що за таку невелику кількість годин неможливо надати учням усі важливі знання, тому необхідно навчити їх самостійно опановувати нові технології і критично переосмислювати дані, що вони отримують. Таке навчання передбачає розвиток навичок ХХІ століття, у тому числі й критичного мислення. Збільшення часу на вивчення окремих тем, з одного боку, є неможливим, а, з іншого, — недоцільним, то ж потрібно відмовитися від застарілих підходів репродуктивного навчання, стимулювати, спрямовувати, керувати пізнавальною активністю учнів, навчити критично переосмислювати здобуті знання.

Під критичним мисленням слід розуміти особливий вид розумової діяльності, характерними ознаками якого є: вироблення стратегій прийняття правильних рішень у розв'язанні будь-яких завдань на основі здобуття, аналізу, опрацювання відомостей; здійснення рефлексивних дій (аналітичних, перевірочних, контролюючих, оцінних), які виконуються стосовно будь-якого об'єкта чи явища, зокрема, і власного процесу мислення; виважений аналіз різних думок і поглядів, вияв власної позиції, об'єктивне оцінювання процесу і результатів як своєї, так і сторонньої діяльності.

Уміння критичного мислення учнів можна згрупувати у три основні групи:

- інформатичні компетентності;
- рефлексивні компетентності;
- дослідницькі компетентності.

Основним у групі інформатичних компетентностей є вміння працювати з даними, використовувати їх для розв'язання поставлених завдань і проблем як у професійній сфері, так і в повсякденному житті, для набуття нових знань і подальшої освіти, розвитку власних можливостей. Одними з основних проявів рефлексії мислення є самоконтроль й самооцінка діяльності, усвідомлення засад і способів виконання діяльності, контролювання, регулювання дій і операцій. До дослідницьких компетентностей можна віднести такі якості: здатність до аналізу, синтезу, узагальнення, формалізації; виявлення і формулювання

проблеми, задачі; здатність до прогнозування, моделювання ситуації й діяльності, експерименту.

Формування критичного мислення в старшокласників відбувається успішно в разі дотримання таких умов:

- урахування специфічних особливостей викладання предмету інформатики;
- створення інноваційно-рефлексивного середовища;
- поєднання діагностики і самодіагностики стану сформованості критичного мислення старшокласників.

Учні, які навчаються критично мислити, будуть здатні:

- аналізувати проблеми з різних ракурсів, використовуючи наразі дані різних джерел;
- навчатися самостійно протягом життя і водночас не будуть обмежені відомостями з підручників, які дуже швидко застарівають і знаннями вчителів і викладачів;
- слідкувати за бурхливим розвитком знань з інформатики;
- задовольняти власні інтелектуальні і творчі потреби;
- мати можливість будувати власні траєкторії навчання;
- моделювати нові інструменти пізнання і практичної діяльності.

Технологія розвитку критичного мислення відноситься до типу рамкових. Своєрідною рамкою, до якої вписується урок, є так звана базова модель технології, що складається з трьох етапів (стадій): стадії викику, смислової стадії і стадії рефлексії.

Базова модель («Викик — Осмислення — Рефлексія») задає не тільки певну логіку побудови уроку, але й послідовність, і способи поєднання конкретних методичних прийомів. Структура технології струнка і логічна.

Кожна стадія має свої цілі і завдання, а також набір характерних прийомів, спрямованих спочатку на активізацію дослідницької, творчої діяльності, а потім на осмислення й узагальнення здобутих знань.

Перша стадія — «викик», під час якої в учнів активізуються знання, які вони здобули раніше, прокидається інтерес до теми, визначаються цілі вивчення майбутнього навчального матеріалу. Цей етап виконує такі функції:

- мотиваційну — спонукання до роботи з новими даними, спонукання інтересу до теми;
- інформаційну — виклик на «поверхню» наявних знань з теми;
- комунікаційну — безконфліктний обмін думками.

Друга стадія — «осмислення» — змістова, у ході якої відбувається безпосередня робота учня з текстом, причому вона направлена, осмислена. Виконує функції:

- інформаційну — одержання нових відомостей з теми;
- систематизаційну — класифікація отриманих повідомлень з теми.

Третя стадія — стадія «рефлексії» — фаза уроку, протягом якої учні роблять огляд ідей, що були відкриті ними, значення яких вони усвідомили. Рефлексія реалізує такі функції:

- комунікаційну — обмін думками про нові знання;
- інформаційну — придбання нових знань;
- мотиваційну — спонукання до подальшого розширення інформаційного поля;
- оцінювальну — спонукання до подальшого розширення інформаційного поля.

План уроку інформатики, на якому відбувається розвиток критичного мислення, я будую у вигляді технологічних схем, зразки яких представлені у додатку 1.

У процесі експерименту на уроках інформатики було встановлено, що на кожному етапі уроку з розвитку критичного мислення доцільно використовувати певні прийоми розвитку критичного мислення.

Так на стадії виклику, на мою думку, доцільно використовувати такі прийоми розвитку критичного мислення, як «Мозковий штурм», «Кошик» ідей, понять, імен..., «Дерево знань», «Перевернені логічні ланцюги», «Переплутані логічні ланцюжки», «Термін тричі», «Вільний лист».

Завдання етапу осмислення: одержання нових даних з теми, що розглядається. На цьому етапі відбувається класифікація отриманих відомостей з категорій знань. Цьому сприяють методи і прийоми: прийом «Інsert» («Маркувальна таблиця»), «ЗДХ» (знаю, довідався, хочу довідатися), «Подвійний щоденник».

Завдання етапу рефлексії: допомогти учням самостійно узагальнити вивчений матеріал і самостійно визначити напрямки подальшого вивчення матеріалу. На стадії рефлексії можна запропонувати старшо-

класникам такі прийоми: «Кола Венна», «Синквейн» («Сенкан»), «Написання есе».

Деякі прийоми можуть бути використані на будь-якому з етапів кроку. До них можна віднести: «Розбивка на кластери» («Групування»), «Бортовий журнал», «Трьохстовпчастий щоденник» (Концептуальна таблиця), «Сюжетні таблиці», «Візуальні форми», «Дискусійна сітка Алверманна».

Мною розроблені і використовуються на уроках інформатики робочі зошити з розвитку критичного мислення для учнів 9–11-их класів, зразок одного з уроків якого представлено в додатку 2.

Застосовуючи запропоновані прийоми розвитку критичного мислення на уроках інформатики, спостерігається помітне зростання якості навчання школярів за рахунок усвідомленості і самостійності мислення, що помітно позначається на глибині мислення. У цілому ж розвиток якостей критичного мислення і зростання якості навчання школярів дозволяє підвищити якість знань, тобто рівень засвоєння теоретичного матеріалу, його осмислення і розуміння, а також уміння розв'язувати не тільки стандартні завдання, але і вправи в зміненій формі.

Додаток 1

Тема: Текстовий процесор

Урок 1

Тема уроку: Поняття про шаблон документа; створення документа за допомогою майстра

Мета уроку:

Навчальна: познайомити учнів з поняттям шаблону документа. Навчити створювати документ за допомогою майстра.

Розвивальна: розвивати пізнавальний інтерес учнів, уміння застосовувати отримані знання на практиці.

Виховна: підвищити рівень інформаційної культури учнів.

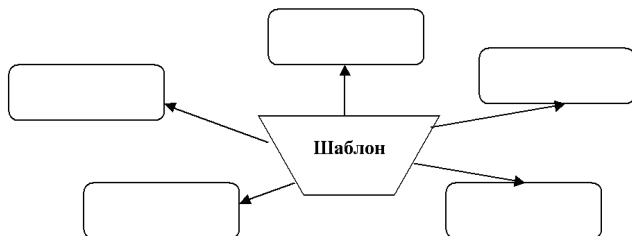
Етапи уроку	Цілі етапу	Хід уроку		
		Дяльність учителя	Діяльність учня	
1. Етап «Виклик»	Виявлення наявних знань	Які відомості з даного питання ви знаєте?	Заповнення індивідуальної таблиці «Знаю»	
		Обміняйтесь думками з товаришем по парті	Обмін знаннями	
		Давайте обговоримо	Заповнення загальної таблиці «Знаю»	
2. Етап «Осмислення»	Актуалізація попередніх знань та мотивація навчання	Прийомам «Кошик понять» пропонується розшифрувати поняття «Шаблон»	Учні заповнюють у власних зошитах схему	
		Записуються на дошці всі думки учнів		
		Кожному учню пропонується вивчити текст за даною темою і зробити помітки: V — знаю; — — не знаю; ? — хочу дізнатися; + — це для мене нове		Читають текст і роблять помітки
		Продовжіть заповнення таблиці «Хочу дізнатися» і «Дізнався нове»		Заповнюють індивідуальні таблиці за колонками «Хочу дізнатися» і «Дізнався нове»
		Давайте обговоримо «Хочу знати» і заповнимо на дошці цю колонку		Обговорення
Давайте обговоримо «Дізнався нове» і заповнимо на дошці цю колонку	Обговорення			
3. Етап «Рефлексії»	Контроль за якістю проведення заняття	В ході обговорення дається установка на домашнє завдання, а також виявляються бажані підготуватися і виступити на наступному занятті з запитань, що містяться в колонці «Хочу дізнатися»	Вибирають тему для повідомлення з питань, що цікавлять	
		За допомогою запитань або методу «Синквейн» з'ясувати результативність заняття з точки зору учнів	Висловлюють свою оцінку результативності заняття, відповідаючи на запитання або зачитуючи, отримані «піраміди знань»	
	Закріплення знань основних питань з даної теми	Вказати основні поняття вивченої теми		

Додаток 2

Тема: Поняття про шаблон текстового документа

1. Прийом «Кошик понять»

У повсякденному житті всі ви зустрічалися з поняттям «шаблон». Вам пропонується до поданих нижче клітинок записати ті асоціативні поняття, які у вас виникають, коли ви чуєте слово «шаблон».



2. Прийом «ЗДХ»

Протягом уроку вам пропонується занотовувати до поданої нижче таблиці факти, які вам були відомі з даної теми; факти, що стали для вас новими на уроці й те, чого, на вашу думку, вам не вистачило до повного розуміння даного матеріалу.

Знаю	Дізнався	Хочу дізнатися

3. Прийом «Логічні ланцюги»

Сьогодні на уроці ви дізналися, як можна створити певні документи. Для того, щоб перевірити свої знання, спробуйте відновити послідовність дій, необхідних для створення шаблону документа. У рядок відповідь впишіть по черзі номери дій, які потрібно виконати, щоб досягти бажаного результату.

Набір дій:

1. Створення за допомогою шаблону.
2. Новий документ.
3. Файл.
4. Вибрати вкладку, яка відповідає типу документа, який треба створити.
5. Створити.
6. Загальні шаблони.
7. Шаблон.
8. Створити.

Відповідь:

4. Метод «Синквейн»

Для стислого узагальнення знань про вивчені на уроці поняття вам пропонується написати «вірш» за певними правилами. Цей вірш має чітку структуру, складається з 5 рядків і відкидає будь-яку другоряд-

ну інформацію. Його структура така: у першому рядку вам пропонується одне слово-іменник. У другому рядку опишіть це поняття двома прикметниками, у третьому рядку передайте дію іменника трьома дієсловами, у четвертому запишіть речення із 4 слів, яке характеризує цей іменник, у п'ятому рядку запишіть іменник — синонім даному.

1.

Шаблон

2.

--	--
3.

--	--	--
4.

--	--	--	--
5.

--

Література

1. *Выготский Л.С.* Педагогическая психология / [под ред. В.В. Давыдова]. — М.: Педагогика, 1991. — 480 с.
2. *Выготский Л.С.* Собрание сочинений / Л.Выготский. — М.: Педагогика, 1983. — (Сочинения: в 6 кн. / Л.Выготский; кн. 3).
3. *Гальперин П.Я.* Формирование знаний и умений на основе теории поэтапного усвоения умственных действий : сб. Статей / [под ред. П.Я. Гальперина, Н.Ф. Талызиной]. — М.: Изд. Моск. ун-та, 1968. — 135 с.
4. *Дженні Л.* Методична система «Розвиток критичного мислення у навчанні різних предметів» / Дженні Л., Куртіс С., Темпл Ч. — Ч. I–IV, 1998.
5. *Кузьмінська О.Г.* Розвиток інтелектуальної активності ліцеїстів у процесі навчання інформатики : дис. ... канд. пед. наук : 13.00.02 / Олена Геронтівна Кузьмінська. — К.: Національний педагогічний університет імені М. П. Драгоманова, 2008. — 261 с.
6. Майбутнє шкільної інформатики. Тенденції розвитку освітніх інформаційно-комунікативних технологій [Електронний ресурс] / Співаковський О.В. — Режим доступу: http://pcti-ketrin.blogspot.com/2010/04/blog-post_09.html.
7. *Морзе Н.В.* Основи інформаційно-комунікаційних технологій / Наталія Вікторівна Морзе. — К.: Видавнича група ВНУ, 2006. — 352 с.
8. *Морзе Н.В.* Методика навчання інформатики : навч. посіб. : [у 3 ч.] / [за ред. акад. М.І. Жалдака]. — К.: Навчальна книга, 2004. — Ч. 1: Загальна методика навчання інформатики. — 2004. — 256 с.
9. Національна доктрина розвитку освіти України у XXI столітті. — К.: Шкільний світ, 2001. — С. 4
10. *Тягло А.В.* Критическое мышление: Проблема мирового образования XXI века / Тягло А. В., Воропай Т. С. — Х.: Ун-т внутр. дел, 1999. — 285 с.
11. *Шолом Г.І.* Використання інноваційних методів навчання на уроках інформатики / Г.І. Шолом // Комп'ютер у школі та сім'ї. — 2011.— №5(93).— С.11–13.
12. *Lipman M.* Critical thinking: what can it be? — N.J.: Institute for Critical Thinking, 1998.
13. *Paul R.W.* (1990). Critical Thinking: what every person needs to survive in a rapidly changing world. Rohnert Park, CA: Center for Critical Thinking and Moral Critique, Sonoma State Univ.



ОРГАНІЗАЦІЯ РОБОТИ З ОБДАРОВАНИМИ ДІТЬМИ НА УРОКАХ ІНФОРМАТИКИ

Романчук Тетяна Миколаївна,

учитель інформатики I категорії Чернівецької гімназії №7.



Сьогодні у нашому сучасному суспільстві, яке характеризується стрімкими змінами в різних сферах життя: політичній, економічній, науковій і культурно-мистецькій, поряд з освіченою та фаховою компетентністю особливого значення набувають уміння людини самостійно мислити, висувати нестандартні ідеї та прогнозувати, виявляти творчий підхід у будь-якій діяльності. На зміну старій уніфікованій школі, зорієнтованій на середнього учня, приходить нова система альтернативної освіти, спрямована на ефективний розвиток обдарованих дітей.

Обдаровані діти — майбутній цвіт нації, інтелектуальна еліта, гордість і честь України, її світовий авторитет. Тому перед кожним учителем стоїть завдання, спрямоване на забезпечення формування інтелектуального потенціалу нації шляхом створення оптимальних умов для всебічно обдарованої молоді. Адже обдаровані люди приносять велику користь суспільству, державі, є її гордістю. Тому перед суспільством, державою, школою і сім'єю постає проблема навчання і виховання таких дітей. Одним із важливих напрямків діяльності вчителя є його робота з обдарованими учнями. Вони характеризуються порівняно високим розвитком мислення, довготривалим запам'ятовуванням навчального матеріалу, великою працездатністю, умінням самостійно працювати. Їм притаманна свобода висловлювання думки, багатство уяви, чіткість різних видів пам'яті, швидкість реакції, уміння піддавати сумніву й науковому осмисленню певні явища, стереотипи.

Обдарованість дитини — це поєднання трьох характеристик (рис. 1).

Працюючи над проблемою «Організація роботи з обдарованими дітьми», учитель повинен ставити перед собою таку мету:

- зробити уроки інформатики справжньою школою духовності та творчості;
 - налагодити тісну співпрацю між учителем і учнем;
 - навчити учнів самостійно здобувати знання, творчо мислити і без страху підходити до вирішення будь-якої життєвої ситуації;
 - формувати творчу особистість, розвивати індивідуальні здібності і талант;
 - виховати національно свідомого, освіченого, цивілізованого громадянина України.
- Завдання:
- створити в класі творчу атмосферу;
 - виявити здібності учнів, розвивати їх якомога повніше;



Рис. 1

- пробудити пізнавальну діяльність у школярів;
- зробити вивчення кожної теми чимось близьким для учня, таким, що викликає роздуми, почуття;
- формувати пріоритети і вміння ставити мету і досягати її;
- формувати вміння правильно й образно висловлювати власну думку;
- навчити дитину бачити прекрасне, тонко сприймати навколишній світ.

Для втілення даної мети та завдань виділяю 4 етапи (рис. 2).

Для визначення типу обдарованості можна скористатися методикою Клименка, Хаана і Кафа, різноманітним тестуванням та анкетуванням.

Формами роботи можуть бути групові та індивідуальні заняття на уроках і в позаурочний час, факультативи. Зміст навчальної інформації має доповнюватися науковими відомостями, які можуть одержати в процесі виконання додаткових завдань у той же час, що й інші учні, але за рахунок вищого темпу обробки навчальної інформації.

Серед методів навчання обдарованих учнів мають переважувати пошуковий і дослідницький підходи до засвоєння знань, умінь і навичок, самостійна робота, метод проєктів. Контроль за їх навчанням повинен стимулювати поглиблене вивчення, систематизацію, класифікацію навчального матеріалу, перенесення знань у нові ситуації, розвиток творчих елементів у їх навчанні. Домашні завдання повинні мати творчий, диференційований характер.

Вищеперелічені аспекти, які мають бути органічно вплетеними в урок, доповнюються системою позакласної та позашкільної роботи: виконання учнем позанавчальних завдань; заняття у наукових товариствах; відвідування гуртка або участь у тематичних масових заходах (вечорах любителів інформатики, історії, фізики, математики та ін.); огляди-конкурси художньої, технічної та інших видів творчості, зустрічі з ученими тощо.

Індивідуальні форми позакласної роботи передбачають виконання різноманітних завдань, участь в очних і заочних олімпіадах, конкурсах на кращу науково-дослідну роботу. Вчителі повинні послідовно стежити за розвитком інтересів і нахилів учнів, допомагати їм в обранні профілю позашкільних занять.

Помітна роль у розвитку інтелектуально обдарованих дітей належить Малій академії наук України, її територіальним відділенням.

Для того, щоб робота з обдарованими дітьми була більш результативною, потрібно співпрацювати з батьками, при цьому здібності будуть розвиватися як у школі, так і вдома. Для цього необхідно з батьками обдарованої дитини проводити індивідуальні бесіди, залучати їх до проведення учнівських конференцій, захистів проєктів, презентацій, Web-сайтів тощо. Адже

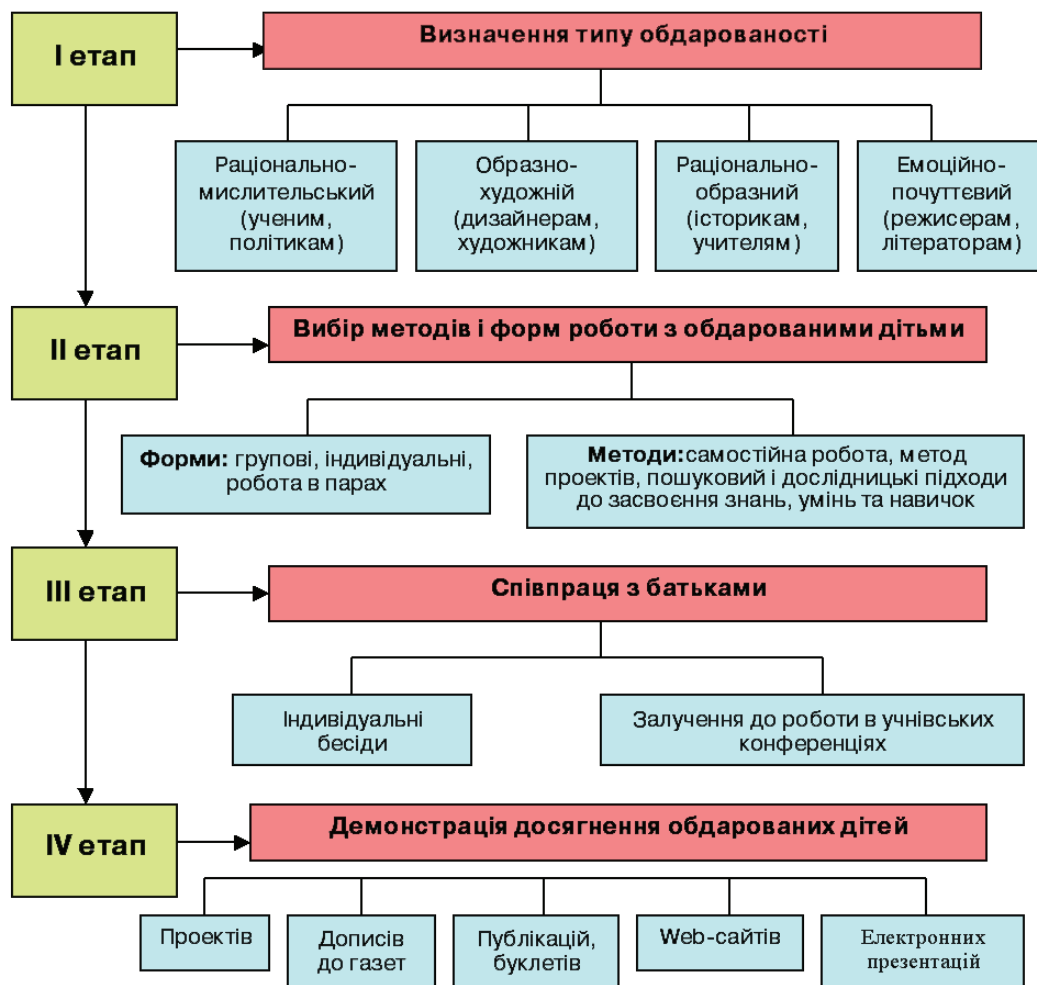


Рис. 2

батькам буде приємно спостерігати, як їхня дитина демонструє свої знання, уміння, кмітливість.

Звичайно, одним із головних етапів роботи з обдарованими дітьми є демонстрація їхніх робіт. Адже для кожної людини важливо, щоб її праця дала плоди, щоб про результати її праці знали однокласники, учителі-предметники.

На своїх уроках у вивченні тем «Комп'ютерні презентації», «Створення друкованих видань та їх дизайн», «Мова HTML і WEB-дизайн» я реалізую метод проектів. Учні створюють проекти у вигляді електронних презентацій, публікацій, буклетів, web-сайтів на різні теми, які підбираю їм я або вони самі. Дана робота реалізує не тільки метод проектів, але й учні вчаться працювати самостійно, виробляють вміння працювати з різними джерелами інформації, виділяти із знайденої інформації потрібну, а також компонувати її. У свою чергу вчитель виступає консультантом, який направляє дітей у правильне русло.

Кожна людина індивідуальна й обдарованість відбиває її характер, схильність до чогось чи захопленість чимось. Але розвиток таланту залежить від бажання постійно працювати й удосконалюватися в одному з напрямків науки.

Кожна людина народжується талановитою, а чи розвине вона свій талант обдарованості залежить від обставин, у яких вона буде жити і вчитися і від її бажання, бо талант можна розвинути і «закопати». Існує така біблійська притча «Батько дав по одному «таланту» (грошова одиниця в той час) своїм синам, і через пев-

ний час він прийшов до своїх дітей. Один син, якому батько подарував «талант», використав на розведення господарства. Другий — сказав батькові: «Я закопав свій «талант» так глибоко, що ніхто його не знайде» Коли батько це почув, він гірко заплакав». Із тих пір існує вислів «закопати талант», тобто не розвинути його, а знищити. Тому наше завдання полягає в тому, щоб створити таку ситуацію, у якій би наші учні свій талант не закопували, а розвивали.

Отже, можна сказати, що одне з головних завдань вчителя у роботі з обдарованими учнями — це побачити в кожному з них індивідуальність. При цьому слід мати на увазі, що сама обдарованість — індивідуальна. В одних випадках вона, як весняна квітка, що рано викликає подив і милування. В інших — вона захована у внутрішньому світі дитини, закрита різними комплексами, переживаннями, особливостями темпераменту і характеру.

Підтримати обдарованість, помітити її, відкрити, допомогти розкритися — ось головне завдання вчителя.

Література

1. Аніскіна Н. Педагогічна підтримка обдарованості. — К.: Шкільний світ, 2005.
2. Максименко М. Робота з обдарованими учнями в умовах використання ІКТ у гімназії. — К.: Інформатика, 2008. — №5. — С. 16–22.
3. Розум Л. Робота з обдарованими учнями на уроках інформатики. — К.: Інформатика, 2004. — №3. — С. 6–7.
4. Сухомлинський В.О. Вибрані твори. В 5-ти томах. Том 5. Статті. — К.: «Радянська школа», 1977, — 639 с.
5. Карпенко З.С. Експресивна психотехніка для дітей. — К.: НПЦ Перспектива, 1997. — 96 с.

ФОРМУВАННЯ УМІНЬ ПРОДУКТИВНО-ТЕХНОЛОГІЧНОЇ ДІЯЛЬНОСТІ УЧНІВ НА УРОКАХ ІНФОРМАТИКИ

Гетманцев Василь Сергійович,

вчитель інформатики вищої категорії, старший учитель
Миколаївського муніципального колегіуму, м. Миколаїв.



Предмет «Інформатика» належить до освітньої галузі «Технологія» і пріоритетною складовою змісту освіти є елемент, що формує інформаційно-технологічні вміння, тобто вміння кваліфікованого й ефективного використання сучасних інформаційно-комунікаційних технологій у навчально-пізнавальній діяльності й повсякденному житті.

У сучасній інформаційній освіті на перше місце виходить технологічний аспект — використання інформаційних технологій. Це вимагає відповідного сучасного рівня методичного забезпечення уроку інформатики. До того ж постійне оновлення фактичного матеріалу потребує від учителя багато часу на його опанування, а традиційна методика, яка існує зараз під час передавання знань (пояснення нового матеріалу, відтворення його учнями, постановка і розв'язок задач, контроль знань і вмінь учнів), не дає достатньої і раціональної, з методичної точки зору, можливості вчителю інформатики викладати теми розділу «Інформаційні технології». Ця проблема породжує необхідність впровадження істотно нової методики навчання інформаційних технологій, яка б робила акцент саме на технологічну складову. Мету навчання інформаційних технологій потрібно розуміти як здатність створювати інформаційні продукти за допомогою комп'ютера. Поширена на даний час методика навчання не розглядається з точки зору створення саме інформаційного продукту, вона спрямовує учнів на опанування універсальними вміннями користувача для вивчення різних програмних продуктів.

У шкільному курсі інформатики можна виділити основні змістові лінії:

- **теоретична інформатика;**
- **алгоритмізація та програмування;**
- **інформаційні технології.**

Під час викладання тем змістової лінії «Інформаційні технології» у значної частини вчителів, так само, як і в моїй практиці, виникають подібні проблеми [1].

1. Проблема невизначеності цілей навчання інформаційних технологій.
2. Проблеми з викладанням змісту розділу: проблематичним залишається питання надмірності фактичного матеріалу порівняно з кількістю годин, які передбачені програмою; проблема співвідношення у формуванні теоретичних знань і практичних умінь учнів.
3. Проблема організації практичної роботи учнів у комп'ютерному класі.
4. Проблема оцінювання знань і вмінь учнів за цими темами.

Один із варіантів розв'язання зазначених проблем полягає в орієнтації вчителя інформатики на впровадження методики технологічного навчання, основою якої є перетворююча діяльність учня, націлена на виготовлення необхідного «інформаційного продукту» за допомогою роботи з відповідними «інструментами» — прикладними програмними засобами.

Ідея досвіду

Формування умінь продуктивно-технологічної діяльності учнів є результатом запровадження методики технологічного навчання під час викладання тем змістової лінії «Інформаційні технології». Ця методика до цього часу не набула практичного поширення в загальноосвітній школі, окрім уроків праці.

Уміння продуктивно-технологічної діяльності — це узгоджена сукупність дій, спрямованих на створення інформатичного продукту, які учень може свідомо здійснити на основі здобутих знань і сформованих навичок.

Складові методичної системи технологічного навчання інформатики

Мета: формування інформаційно-технологічних умінь створення інформаційних продуктів за допомогою комп'ютера.

Зміст: інформаційно-технологічні знання, уміння та навички.

Методичні підходи

- **Формально-операційний.** Метою навчання за даного підходу є ознайомлення учнів з функціональними можливостями програмного забезпечення й алгоритмом виконання простих операцій. За формально-операційного підходу йде відпрацювання навичок роботи, коли учні не бачать кінцевого результату своєї діяльності, а лише проробляють технологію поелементно.
- **Задачно-інструктивний.** Метою навчання є формування інформаційно-технологічних умінь під час створення інформатичного продукту за поданим зразком й описаною технологією. За задачно-інструктивного підходу вчитель пояснює учням технологію роботи і спрямовує їх на кінцевий результат (результат має бути відомий як обов'язковий елемент). У застосуванні задачно-інструктивного підходу має чітко простежуватися технологія створення інформатичного продукту.

Поєднання методичних підходів на уроках технологічного навчання інформатики різних типів

Дидактична мета уроку	Поєднання методичних підходів
Вивчення нового матеріалу	Формально-операційний . Задачно-інструктивний
Закріплення знань і вмінь	Задачно-інструктивний. Задачно-технологічний
Перевірка засвоєння знань і вмінь вивчення нового матеріалу, формування вмінь і навичок (комбінований тип уроку)	Задачно-інструктивний. Задачно-технологічний. Проблемний
Контроль і корекція знань, вмінь, навичок	Задачно-інструктивний. Задачно-технологічний. Проблемний

• **Задачно-технологічний. Мета** підходу — сформулювати інформаційно-технологічні уміння створення інформаційного продукту за наданими вимогами. Підхід характеризується наявністю в учнів мети (вони бачать, яким має бути кінцевий результат), прагненням досягти бажаного результату, використовуючи обрану технологію. За цього підходу учні створюють інформаційний продукт згідно з проектом, планом.

• **Проблемний. Мета** підходу — розвинути проєктувальні й творчі можливості учня, а також здатність застосовувати інформаційно-технологічні уміння за нових умов. За проблемного підходу, як правило, постає задача з неявно заданою умовою (структурою). Учні потрібно самому скласти структуру даних і розв'язати задачу, застосовуючи певну технологію.

Форми навчання: практичні, групові та індивідуальні самостійні роботи.

Засоби навчання: комп'ютерні прикладні програмні засоби, презентації, відеоуроки тощо.

Результат навчання: сформованість системи інформаційно-технологічних умінь старшокласників як основи їхньої інформаційно-комунікаційної компетентності.

Наукові і практичні дослідження

Педагогічні основи технологічної освіти обґрунтовані у роботі М.Б. Павлової [2]. Проблеми підготовки вчителів до впровадження нових інформаційних технологій навчання розглядали В.Ю. Биков, Р.С. Гуревич, М.І. Жалдак [3], Н.В. Морзе [4]. Методиці технологічного навчання інформатики присвячене дослідження Ю.О. Дорошенка, Т.В. Тихонової, Г.С. Луньової «Технологічне навчання інформатики» [5], у якому наведено ґрунтовне методичне і дидактичне забезпечення викладання інформатики за цією методикою.

Алгоритм реалізації методики технологічного навчання

1. Визначити мету навчання теми як формування вмінь створення певних інформаційних продуктів або здійснення результативної діяльності.

2. Визначити зміст теми як сукупність:
- інформаційно-технологічних вмінь;
 - інформаційно-технологічних знань;
 - інформаційно-технологічних навичок.

3. Скласти структуру курсу (теми).

4. Для кожної теми курсу (певного уроку теми) розробити сукупність і послідовність практичних завдань, що формують інформаційно-технологічні вміння (від простих технологічних операцій до вмінь створення закінченого інформаційного продукту).

5. Розробити засоби оцінювання сформованості інформаційно-технологічних умінь (самостійні і практичні роботи, тестові завдання).

Апробація методики технологічного навчання

На фінальному етапі Всеукраїнського конкурсу «Учитель року 2013» у м. Красноград Харківської області під час проведення відкритого уроку у 9 класі було використано методику технологічного навчання. Тема уроку була визначена журі як «Основи робо-

ти з текстовою інформацією». Тип уроку — узагальнення і систематизація знань.

Задля організації самостійної роботи учнів їм було запропоновано два завдання за різними методичними підходами технологічного навчання: задачно-інструктивним і задачно-технологічним. Метою завдань було створення інформаційного продукту — відредагованого й відформатованого тексту і бейджу. Учні мали змогу самостійно вибрати спосіб виконання завдання й отримати відповідний бал.

Завдання №1. Редагування і форматування тексту

Учні отримують файл із текстом, що містить зайві пробіли між словами і перед розділовими знаками, розриви рядків, відсутнє форматування за стандартними вимогами.

Задачно-інструктивний підхід, максимальна оцінка 8 балів

Інструкція

1. Ввімкнути режим відображення недрукованих символів (рис. 1).

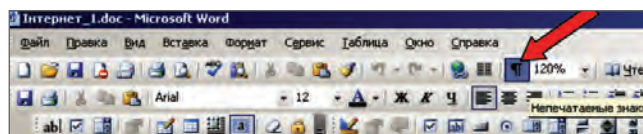


Рис. 1

2. Виконати команду **Правка\Замينить** (рис. 2). У поле **Найти** ввести два пробіли.

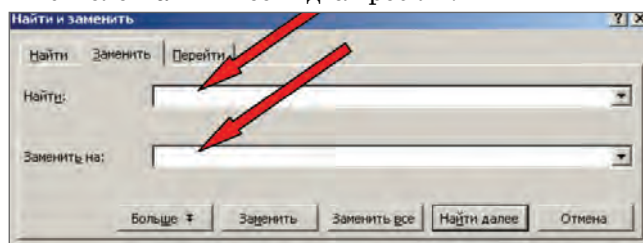


Рис. 2

У поле **Заминить на** ввести один пробіл.

3. Виконати команду **Правка\Заминить** (рис. 3). У поле **Найти** видалити два пробіли і ввести символи **пробіл+кома**.

У поле **Заминить на** видалити пробіл і ввести символ **коми**.

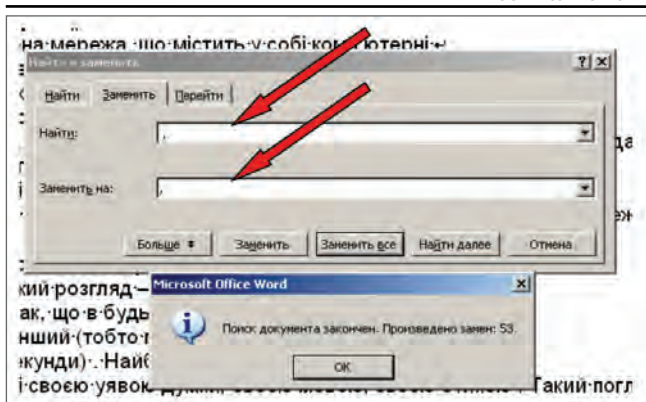


Рис. 3

Аналогічно замінити пробіл + крапка на крапку (рис. 4).

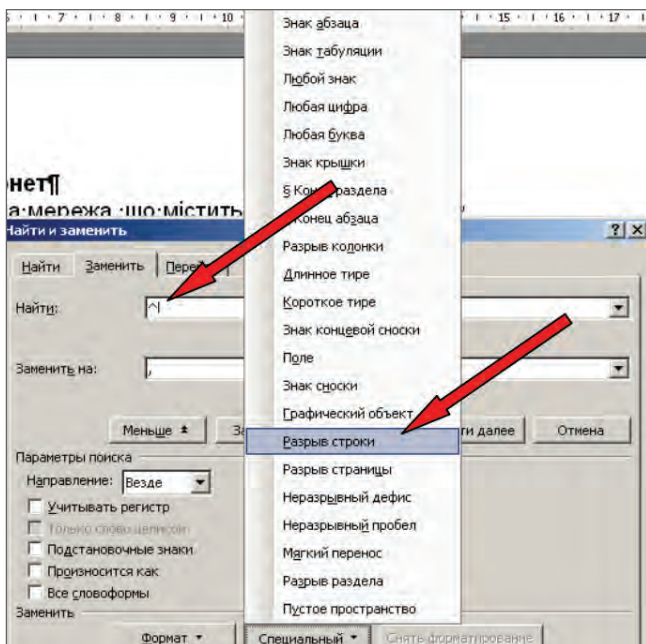


Рис. 4

4. Виконати команду **Правка\Заменить**.
У поле **Найти** видалити пробіл і крапку й увести символ **Разрыв строки**.

У поле **Заменить на** видалити крапку і залишити поле пустим.

5. Виділити весь текст і вирівняти по ширині.
Заголовки вирівняти по центру (рис. 5).

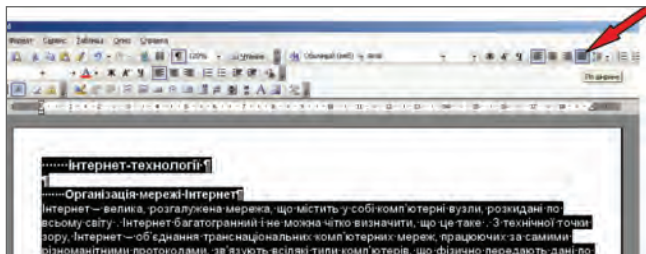


Рис. 5

6. Виділити весь текст і зробити відступ абзацу на 1,25 см (рис. 6).

7. Встановити параметри сторінки **Файл\Параметры страницы** (рис. 7).

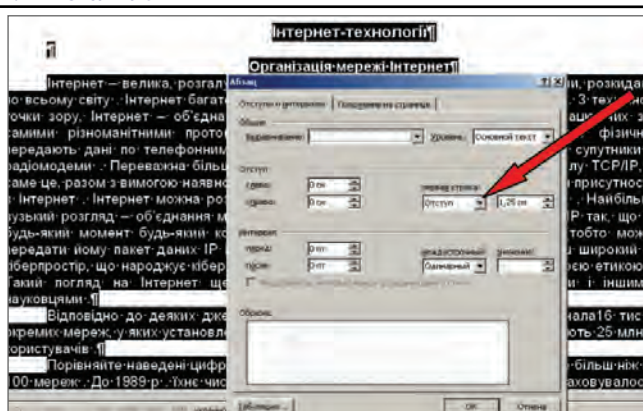


Рис. 6

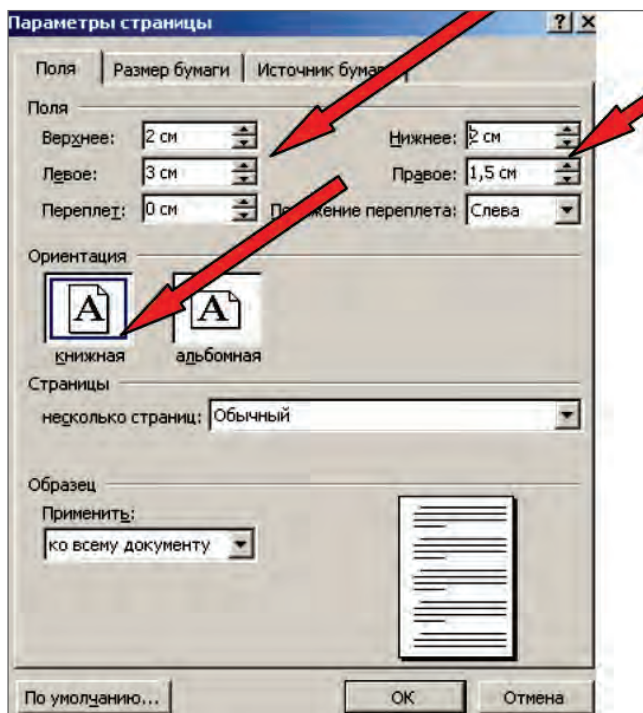


Рис. 7

8. Виділити весь текст і встановити шрифт Times New Roman, розмір 14 пт, міжрядковий інтервал — 1,5 (рис. 8).

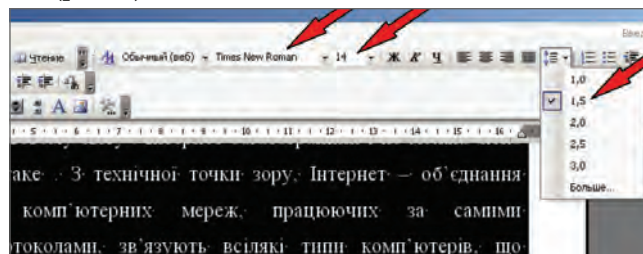


Рис. 8

Завдання №1. Редагування і форматування тексту
Задачно-технологічний підхід, максимальна оцінка 10 балів

Технологічні вимоги до оформлення документа

Параметри тексту:

- Слова відокремлюються одним пропуском — 1 бал.
- Перед розділовими знаками пропуск не ставиться — 1 бал.
- Вирівнювання тексту — по ширині сторінки — 1 бал.
- Вирівнювання заголовків — по центру сторінки — 1 бал.
- Відступ абзацу — на 1,25 см — 1 бал.

Параметри сторінки:

- Розмір — А4 — 1 бал.
- Орієнтація — книжкова — 1 бал.
- Поля: ліве — 3 см, праве — 1,5 см, верхнє та нижнє — 2 см — 1 бал.

Параметри шрифту:

- Шрифт Times New Roman, розмір 14 пт — 1 бал.
- Міжрядковий інтервал — 1,5 — 1 бал.

Завдання №2. Створення бейджу
Задачно-інструктивний підхід, максимальна оцінка 9 балів

Створити бейдж за зразком (рис. 9).



Рис. 9. Зразок бейджу

Інструкція з виконання

1. Виконати команду **Вставка\Надпись** (коли з'явиться поле для рисунка, натиснути **Esc**).

2. Задати формат рамки з контекстного меню (рис. 10).

3. Розмістити текст всередині об'єкта **Надпись** згідно зразку, вирівнювання по центру:

- назва школи — шрифт Century Gothic, розмір 10;
- прізвище, ім'я — шрифт Monotype Corsiva, розмір 28;
- клас — шрифт Arial, розмір 20.

4. Вставити картинку з хлопчиком (дівчинкою): **Вставка\Рисунок\Из файла**, викликати контекстне меню об'єкта **Формат рисунка\Положение\перед текстом**, зменшити розміри картинку і перемістити її на надпис (рис. 11).

5. Виділити, натиснувши **Shift**, надпис і картинку і згрупувати їх у один об'єкт (рис. 12).

Завдання №2. Створення бейджу
Задачно-технологічний підхід, максимальна оцінка 12 балів

Створити бейдж за зразком (див. рис. 9).

Технологічні вимоги до оформлення бейджу

1. Основою бейджу є об'єкт **Надпись**.

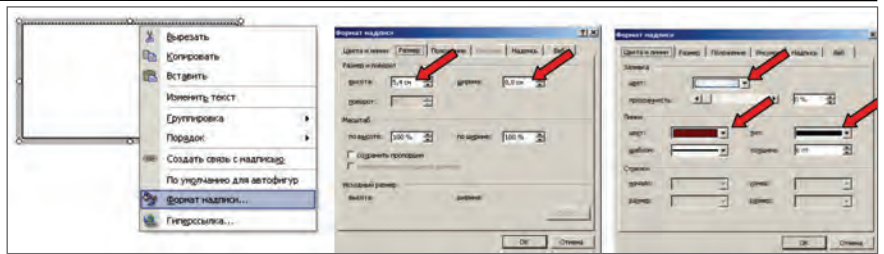


Рис. 10



Рис. 11

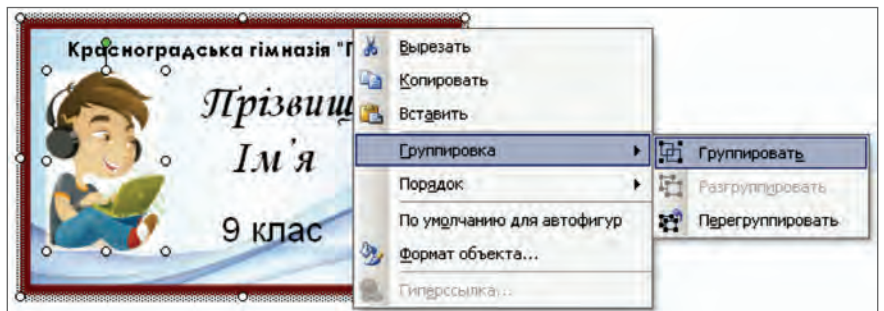


Рис. 12

2. Формат надпису:

- висота — 5,4 см,
 - ширина — 8,8 см,
 - товщина лінії — 6 пт,
 - заливка — рисунок **фон.jpg**.
3. Параметри тексту на бейджі:
- назва школи — шрифт Century Gothic, розмір 10;
 - прізвище, ім'я — шрифт Monotype Corsiva, розмір 28;
 - клас — шрифт Arial, розмір 20.

4. Надпис і картинка згруповані в один об'єкт.

Результати і висновки

За результатами самостійної роботи всі 12 учнів виконали обидва завдання за відведений час у 20 хвилин, наразі 8 учнів обрали задачно-інструктивний підхід до виконання завдання, 4 учні — задачно-технологічний підхід. Під час рефлексії всі учні зазначили, що перше завдання виявилось складнішим, бо важко було визначити ефективний алгоритм редагування тексту.

Успішне виконання всіма учнями практичних завдань на створення інформаційних продуктів за методикою технологічного навчання

під час відкритого уроку фінального етапу Всеукраїнського конкурсу «Учитель року 2013» свідчить про ефективність даної методики і можливість її впровадження без попередньої підготовки учнів на будь-якому етапі вивчення тем курсу змістової лінії «Інформаційні технології».

Література

1. Тихонова Т.В. Методика навчання учнів інформаційним технологіям на уроках інформатики / Т.В. Тихонова, Г.С. Луньова // Вересень. — 2003. — №1 (23). — С. 61–66.
2. Павлова М.Б. Образовательная область «Технология»: теоретические подходы и методические рекомендации / М.Б. Павлова, Дж. Питт. — Н. Новгород : Нижегородский гуманитарный центр, 1998. — 96 с.
3. Жалдак М.І. Методика вивчення основ інформатики та обчислювальної техніки у педагогічному вузі : навч. посібник / М.І. Жалдак. — К. : КДПІ, 1986. — 75 с.
4. Морзе Н.В. Методика навчання інформатики : навч. посібник у 3-х ч. / [Н.В. Морзе; за ред. М.І. Жалдака]. — К. : Навчальна книга, 2004.
5. Технологічне навчання інформатики : навчально-методичний посібник / Ю. О. Дорошенко, Т. В. Тихонова, Г. С. Луньова. — Х. : Вид-во «Ранок», 2011. — 304 с.

МАЙСТЕР-КЛАС НА ТЕМУ «ІНФОРМАЦІЙНА КОМПЕТЕНТНІСТЬ»

Кривко Наталія Валеріївна,

вчитель інформатики Полтавської гімназії №33.

На початку майстер-класу я попрошу вас звернути увагу на екран (*на екрані — зображення моря*).

Які асоціації у вас виникли? Відпочинок, сонце, чайки, море. Добре відпочити влітку на морі! Але ж море може таїти в собі небезпеку: морська безодня, морська стихія, морське цунамі. А ще буває **МОРЕ ІНФОРМАЦІЇ**, у якому можна легко потонути. Щоб це море було безпечним, для людини необхідно володіти інформаційною компетентністю.

І сьогодні ми зупинимося на деяких аспектах формування інформаційної компетентності школярів на уроках інформатики. Сподіваюся, що в ході майстер-класу ви подумки перенесете запропоновані мною методи й прийоми у свої предметні галузі і зможете використовувати їх на своїх уроках.

У структурі інформаційної компетентності (ІК) я виділяю:

1) мотиваційний компонент — він знаходить своє відображення в розвитку мотивації особистості на розв'язання інформаційних ускладнень, що виникають у процесі оволодіння способами роботи з інформацією;

2) об'єктний компонент — особистість володіє відповідними інформаційними знаннями й уміннями;

3) суб'єктний компонент — полягає в оцінюванні особистістю своєї власної інформаційної діяльності.

Як найбільш адекватні завданню формування інформаційної компетентності школярів я вважаю такі технології, як: **технологію кооперативного навчання** (робота парами і спілкування в групах); **технологію колективно-групового навчання** (обговорення проблеми в загальному колі, заохочення учнів до дискусії, кожен учить кожного); **технологію ситуативного моделювання** (ігрове моделювання явищ, які вивчаються); **технологію опрацювання дискусійних питань** (щоб кожний учень брав дієву участь в уроці, використовую різновид ігрових форм занять, коли з обговорюваної проблеми висловлюються всі учасники спільної діяльності, але керівна роль належить учителеві як організатору дискусії).

Лінгвісти знайомі з таким поняттям як **інтерференція** — це накладання системи однієї мови на систему іншої мови. Як приклад можна навести поповнення лексичного складу мови шляхом прямого запозичення — слово-«калька». Комп'ютерна термінологія має здебільшого англійське походження, причому способом калькування. Для формування навички роботи з текстами, що містять нову термінологію (у нашому випадку — комп'ютерну), ми застосовуємо прийом **оберненої інтерференції**: 1) утворити для терміна англійське «кальку»; 2) утворити від слова-«кальки» слово-«основу»; 3) знайти в словнику переклад і відібрати з наявних варіантів той, який найбільше підходить за контекстом.

Обернена інтерференція

Приклад: перед дітьми було завдання сформулювати визначення клієнта, орієнтуючись на переклад з англійської мови і спираючись на власний досвід або, наприклад, на інформацію з тлумачного словника.

Client (у перекладі з англ.) — клієнт; покупець, замовник; споживач.

Тлумачний словник дає таке означення: клієнт — споживач, який через певні обмеження змушений залежати від постачальника.

Відповідно учні сформулювали такі визначення.

Клієнт — комп'ютер (або програма), що використовує ресурси, надані іншим комп'ютером (або програмою).

Клієнт — апаратний або програмний компонент обчислювальної системи, який надсилає запити іншому комп'ютеру.

Порівняємо їх з визначенням, поданим у підручнику: клієнт — це абонент мережі, який лише використовує мережні ресурси, але свої ресурси в мережу не віддає.

Як бачимо, учні змогли самостійно сформулювати правильне визначення.

А тепер завдання для Вас: сформулюйте визначення серверу, спираючись на власний досвід або визначення, подане у словнику (*переклад і визначення на екрані*).

Перевірка виконання завдання.

За допомогою даного прийому я створюю умови для перенесення знань учнів з однієї предметної галузі (англійська мова) в іншу (інформатика), закріплюю навички пошуку інформації, а також синтезу нового знання, що складає суть об'єктного компонента ІК. У нашому випадку учасники працюють з інформацією, поданою на екрані, на уроці ж можна використати варіант друкованого, електронного або Інтернет-словника.

Конструювання поняття

Перш ніж приступити до розв'язання проблеми, необхідно сформулювати понятійний апарат, який у курсі інформатики є достатньо насиченим і великим за обсягом (прикладом може служити поняття «захист інформації»).

Метод евристичних запитань

Будь-яку проблему можна розглянути з боку таких запитань: Хто? (суб'єкт) Що? (об'єкт) Коли? (час) Для чого? (мета) Чим? (засіб) Як? (метод) Де? (місце).

Метод базується на принципах:

1) співвідношення проблемності й оптимальності проблеми;



- 2) подрібнення інформації;
- 3) цілеспрямованості.

Під час з'ясування суті проблеми вищевказані запитання можуть попарно комбінуватися. Діти працюють групами.

В організації роботи з формування понятійного апарату використовуємо метод конструювання поняття, запозичений з евристичного навчання й описаний Хуторським О.В. Формування поняття починається з актуалізації наявних в учнів знань і уявлень. У ході співставлення й обговорення вчитель допомагає побудувати їх до деяких культурних форм. Результатом роботи виступає творчий продукт — сформульоване визначення поняття. Одночасно можна познайомитися з іншими формулюваннями, наприклад, у підручнику й додатковій літературі, електронному підручнику, on-line словнику.

Наприклад, роботу над поняттям «захист інформації» починаємо з того, що згадуємо різні словосполучення зі словом «захист». Далі аналізуємо словосполучення за запропонованою схемою «хто? що? коли? ...».

Хто?	Суб'єкт	Власник та користувач інформації
Що?	Об'єкт	Інформація (дані і відомості, представлені в різних формах)
Для чого?	Мета	Захист від знищення, модифікації, блокування, розголошення, несанкціонованого доступу, витоку
Де?	Місце	В інформаційно-комунікаційних системах, у системах електронного документообігу
Чим?	Засіб	Антивіруси, фаєрволи, маршрутизатори, токени, смарт-карти, обмежуючі конструкції, охоронно-пожежна сигналізація тощо
Як?	Метод	Обмеження доступу до носія повідомлення апаратно-технічними засобами; попередження руйнування носія інформації; попередження доступу до інформації

Відповідно до розглянутої схеми учні формулюють таке визначення: захист інформації — це сукупність методів (обмеження доступу до носія повідомлення апаратно-технічними засобами; попередження руйнування носія інформації; попередження доступу до інформації) й засобів (антивіруси, фаєрволи, маршрутизатори, токени, смарт-карти, обмежуючі

конструкції, охоронно-пожежна сигналізація тощо), що забезпечують захист інформації від знищення, модифікації, блокування, розголошення, несанкціонованого доступу, витоку за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

Тепер Ваше завдання сконструювати поняття за розглянутою схемою. Спробуйте сформулювати поняття локальної комп'ютерної мережі, а потім порівняйте його з наведеним у підручнику.

Перевірка виконання завдання.

Як показує практика, учні формують не тільки визначення локальної комп'ютерної мережі, а й більшість ознак, поданих у підручнику.

Використовуючи цей метод, ми розвиваємо в учнів мотиваційний і об'єктний компоненти ІК.

Схеми фішбоун

У рамках проблемного навчання я використовую прийом складання схем фішбоун, який дозволяє організувати ефективну роботу з розв'язання ускладнень: уточнити проблему, виявити причини її виникнення, структурувати ключові факти за ступенем їх значущості. Вашій увазі пропонується схема (рис. 1), складена школярами в ході розв'язання проблеми «Вільно поширюване програмне забезпечення — це краще рішення?».

Наступне завдання: заповнимо схему фішбоун з розв'язання проблеми «Чи є Інтернет найбільш ефективним джерелом інформації?» (1 група виявляє факти «За», 2 група — «Проти»).

Для роботи учасникам можна запропонувати список властивостей або суцільний текст, також можливий варіант самостійного заповнення з опорою на особистий досвід або в ході організованої дискусії, у тому числі у формі участі у форумі, організованому самими учнями або відкритому в мережі.

Перевірка завдання, формулювання висновку з проблеми.

Шановні колеги, наш майстер-клас добігає кінця. Ключовим поняттям для нас стала «інформаційна компетентність». Для її формування нами були відібрані технології, найбільш адекватні завданню формування ІК і методи й прийоми в рамках цих технологій.



Рис. 1

ВИКЛАДАННЯ ОСНОВ РОБОТОТЕХНІКИ У 5–8 КЛАСАХ

Ігнатченко Костянтин Андрійович,
старший майстер відокремленого структурного
підрозділу НПУ і.м. М.П. Драгоманова
«Вище професійне училище»,

Ігнатченко Наталія Владиславівна,
вчитель інформатики спеціалізованої школи
з поглибленим вивченням інформаційних
технологій №52 м. Києва.

У спеціалізованій школі №52 вже декілька років існує й успішно втілюється в життя програма за вибором «Основи робототехніки». Можливо, у когось виникне запитання, а чому саме робототехніка? А чи не важко буде її пізнавати дітям? Чому? Тому, що роботи чи автоматизовані пристрої оточують нас усюди. Роботи будують транспорт (автопідприємства), роботи сортують пошту (автоматичні конвеєрні лінії), роботи навіть видають нам гроші (банкомати).

Важко? Можливо, але все залежить від форми навчання, від того, як подавати матеріал учневі.

У нашій школі в основу програми покладене навчання учнів роботі з конструкторами LEGO останнього покоління з використанням програмованих контролерів, різноманітних датчиків та серводвигунів. LEGO Mindstorms — це конструктор (набір деталей, що сполучаються, і електронних блоків) для створення програмованого робота (рис. 1).



Рис. 1

Уперше цей конструктор був представлений компанією LEGO в 1998 році на базі мікроконтролера RCX (рис. 2). Через 8 років (2006 р.) у світі з'явилася модель LEGO Mindstorms NXT вже на базі контролера NXT (рис. 3), а в 2009 р. виходить нова версія — LEGO Mindstorms NXT 2.0 (рис. 4).

Набори LEGO Mindstorms комплектуються набором стандартних деталей LEGO (балки, осі, колеса,

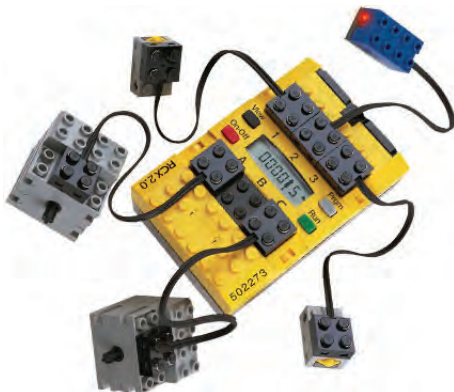


Рис. 2



шестерні) і набором, що складається із сенсорів, двигунів і програмованого блоку. Набори діляться на базовий набір і розширений. Базовий набір поставляється у двох версіях: версія для широкого продажу і базовий навчальний набір. Обидва набори можуть бути використані для участі в змаганнях з робототехніки.

LEGO втілює ідею модульності, що наочно демонструє дітям те, як можна розв'язувати деякі технічні проблеми, також розвиває навички збирання, ремонту і розбирання техніки.



Рис. 3

У 1980 році компанія LEGO заснувала окремий підрозділ LEGO Education, який розпочав свою діяльність у сфері освіти, співпрацюючи з відомими експертами у розвитку науки, технології, психології. За цей час було розроблено цілісну концепцію навчання для дітей, починаючи з півторарічного віку. Діяльність LEGO Education спрямована на стимулювання у дітей творчих навичок, створення проектних робіт, навчання співпраці у команді.

На сьогоднішній день програма LEGO Education впроваджена більш ніж у 40 країнах світу.

LEGO освіта спрямована на стимулювання у дітей творчих навичок, командної співпраці, створення проектних робіт, залучення дітей до участі в олімпіадах і конкурсах.

Програми LEGO Education — це практичний курс навчання дітей, «навчання через дію».

Курс робототехніки — це один із найцікавіших варіантів вивчення комп'ютерних технологій і програмування на уроках інформатики, на гуртках чи спецдисципліні. Об'єктом дослідження є робот, якого учні самі будують і програмують. Унікальністю робота є його



Рис. 4

комплектація потужним 32-розрядним процесором і сукупністю датчиків, сервомоторів та технічних деталей, що дозволяє учням під час занять почувати себе повноцінними інженерами-науковцями в невимушеній навчально-ігровій атмосфері.

Але незапрограмований робот — всього лиш купа деталей, нехай навіть складених у якусь конструкцію.

«Серцем» конструктора є програмований контролер NXT. Цей контролер має власний дисплей і 4 кнопки керування, що дозволяють запрограмувати його на виконання невеликої кількості базових функцій. Для повноцінної роботи цього, звісно ж, не достатньо, тому основний процес програмування відбувається за допомогою комп'ютера.

Контролер NXT використовує стандартну мову програмування, що дозволяє використовувати велику кількість різноманітних програмних середовищ, таких, як LabView, RobotC, RoboLab та багато інших. Але, на нашу думку, для знайомства зі світом роботів ізсередины найкраще підходить оптимізоване для роботи з дітьми програмне середовище LEGO Mindstorms Education NXT. Цей програмний продукт розроблений на основі промислової програми LabView, і має приємну візуальну оболонку та простий інтерфейс користувача, що дозволяє дитині швидко і легко освоїтись в роботі з цією програмою й оволодіти спершу основними, а в процесі роботи й усіма додатковими можливостями і функціями.

Інтерфейс програмного середовища LEGO Mindstorms Education NXT побудований на ланцюжках із програмних блоків, котрі зручно складати і переміщувати по програмному полю (рис. 5). Є різноманітні палітри таких блоків, що дозволяє програмувати виконання роботами найрізноманітніших функцій. До того ж, це програмне середовище дозволяє в режимі реального часу тримати зв'язок з контролером, що дозволяє швидко й зручно перевіряти результати своєї роботи, як контролюючи показники серводвигунів і датчиків у самій програмі, так й у виконанні програм чи окремих програмних блоків роботом.

Командна робота над проектами з робототехніки спонукає до глибокого вивчення як механіки складових сучасного робота, так і візуального програмування. Додатковим стимулом вивчення робототехніки є участь у щорічних Всеукраїнських і Світових олімпіадах з робототехніки.

Курс має кілька рівнів складності, що дозволяє дитині реалізувати свої знання, починаючи з простого й закінчуючи складним.

В Україні широкого розповсюдження робототехніка набула в 2008 році поряд з появою офіційної програми курсу за вибором «Основи робототехніки», затвердженою Міністерством освіти та науки України. Автор програми Д. Кожемяка. Метою цього курсу є:

- навчання основ об'єктно-орієнтованого і графічного програмування;
- підвищення мотивації учня до навчання програмування;
- розвиток творчих і розумових здібностей;
- розвиток конструкторського мислення у дітей.

Програма розрахована на учнів 5–8-их класів з розрахунку 1 година на тиждень (35 годин на рік).

Оскільки наша школа є спеціалізованою, з поглибленим вивченням інформаційних технологій, наше керівництво, зокрема директор школи Шевченко В.М. просто не могли оминати таку новинку.

Спочатку програма реалізовувалась лише у вигляді гуртка, а згодом запровадили й як обов'язковий курс, що викладається в 5-х класах. На відміну від основного предмету, гурток орієнтований для більш ширшої аудиторії — учнів 5–11 класів, який вони можуть відвідувати за бажанням в позаурочний час.

Основними напрямками роботи гуртка є:

- підготовка й участь в олімпіадах;
- розробка і представлення проектів;
- участь у навчальних виставках.

Згідно норм, наведених у програмі, затвердженій МОН України, ми використовуємо 1 комп'ютер і 1 базовий набір з розрахунку на 2-х учнів. Наразі загальна кількість робочих місць складає 12 шт.

Одним із вищезгаданих напрямків роботи курсу є залучення дітей до проектної діяльності, де учні не лише розробляють власні проекти, а й використовують навчальні проекти, розроблені спеціально для конструктора LEGO Mindstorms. У зв'язку з цим хотілось би виокремити проект Green City. Цей проект спрямований на вивчення і краще сприйняття дітьми проблем екології у світі. Під час занять учні використовують базове поле, яке нагадує карту місцевості, де розташовані моделі з енергозберігаючими ресурсами. На кожному занятті учні виконують якусь окрему місію-завдання, тим самим закріплюють нові й набуті знання.

Результатами нашої роботи є участь у всіх чотирьох Всеукраїнських олімпіадах, що почали проводитись з 2009 р., і здобуті призові місця.

Участь у I Всеукраїнській олімпіаді з робототехніки — 2009 рік.

Участь у II Всеукраїнській олімпіаді з робототехніки — 2010 рік (молодша категорія — III місце, старша категорія — II місце).

Участь у III Всеукраїнській олімпіаді з робототехніки 2011 рік (старша ліга — II місце).

Участь у IV Всеукраїнській олімпіаді з робототехніки 2012 рік (підготовча ліга — II місце, середня ліга — III місце).

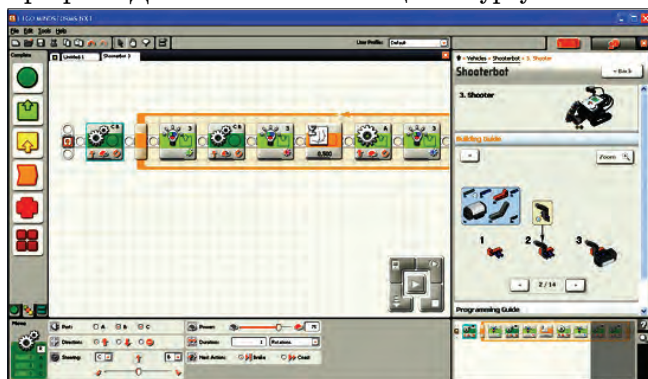


Рис. 5



МЕТОДИКА ВИКЛАДАННЯ ТЕМИ «ГРАФІЧНИЙ РЕДАКТОР» В 5-МУ КЛАСІ

Казанцева Ольга Павлівна,

вчитель інформатики Херсонського ліцею Херсонської обласної ради, заслужений учитель України.

Закінчення, початок у №1–3 за 2013 рік

УРОК 7

Тема: Розробка плану побудови зображення. **Практична робота «Створення графічних зображень за поданим планом»**

Мета: Навчитись створювати план побудови зображення. Відпрацювати вміння й навички створення графічного зображення за створеним планом.

Задачі уроку

Навчальні

Учні повинні знати:

- призначення інструментів графічного редактора;
- основні команди графічного редактора.

Учні повинні вміти:

- створювати план побудови графічного зображення;
- створювати графічне зображення за наведеним планом.

Розвивальні:

- розвиток уважності, пам'яті, логічного мислення;
- розвиток алгоритмічного мислення;
- розвиток навичок самостійної роботи на комп'ютері.

Виховні:

- виховання відповідальності, самостійності, умінь узагальнювати, міркувати;
- підвищення інформаційної культури учнів.

Тип уроку: комбінований.

Наочність, обладнання та програмні засоби: комп'ютери, проектор, екран, презентація до уроку.

Структура уроку

1. Організація учнів класу — 1 хв.
2. Перевірка домашнього завдання — 2 хв.
3. Мотивація навчальної діяльності — 1 хв.
4. Теоретична підготовка — 13 хв.
5. Фізкультхвилинка — 2 хв.
6. Практична робота — 20 хв.
7. Вправи для очей — 2 хв.
8. Домашнє завдання — 1 хв.
9. Підведення підсумків уроку — 3 хв.

Хід уроку

I. Організація учнів класу

Привітання. Перевірка готовності учнів й обладнання до уроку. Перевірка присутності учнів. Оголошення теми, мети та плану уроку.

II. Перевірка домашнього завдання

Учні повідомляють, які інструменти і команди було використано для створення зображення метелика.

III. Мотивація навчальної діяльності

Учитель. Сьогодні ми повторимо вже вивчені інструменти і команди графічного редактора Paint та навчимося самостійно створювати план для побудови майбутнього зображення.

IV. Теоретична підготовка

Учитель. Уявіть, що вам необхідно на комп'ютері створити зображення та зафарбувати пазли, представлені на малюнку, використовуючи всі відомі вам засоби графічного редактора (рис. 92).

Це завдання ви маєте виконати самостійно без детальних вказівок. Тобто ви маєте самостійно визначити, якою буде ваша послідовність дій під час виконання завдання. Отже, вам необхідно чітко записати алгоритм ваших дій для виконання практичного завдання.

Учні створюють алгоритми за командами (команди формуються по 3–5 учнів у команді залежно від кількості учнів у класі).

Після роботи над створенням алгоритмів учні представляють свою роботу. План побудови корегується вчителем та іншими учнями.

Орієнтовний алгоритм.

1. Запустіть програму Paint.
2. Побудуйте квадрат (рис. 93).
3. Розбийте квадрат на 9 квадратиків (рис. 94).

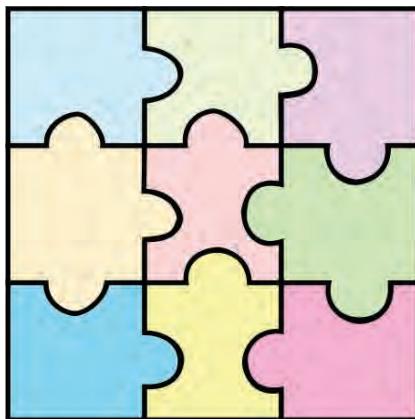


Рис. 92



Рис. 93

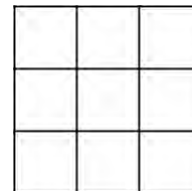


Рис. 94

4. Витріть гумкою зайві частини для з'єднання пазлів (рис. 95).

5. Створіть зображення з'єднаних частин (рис. 96), використовуючи інструмент **Кривая (Крива)**.

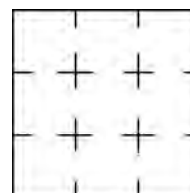


Рис. 95

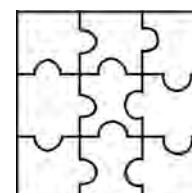


Рис. 96

6. Зафарбуйте окремі частини пазлів, використовуючи можливості палітри і додавання кольорів до неї.

V. Фізкультхвилинка

VI. Практична робота

Учні в групах створюють алгоритми для побудови зображень по варіантах і потім виконують побудову даних зображень на комп'ютері.

Варіант 1



Рис. 97

Варіант 2

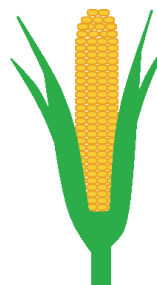


Рис. 98

Варіант 3

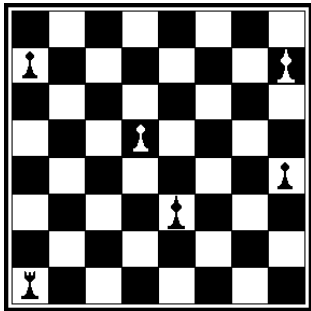


Рис. 99

VII. Вправи для очей

VIII. Домашнє завдання

Створити алгоритм побудови зображення (рис. 100) та виконати його побудову в графічному редакторі Paint.

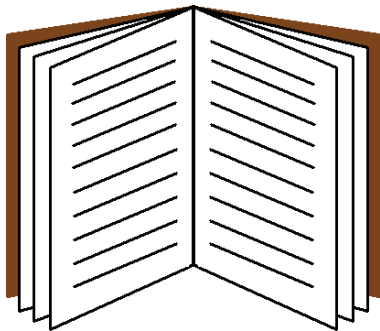


Рис. 100

IX. Підведення підсумків уроку

Учні дають відповіді на запитання вчителя. Учитель узагальнює ці відповіді.

- 1) Що сьогодні ви дізнались нового?
- 2) Чому навчилися?
- 3) Що сподобалось на уроці, що ні?
- 4) У чому виникли труднощі?

Висновки. У результаті цього уроку учні вчаться створювати алгоритм побудови зображення, закріплюють вміння та навички використання колірної палітри, дій копіювання та переміщення. Під час виконання домашнього завдання учні закріплюють вміння створювати алгоритм побудови зображення, роботи з інструментами та командами графічного редактора.

УРОК 8

Тема: Додавання тексту. Основні елементи формату символів: шрифт, накреслення, колір, розмір

Мета: Ознайомити учнів з можливостями введення та форматування тексту в графічному редакторі Paint. Навчити учнів вводити текст до зображення та змінювати його формат.

Задачі уроку

Навчальні

Учні повинні знати:

- порядок введення тексту;
- послідовність команд для зміни формату шрифту.

Учні повинні вміти:

- вводити текст до зображення;
- виводити панель атрибутів тексту;
- виконувати форматування тексту;
- переключати мовні режими роботи клавіатури.

Розвивальні:

- розвиток уважності, пам'яті, логічного мислення;
- розвиток навичок самостійної роботи на комп'ютері.

Виховні:

- виховання відповідальності, самостійності, умінь узагальнювати, міркувати;
- підвищення інформаційної культури учнів.

Тип уроку: вивчення нового матеріалу.

Наочність, обладнання та програмні засоби: комп'ютери, проектор, екран, презентація до уроку, шаблони завдань до практичних вправ.

Структура уроку

1. Організація учнів класу — 1 хв.
2. Перевірка домашнього завдання — 2 хв.
3. Мотивація навчальної діяльності — 1 хв.
4. Теоретична підготовка — 8 хв.
5. Фізкультхвилинка — 2 хв.
6. Практичні вправи — 20 хв.
7. Вправи для очей — 2 хв.
8. Робота з робочим зошитом — 5 хв.
9. Домашнє завдання — 1 хв.
10. Підведення підсумків уроку — 3 хв.

Хід уроку

I. Організація учнів класу

Привітання. Перевірка готовності учнів і обладнання до уроку. Перевірка присутності учнів. Оголошення теми, мети та плану уроку.

II. Перевірка домашнього завдання

Учні демонструють створені зображення.

III. Мотивація навчальної діяльності

Учитель. А ви знаєте, що в графічному редакторі можна не тільки малювати, але і писати. Вводити текст. Крім того ви зможете йому надавати ще й різного накреслення.

IV. Теоретична підготовка

Paint — фантастичне місто! А от чи вміють у цьому місті писати? Якщо нам знадобиться, скажімо, надіслати комусь вітальну листівку, то чи можна буде підписати її так, щоб і букви було добре видно, і шрифт був гарний?

Сьогодні ми ще завітаємо до **Писаря**, який створив геть усі написи в цьому місті.

Отже, щодо листівки, то вона, справді, має виглядати пристойно. Після того, як ви клацнете на панелі інструментів кнопку **Написати (Текст)** і створите рамку для введення тексту, на екрані буде відображено панель інструментів **Шрифты (Шрифти)** (рис. 101).

Саме тут ви зможете обрати шрифт для свого напису, а також його розмір і написання (напівжирне, курсивне і підкреслення).

Давайте спробуємо підписати на зображенні човна (рис. 102) його назву, наприклад «Чайка».

Досить часто користувачі роблять помилку і намагаються збільшити масштаб зображення аби наблизити місце, у якому слід ввести напис. Але в цьому режимі інструмент напис перестає обиратися (учитель демонструє це).

Отже, цей інструмент можна використовувати лише за справжніх розмірів малюнка (проте у версіях Paint, вбудованих до операційної системою Windows 7 і ця незручність уже врахована).

Після вибору інструменту **Текст (Текст)** переконайтесь, що у вас встановлено ту мову введення тексту, якою ви маєте вводити текстовий рядок. Це можна побачити у нижньому правому куті на **Языковой панели (Мовній панелі)**. Значення **UK** означає, що встановлено українську мову введення, **RU** — російську, **EN** — англійську. Для зміни мови введення тексту слід клацнути на значок мовної панелі й обрати необхідну. Також це можна зробити, натиснувши комбінацію клавіш **Ctrl+Shift** або **Alt+Shift** залежно від налаштувань на вашому комп'ютері.

Після цього ви можете уже ввести свій текст (рис. 103). Напевно, вам не дуже сподобається наявність білого прямокутника навко-

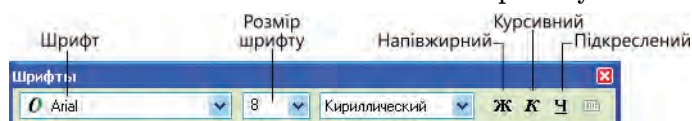


Рис. 101

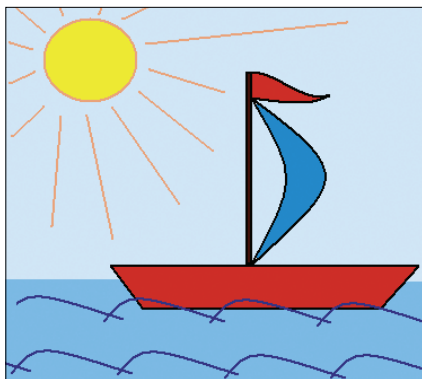


Рис. 102



Рис. 103

ло тексту. Його не було б, якби ви, обравши інструмент **Надпись (Текст)**, клацнули в області під панеллю інструментів нижній значок. Тоді напис було б створено на прозорому тлі. Це той самий значок, який ми використовували під час копіювання об'єктів, для того щоб разом із ними не було скопійовано колір тла.

Ось так козаки підписали все на своєму малюнку (рис. 104).



Рис. 104

Якщо ви добре розібралися з написами, пропоную виконати фізкультхвилинку та перейти до практичних завдань.

V. Фізкультхвилинка

VI. Практичні вправи

Вправа 1. Запустіть програму Paint і відкрийте в її вікні файл **Назви_тварин.bmp**. Скориставшись інструментом **Надпись (Текст)**, напишіть під малюнками назви птахів і тварин.

Вправа 2. Відкрийте у вікні програми файл **Зошит.bmp**, і підпи-

шіть обкладинку зошита, як показано на рисунку 105.

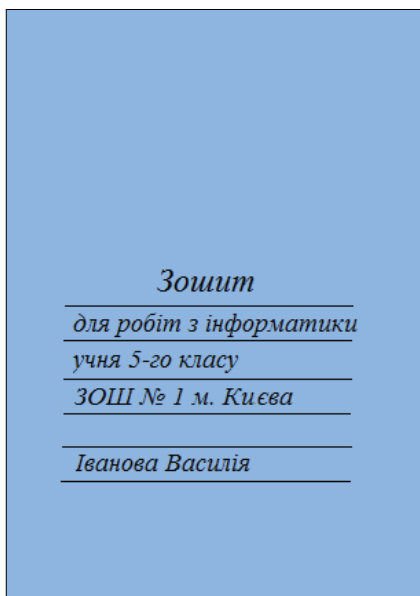


Рис. 105

Вправа 3. Виконайте оформлення вітальної листівки, використовуючи шаблони зображень та текст привітання з окремих файлів **шаблон_1.bmp, шаблон_2.bmp, шаблон_3.bmp, шаблон_4.bmp, шаблон_5.bmp, шаблон_6.bmp, шаблон_7.bmp, вітання.txt**.

Аналіз практичної вправи.

Учитель. Сьогодні вам буде запропоновано шаблони листівок та зразки текстових вітань. Вам необхідно обрати будь-який шаблон, а також текстове привітання. Скопіювати текст обраного привітання до шаблону зображення. Скопійованому тексту надати привабливе оформлення. Шаблони ви можете обирати різні, але алгоритм виконання завдання у всіх буде практично однаковим. Отже спочатку створіть алгоритм виконання завдання, а потім приступайте до його практичної реалізації.

VII. Вправи для очей

VIII. Робота з робочим зошитом

1. Запишіть на виносках рисунка 106, які атрибути ви можете призначити для щойно введеного тексту.

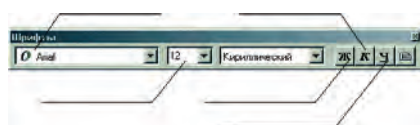


Рис. 106

IX. Домашнє завдання

1. Оформіть підпис шаблону листівки за власним бажанням (рис. 107).



Рис. 107

2. Дайте відповіді на запитання.

- 1) Для збільшення або зменшення розмірів зображення використовують інструмент: _____
- 2) Для побудови квадрата або кола слід утримувати клавішу _____
- 3) Для збереження файлу необхідно: _____
- 4) Запишіть один із способів копіювання фрагменту зображення: _____

5) Яким повинно бути значення масштабу для введення тексту? _____

X. Підведення підсумків уроку

Учні дають відповіді на запитання вчителя. Учитель узагальнює ці відповіді.

- 1) Що сьогодні ви дізналися нового?
- 2) Чому навчилися?
- 3) Що сподобалось на уроці, що ні?
- 4) У чому виникли труднощі?

Висновки. У результаті даного уроку учні знайомляться з можливостями введення тексту, із способами зміни параметрів шрифту і переключення мовного режиму клавіатури. Виконуючи практичне завдання, учні вчать вводити текст і змінювати параметри шрифту. Домашнє завдання дозволить краще зрозуміти можливості додавання тексту в графічному редакторі.

УРОК 9

Тема: Підсумковий урок

Мета: Перевірити, узагальнити та систематизувати знання учнів з теми «Графічний редактор».

Задчі уроку

Навчальні:

Учні повинні знати:

- призначення й основні функції графічного редактора;
- правила роботи з графічним редактором;
- можливості інструментів графічного редактора;
- основні операції зі створення і редагування зображень в графічному редакторі;
- можливості використання палітри редактора Paint;
- можливості введення та редагування тексту в графічному редакторі;

• команди для відображення й обертання зображення.

Учні повинні вміти:

- запускати графічний редактор;
- створювати зображення в графічному редакторі;
- користуватися всіма інструментами малювання в редакторі Paint;
- задавати і змінювати колір фігури і колір фону;
- виділяти фрагменти малюнка;
- копіювати і переміщувати виділений фрагмент;
- обертати, відображати, розтягувати/стискати малюнок;
- змінювати кольори палітри;
- уводити текст;
- змінювати масштаб зображення.

Розвивальні:

- розвиток уважності, пам'яті, логічного мислення;
- розвиток алгоритмічного мислення;
- розвиток навичок самостійної роботи на комп'ютері.

Виховні:

- виховання відповідальності, самостійності, умінь узагальнювати, міркувати;
- підвищення інформаційної культури учнів.

Тип уроку: перевірки знань.

Наочність, обладнання та програмні засоби: комп'ютери, шаблон завдання до письмової роботи.

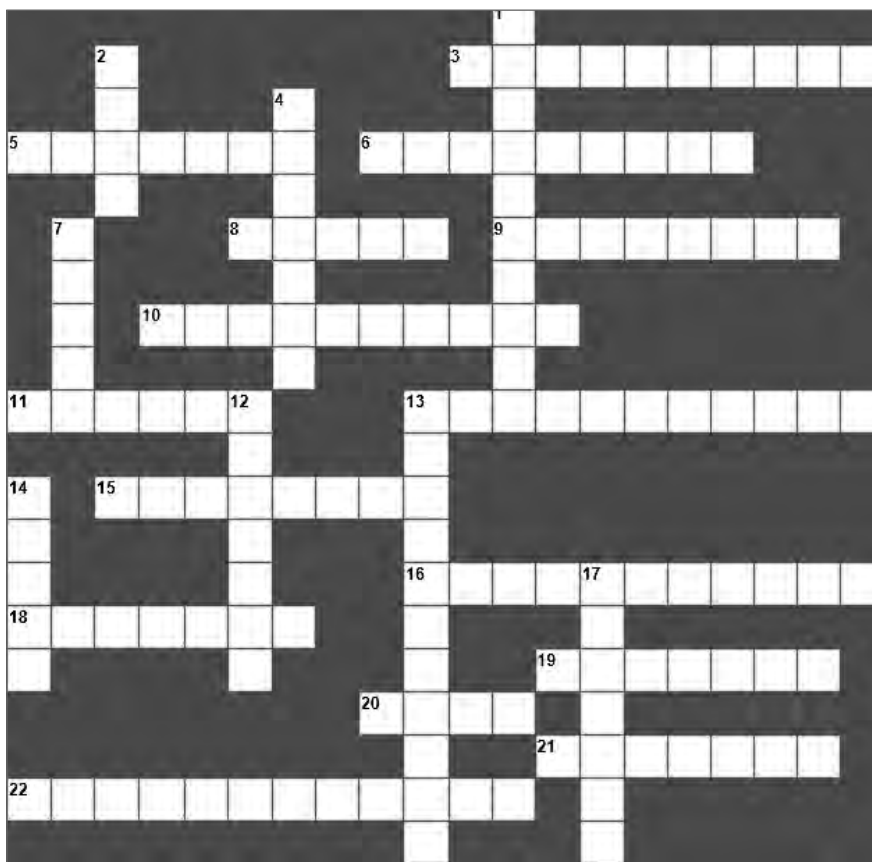


Рис. 108

Таблиця 3

Запитання і відповіді до них

По горизонталі	По вертикалі
1. Балончик з фарбою (розпилювач).	1. Ця команда потрібна для отримання однакових зображень (копіювання).
5. Дія зафарбування (заливка).	2. Фігура, що створюється при використанні інструменту Еліпс та утриманні клавіші Shift (коло).
6. Після використання цього інструменту можна виконати копіювання або переміщення (виділення).	4. Має багато кольорів (палітра).
8. Вона з'являється при збільшенні масштабу (сітка).	7. Лінія, що виходить без використання лінійки (крива).
9. Фігура, що утворюється після використання інструменту Лінія (відрізок).	12. Ця фігура виходить при використанні інструменту Прямокутник та утриманні клавіші Shift (квадрат).
10. Цю команду використовують, щоб робота не пропала марно (збереження).	13. Ця команда потрібна для перенесення зображення з одного місця на інше (переміщення).
11. Витирає зайве (ластик).	14. Для її побудови на папері використовують лінійку (пряма).
13. За допомогою цього інструменту не виникає проблем при зображенні шкільної дошки, шафи тощо (прямокутник).	17. Малює тонкими лініями (олівець)
15. Paint — це графічний ... (редактор).	
16. Будує фігуру з великою кількістю кутів (многокутник).	
18. Інструмент для збільшення (масштаб).	
19. Будь-яка точка зображення (піксель).	
20. Рядок графічного редактора, що має різні команди (меню).	
21. Малює лініями різних видів та розмірів (пензлик).	
22. Цю команду використовують для отримання симетричного або дзеркального зображення (відображення)	

Структура уроку

1. Організація учнів класу — 1 хв.
2. Кросворд — 20 хв.
3. Вправи для очей — 2 хв.
4. Робота з бланком завдання — 17 хв.
5. Домашнє завдання — 2 хв.
6. Підведення підсумків уроку — 3 хв.

Хід уроку

I. Організація учнів класу

Привітання. Перевірка готовності учнів та обладнання до уроку. Перевірка присутності учнів. Оголошення теми, мети та плану уроку.

II. Перевірка домашнього завдання

Учні демонструють оформлені листівки і надають відповіді на запитання.

III. Кросворд

Учитель. Сьогодні ми перевіримо ваші знання з усіх питань, з якими ви встигли познайомитись, подорожуючи країною Інформатика, ознайомлюючись з мальовничим містом Paint. Перевірку ваших знань здійснимо використовуючи електронний кросворд (рис. 108, табл. 3).

Для відповіді на запитання клацайте на номер запитання в кросворді та введіть правильну відповідь в поле для відповіді, після чого натискайте кнопку **Відповідь**. Для одержання підказки натискайте кнопку **Підказка**. Пам'ятайте, що, користуючись підказкою, ви отримуєте

менше балів. Не натискайте кнопку **Перевірити** до закінчення тестування, оскільки ви не будете мати змоги повернутися до заповнення кросворду. Програма здійснює оцінювання у відсотках (табл. 4).

**IV. Вправи для очей
Фізкультхвилинка**

Таблиця 4

12	93%-100%	6	43%-50%
11	84%-92%	5	34%-42%
10	76%-83%	4	26%-33%
9	68%-75%	3	18%-25%
8	59%-67%	2	9%-17%
7	51%-58%	1	0%-8%

V. Робота з бланками завдань.

Учні виконують завдання по роботі з графічним редактором Paint, працюючи в бланках завдань **Завдання.doc**.

Завдання до теми «Графічний редактор Paint»

1. Визначіть елементи вікна графічного редактора Paint (рис. 109).

2. Запишіть поряд з інструментом номер, що відповідає його призначенню (рис. 110).

3. Запишіть алгоритм створення зображення (рис. 111).

VI. Підведення підсумків уроку

У результаті вивченої вами першої теми з інформатики ви отримали певні знання, проаналізуйте вашу роботу протягом процесу вивчення.



Рис. 111

Закінчіть фрази:

1. Подорожуючи з козаками країною Інформатика, я

- узнав _____
- зрозумів _____
- навчився _____

2. Більше всього мені подобається _____

3. Краще всього на уроках в мене виходило _____

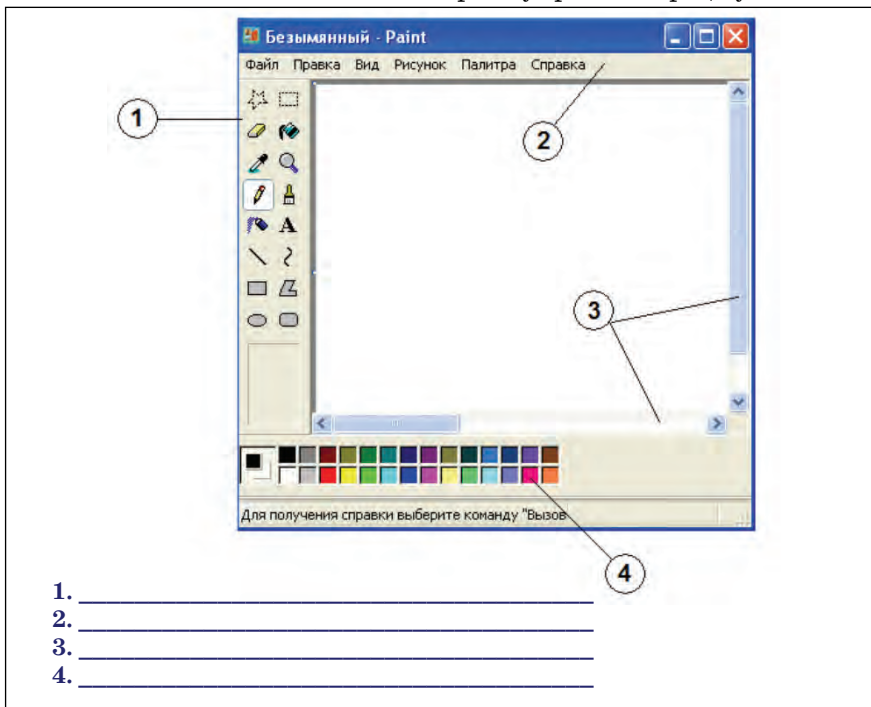
4. Основні труднощі в мене були _____

5. Я хотів би ще дізнатися _____

Учитель. Отже, можна зробити висновок, що ще одна подорож країною **Інформатика** допомогла вам зрозуміти, що таке графічний редактор Paint. Ви познайомились з новими термінами, розглянули призначення інструментів графічного редактора, дізнались про існування різних команд для виконання дій з фрагментами зображення. Навчилися використовувати інструменти графічного редактора та виконувати різноманітні дії з фрагментами зображень.

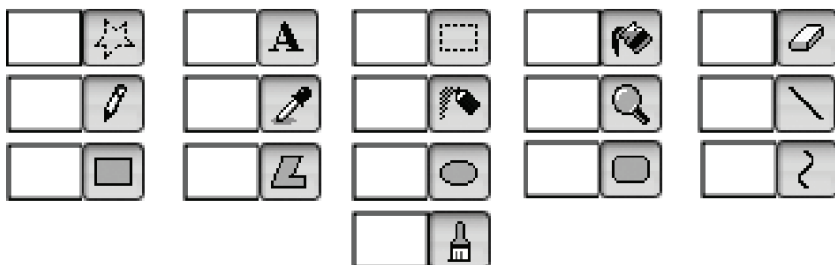
VI. Домашнє завдання

Проте на цьому наші мандрівки не закінчуються. Оскільки козаки вирішили, що малювання на комп'ютері для них не достатньо і вони ще хочуть навчитись представляти свою роботу в привабливому вигляді на комп'ютері. Тепер ми разом з ними вирушимо в нову подорож, що допоможе ознайомитись з програмою, яка дозволить створювати власні комп'ютерні проекти-презентації. Можливості програми, що ви будете вивчати, є багатогранними. За допомогою цієї програми можна своїм друзям розповісти про світ своїх захоплень. Підготуйте дома доповідь про світ ваших захоплень і запропонуйте, що з вашої доповіді можна було б представити на комп'ютері, аби ваша розповідь була підтверджена наочно.



1. _____
2. _____
3. _____
4. _____

Рис. 109



- | | |
|--|---|
| 1. Побудова прямокутника. | 10. Збільшення масштабу. |
| 2. Побудова кривої лінії. | 11. Витирання зайвих фрагментів зображення. |
| 3. Зображення ліній різного виду різної товщини. | 12. Вибір кольору. |
| 4. Виділення прямокутної області зображення. | 13. Виділення довільної області зображення. |
| 5. Побудова відрізка. | 14. Вставка тексту. |
| 6. Розпилювання фарби. | 15. Побудова прямокутника з округленими кутами. |
| 7. Заливка замкнутої області. | 16. Побудова многокутника. |
| 8. Побудова довільних тонких ліній. | |
| 9. Побудова овалів. | |

Рис. 110

ХМАРНІ ТЕХНОЛОГІЇ. СОЦІАЛЬНЕ СЕРЕДОВИЩЕ ПРОГРАМУВАННЯ TOUCHDEVELOP

Литвинова Світлана Григорівна,
кандидат педагогічних наук, старший науковий співробітник Інституту інформаційних технологій і засобів навчання НАПН України,

Тебенко Олексій Віталійович,
провідний інженер Інституту інформаційних технологій і засобів навчання НАПН України.



Анотація. Створення єдиного інформаційного простору, який би забезпечив доступ усіх шкіл, педагогічних працівників, учнів і навіть батьків до якісної середньої освіти, довгий час залишався неможливим. Поява хмарних технологій забезпечить реалізацію даної ідеї і буде сприяти розвитку можливостей розробки і використання електронних освітніх ресурсів у навчально-виховному процесі звичайних шкіл. У статті розкрито основні ідеї хмарної технології «інфраструктура як сервіс» на прикладі Touchdevelop. Описано його можливості у розробці ігрових електронних освітніх ресурсів, розкрито особливості, уточнено поняття соціального середовища програмування, узагальнені можливості реалізації в середній загальній освіті, наведено приклади розробки електронних освітніх ресурсів.

Ключові слова: Touchdevelop, середовище програмування, хмарні технології, навчальні ігри, електронні освітні ресурси.

Постановка проблеми. Погляди учнів наразі все частіше затримуються на мобільних телефонах, планшетах та інших гаджетах, основне призначення яких нині залишається розважально-ігрове. Тому педагоги загальної середньої освіти піднімають питання забезпечення навчально-виховного процесу якісним електронним освітнім контентом, не тільки для стаціонарних комп'ютерів, а й для інших сучасних гаджетів, які можна задіяти для навчання як у школі, так і вдома.

Одним з актуальних питань у наукових колах залишається використання мережі інтернет на всіх етапах навчально-виховного процесу і в управлінській діяльності загальноосвітніх навчальних закладів. Такі технології як веб, віртуальні, хмарні радикально змінюють навчальне середовище і роблять освіту більш доступнішою.

Поєднання можливостей сучасних гаджетів і мережі інтернет створює доступне освітнє середовище без обмеження доступу до необхідних даних. Насичення такого середовища освітніми електронними ресурсами підвищить інтерес учнів до навчальних предметів, активізує їх пізнавальну діяльність і створить умови для всебічного розвитку особистості дитини, розкриє її вроджені й набуті здібності.

Реалізацію таких можливостей освітяни покладають на сучасні хмарні технології.

Аналіз останніх досліджень показав, що питання загальної теорії використання хмарних технологій в освіті досліджується вченими Биковим В.Ю., Жалдаком М.І., Спіріним О.М., Семеріковим С.О., Сейдаметовою З.С., Аблязімовою Є.І., Меджитовою Л.М., Сейтвелієвою С.Н., Темненко В.А., використання хмарних обчислень для організації тестування розкрито у роботах Морзе Н.В., Кузьмінської О.Г., організація самостійної роботи за допомогою хмарних сервісів Яндєкс відображено у роботах Алексанян Г.А., організація «віртуальної» учительської засобами Google-site дослі-

джується Рождественською Л.В., а засобами Microsoft Office 365 Литвиновою С.Г., перспективи розвитку програмного забезпечення як послуги для створення документів електронної бібліотеки досліджується Шиненко М.А., Сороко Н.В., створення навчальних ресурсів у середовищі moodle на основі технології «cloud computing» Сергієнком В.П., Войтович І.С.

Питання використання хмарних середовищ програмування електронних освітніх ресурсів для використання у навчально-виховному процесі загальноосвітніх навчальних закладів вченими не досліджувалося.

Мета статті: описати можливості Touchdevelop у розробці ігрових електронних освітніх ресурсів, розкрити його особливості, уточнити поняття соціального середовища програмування, узагальнити можливості реалізації в середній загальній освіті, навести приклади розробки електронних освітніх ресурсів.

Виклад основного матеріалу. Глобальна мережа Інтернет, без перебільшення, слугувала початком інформаційної революції у людській цивілізації. Показники числа Інтернет-користувачів і користувачів широкосмугового Інтернету в Україні перевищують середньостатистичні показники для країн, що розвиваються, поступаючись показникам розвинених країн. Можна зробити висновок, що наявні на даний момент в Україні загальнонаціональні ресурси ІТ-інфраструктури цілком достатні для застосування нових веб-технологій в освіті [4, с. 8–9].

Усе більшої популярності в освіті набувають хмарні технології (cloud technologies) — це новий сервіс, основне завдання якого полягає у віддаленому використанні засобів обробки і зберігання даних [3, с. 99–100]. Тип хмари залежить від призначення, але за темпами розвитку і поширення першості набувають публічні хмари (public cloud) — інфраструктура, призначена для вільного використання широкою публікою [1, с. 23–24].

Педагогічна спільнота тільки почала активно досліджувати можливості хмарних технологій і на їхню думку заслуговує особливої уваги нове соціальне хмарне середовище програмування різноманітних застосунків як ігрових, так і навчальних — TouchDevelop.

Під *соціальним хмарним середовищем програмування* ми розуміємо хмарну платформу з необмеженим доступом користувачів для розробки і використання нових електронних ресурсів.

Хмарне середовище програмування працює на мобільних телефонах, планшетах та на звичайних комп'ютерах і ноутбуках. Високорівневий огляд архітектури екосистеми TouchDevelop (рис. 1) включає спільноту розробників, які використовують ОС Windows, телефони, веб-сайти і хмарні інфраструктури, про що зазначає Найджел Хорспула [5].



Рис. 1. Екосистема TouchDevelop

Нині TouchDevelop нараховує 70 тис. зареєстрованих користувачів, 18 тис. опублікованих скриптів та всього 260 тис. завантажень готових програм. Відзначимо, що у WindowsStore і WindowsPhoneStore завантажено понад 200 готових програм, які були розроблені на touchdevelop [5]. Соціальне середовище програмування дозволяє здійснювати обмін досвідом з друзями в хмарі і працює на WindowsPhone, PC, Mac, IPAD, iPhone, Android та ін.

Розуміння реальних потреб у сфері освіти спрямовує можливості хмарного середовища програмування на розвиток потенціалу учнів і педагогів у розробці інноваційних застосунків для навчання на платформі Windows.

Отже, детально познайомимося з новим хмарним середовищем програмування: www.touchdevelop.com (рис. 2).

Особливості соціального середовища програмування:

- типізована, спрощена мова;
- напівструктурований редактор;
- типізовані, імперативні, процедурні структури даних;
- автозаповнення на основі краудсорсинга.

Звертає на себе увагу і той факт, що для використання хмарного середовища програмування не потрібно мати технічну освіту. Процес має такий вигляд:

- будь-який користувач може публікувати, редагувати свої сценарії і запускати довільні сценарії, які розміщені у хмарному середовищі програмування. Для цього треба пройти обов'язкову процедуру реєстрації на TouchDevelop з вашим ідентифікатором Windows Live ID або Facebook акаунтом (рис. 3);



Рис. 2. Сторінка входу до TouchDevelop

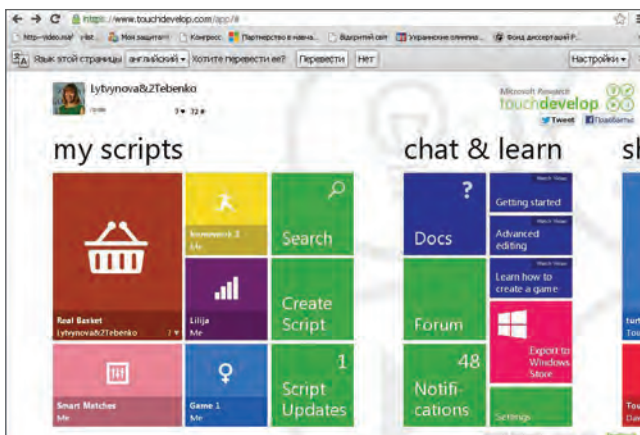


Рис. 3. Домашня сторінка користувача

- ви пишете сценарій навчальної гри, натискаючи тільки кнопки на екрані планшета або мобільного телефону. Вам не обов'язково мати потужний комп'ютер або ноутбук з клавіатурою і глибокі знання мови програмування (рис. 4);

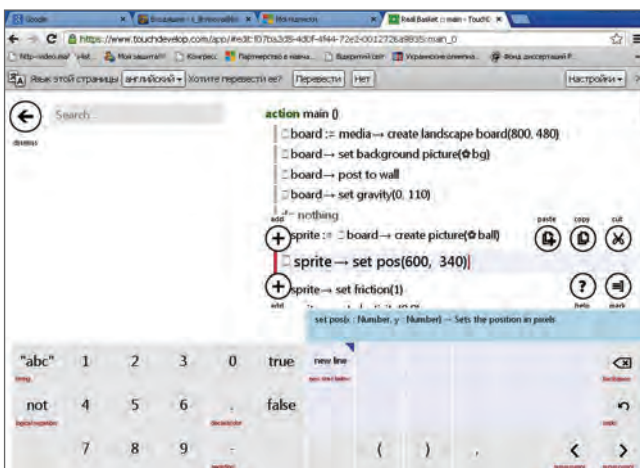


Рис. 4. Вигляд робочого поля для створення навчальної гри

- сценарій розглядається як виконання різних завдань (команд) об'єктом і його виконання подібне до звичайного програвача (рис. 5).
- ви можете поділитися своїми скриптами з іншими колегами, опублікувавши їх у бібліотеці TouchDevelop або шляхом публікування їх як застосунків до WindowsStore або WindowsPhoneStore. Бібліо-

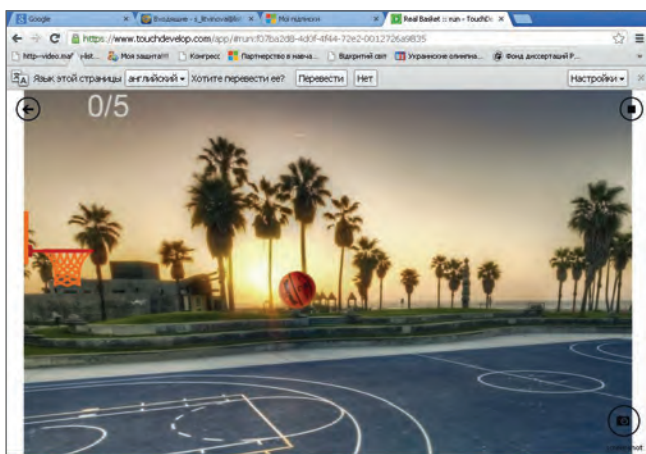


Рис. 5. Момент програвання гри

тека сценаріїв – це місце в хмарі, де публікуються і зберігаються всі застосунки TouchDeveloper. Ви можете переглянути і знайти скрипти у хмарному середовищі програмування або у бібліотеці (рис. 6), яку інтегровано у застосунок TouchDeveloper.

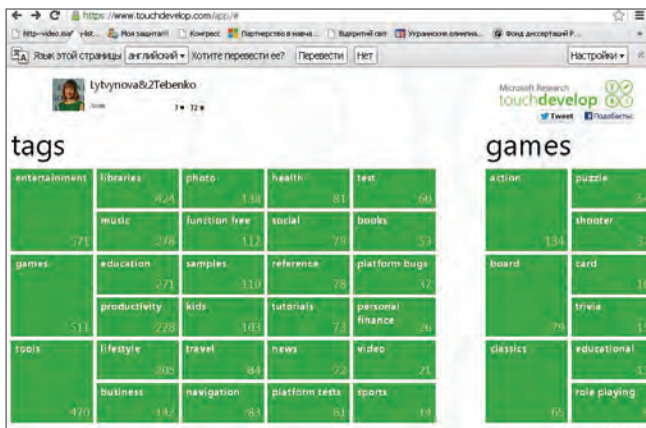


Рис. 6. Бібліотека користувача

Перевага хмарного середовища програмування у тому, що воно дозволяє легко знайти контакт з іншими колегами або професіоналами. Ви можете писати не тільки відгуки та зауваження до навчальної гри. У вас є можливість додати нові скріншоти безпосередньо із запущеного скрипта; відслідкувати загальний рейтинг лідерів, де видно, як інші колеги або учні оцінили застосунок; отримати дієву консультацію або допомогу у розробці інноваційних електронних освітніх ресурсів.

Розробка сценарію навчальної гри для бібліотеки включає такі моменти: вибір назви сценарію, індивідуальний колір та іконку (від billybarker.net) (рис. 7).

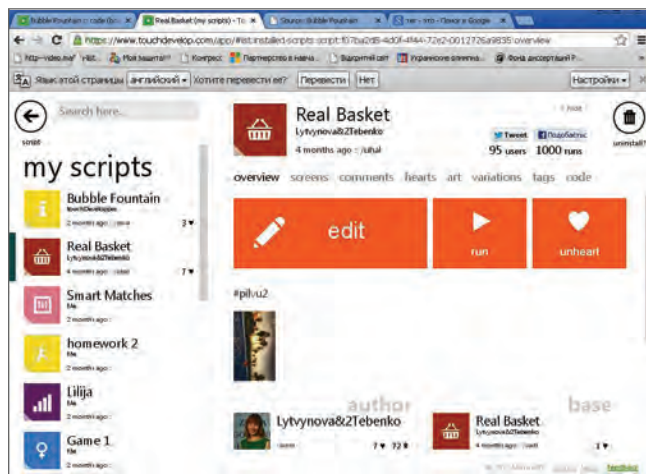


Рис. 7. Відображення іконки гри, кнопок запуску і редагування

У хмарному середовищі програмування дозволено обмін сценаріями, забезпечено можливість усім розробленим сценаріям отримувати короткі URL-адреси, наприклад, touchdeveloper.com/gdmmr (гра BubblePop). Це дає можливість поділитися URL-адресою з іншими колегами.

Можливості реалізації в середній загальній освіті:

- *здійяти власні пристрої* учнів і вчителів. TouchDeveloper дозволяє працювати на більшості пристроїв, які учні приносять до школи (мобільні телефони, планшети, нетбуки, ноутбуки) і створювати новітнє навчальне середовище;
- *розширити клас*. Учні можуть продовжувати працювати з електронними освітніми ресурсами у свій вільний час;
- *активізація й індивідуалізація навчання і розвитку*. Залучення учнів до навчання через гру, створення ігрових моментів на уроках, виконання творчих індивідуальних завдань;
- *банк ресурсів*. TouchDeveloper накопичує не тільки навчальні програми, слайди, книги з використання соціального середовища програмування, а й електронні ресурси для навчання.

Розглянемо базові фрагменти, які можна використати на touchdeveloper (табл. 1 і табл. 2).

Таблиця 1

Фрагменти синтаксису

→	Використовується для доступу до властивостей елемента
:=	Присвоїти змінній значення
//	Позначити коментар

Таблиця 2

Фрагменти команд

varsongs := media → songs	Копіювати збірник пісень від медіабібліотеки і зберегти його у змінній 'пісні'
phone → vibrate(1)	Дати команду телефону вібрувати протягом 1 секунди
"Мене звати ..." → post to wall	Відобразити текст
media → songs → random → play	Грати пісні в довільному порядку
senses → camera → post to wall	Відобразити діючу камеру
senses → recordmicrophone → posttowall	Записати звук і відтворити його
for 0 ≤ i < 10 do	Цикл, повторити 10 разів
senses → camera → preview → post to wall	Показати миттєву картинку з камери на стіні
var c := colors → fromrgb(p → x, p → y, p → z)	Отримати колір з червоного, зеленого, синього каналів (RGB)
wall → set background(c)	Відобразити колір як фон

Висновки

Нове соціальне середовище програмування Touchdeveloper відкриває освітянам нові шляхи реалізації власного потенціалу. Вони мають можливість самостійно розроблювати інтерактивні електронні освітні ресурси для підвищення якості викладання свого предмету, унаочнення навчального матеріалу, активізації пізнавальної діяльності учнів. Разом з тим — це є потужний інструмент у руках учителя інформатики, яким можна повернути інтерес учнів до програмування.

Можливості використання соціального середовища програмування Touchdeveloper вимагають додаткових досліджень, особливо інтеграції у навчально-виховний процес загальноосвітніх навчальних закладів.

Розглянемо приклад №1 «Черепашка Ліло»

Уявіть, що вам подарували черепаху Ліло. Ліло маленька, чудова черепаха, що рухається, повертається і малює кольоровим олівцем за вашою командою. Ліло знає тільки основні команди, але ми навчимо її ще кількох трюків.

Щоб почати гратися з Ліло, створіть скрипт командою **Create Script** з головного меню, потім виберіть **turtle drawings**. Цей шаблон завантажить Ліло і ми готові до гри.

Початковий код

Функція main містить 3 рядки коду. Перший рядок потрібен, щоб Ліло прокинулася і з'явилася на екрані `turtle→initialize landscape`.

Тоді, ми просимо Ліло змінити колір олівця, яким вона буде малювати на випадковий `turtle→pen color(colors→random)`.

Малюємо квадрат

Давайте спробуємо намалювати квадрат з Ліло. Для цього ми повинні дати Ліло такі команди, щоб це була послідовність ходів вперед і поворотів.

Крок вперед

Тепер, коли Ліло готова рухатися, ми повинні дати їй команду рухатися вперед на 200 одиниць. На відміну від своїх реальних колег, віртуальні черепахи досить швидкі, і переміщення відбуваються відразу. Оскільки Ліло завжди з'являється в центрі екрану і дивиться праворуч, то вона за поданою командою малює горизонтальну лінію вправо (рис. 8)

`turtle→forward(200)`.

Поворот

Ліло не знає як рухатися в бік, але вона може повертати! Попросимо Ліло повернути на 90 градусів ліворуч. Ми малюємо квадрат так, що нам потрібно, щоб намальовані лінії були під прямим кутом (рис. 9)

`turtle→left turn(90)`.



Рис. 8. Ліло рухається вперед



Рис. 9. Ліло повернулася на 90 градусів

Крок вперед знову

Ліло дивиться вгору і готова рухатися далі. Ми говоримо їй, щоб вона перемістилася вперед ще на 200 одиниць (рис. 10)

`turtle→forward(200)`.

Поворот і крок вперед

Половину роботи вже зроблено, але ми маємо продовжувати говорити Ліло, що робити далі. Давайте скажемо їй, щоб вона повернула на 90 градусів ліворуч (рис. 11) і рухалася вперед (рис. 12).



Рис. 10. Ліло рухається вперед



Рис. 11. Ліло повернулася на 90 градусів

`turtle→left turn(90)`
`turtle→forward(200)`
 і, ще раз...
`turtle→left turn(90)`
`turtle→forward(200)`

і ще один поворот так, щоб Ліло зупинилася саме там, де вона почала (рис. 13)

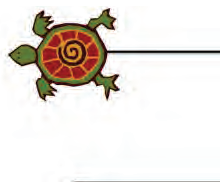


Рис. 12. Ліло рухається вперед



Рис. 13. Ліло зупинилася

`turtle→left turn(90)`.

Ліло намалювала квадрат!

Додаткові можливості

Змініть колір олівця, перш ніж Ліло буде рухатися вперед. Пам'ятайте, що ви можете використовувати функцію **pen color**.

Змініть товщину олівця: `turtle→set thickness(6)`.

Цикли

Ліло намалювала квадрат..., але для цього ми набрали багато тексту! А скільки коду потрібно ввести, якщо буде потрібно намалювати восьмикутника. Можливо, є щось краще? Ми попросили Ліло пройти уперед (forward) і повернути ліворуч (turn left) 4 рази. А що, якщо замість цього ми можемо використати один і той же код 4 рази... Це саме те, що ми збираємось зробити з циклом **for**.

`for 0 ≤ i < 4 do`

...

Весь код всередині циклу for буде виконано один раз за кожну ітерацію. Ідея циклу for — локальна змінна index починається з 0 і збільшується на кожній ітерації, поки не стане більшою або рівною 4. Отже, index буде мати значення 0, 1, 2, 3 і потім цикл закінчиться.

Попросимо Ліло намалювати ще один квадрат, але трохи менший:

`for 0 ≤ j < 4 do`
`turtle→forward(100)`
`turtle→left turn(90)`

Спіралі

Час спробувати новий трюк з Ліло. Цього разу ми дізнаємося, як малювати спіралі.

Малювання спіралі дуже схоже на малювання квадрату, різниця тільки в тому, що ми говоримо Ліло рухатися на кожній ітерації вперед трохи далі. Так як же нам змінити відстань для `turtle→forward` на кожній ітерації? Один з можливих способів є використання індексу циклу `for`. Замість того, щоб передавати 100 до `turtle→forward(100)` ми перемножимо `index` на 5 і додамо отримане число до 100

```
for 0 ≤ j < 50 do
  turtle → forward(100 + j * 5)
...
```

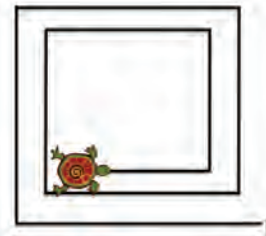


Рис. 14. Ліло рухається по спіралі

Завдання для самостійної роботи

Намалювати трикутник. Ми повинні сказати Ліло 3 рази повернути ліворуч на 120 градусів і рухатися вперед.

Попросіть Ліло намалювати п'ятикутник (5 сторін) або шестикутник (6 сторін).

Розглянемо приклад №2 «Забий м'яча»

Створимо гру (див. рис. 5), у якій користувачу потрібно буде збити м'яча, який вилітає з-за меж ігрового поля випадково.

Спершу створимо дві глобальні змінні — `board` та `ball`, отже вони будуть доступні з будь-якого місця нашої програми:

`board` відповідає за відображення ігрового поля, а `ball` — спрайт, який буде на ньому рухатися. Глобальні змінні створюються в розділі **data** (рис. 15).

Тепер визначимо й ініціалізуємо змінні, потрібні для роботи.

Виконання будь-якого скрипта починається з функції `main: action main ()`.

- Створюємо ігрове поле, на якому будемо розміщувати наші спрайти
`board:=media→create landscape board(800, 480).`
- Опублікувати ігрове поле на стіні (поки поле не опубліковано, воно не буде відображатися користувачу)
`board→post to wall.`
- Встановити колір для ігрового поля
`board→set background(colors→light gray).`
- Створити спрайт, у нашому випадку — коло діаметром 50 пікселів. Квадратик біля назви змінної вказує на те, що ця змінна глобальна (доступна з будь-яких функцій)

`ball :=board→create ellipse(50, 50).`

- Встановити колір для м'яча
`ball→set color(colors→red).`
- Встановити рівень гравітації по ігреку (спрайти повинні набирати швидкість 400 піксель на секунду)
`board→set gravity(0, 400).`

Тепер створимо дві функції в розділі `events`(події) — ці функції системні, їх не можна викликати із самої програми, але вони виконуються автоматично залежно від дій користувача:

- `gameloop` — слугує глобальним циклом (грубо кажучи, ця функція буде викликатися кожну соту секунду під час виконання програми);
- `tap sprite: ball` — з назви зрозуміло, що ця функція буде визиватися, коли користувач натисне на м'яч, `ball→set x(board→width+math→random(board→width - ball→width));`
- зробимо так, щоб м'яч вилітав по параболі, для цього потрібно вказати швидкість відносно координати X
`var x speed := 30 + math→random(120)`
`if ball→x < board→width / 2 then` (якщо м'яч знаходиться на лівій стороні екрану, запускаємо м'яч праворуч)
`ball→set speed x(x speed)`
`else` (інакше запускаємо м'яч ліворуч)
`ball→set speed x(- x speed)`
`else do nothing`
`event tap sprite: ball (`
`sprite : Sprite,`
`x : Number,`
`y : Number)`
`do`
- користувач натиснув на м'яч, перемістивши його за межі ігрового поля, тим самим емулюємо, що м'яч було збито
`sprite→set y(board→height+sprite→height +100).`



Рис. 16. Вигляд розділу `events`



Рис. 15. Вигляд розділу `data`

Література

1. Литвинова С.Г. Віртуальна учительська за хмарними технологіями / С.Г. Литвинова // Комп'ютер у школі та сім'ї. — 2013. — № 2 (106) — С. 23–25.
2. Литвинова С.Г. Віртуальний клас як комп'ютерно орієнтоване навчальне середовище вчителя загальноосвітнього навчального закладу [Електронний ресурс] / С.Г. Литвинова // Інформаційні технології і засоби навчання. — 2011. — №2 (22). — Режим доступу до журналу: <http://journal.iitta.gov.ua/index.php/itt/article/view/331/387#.UX1SV6lepyM>.
3. Литвинова С.Г. Хмарні технології в управлінні дошкільними навчальними закладами / С.Г. Литвинова // Інформаційно-комп'ютерні технології в економіці, освіті та соціальній сфері. Випуск 8. — Симферополь: ФЛП Бондаренко О.А., 2013. — С. 99–101.
4. Облачные технологии и образование / [Сейдаметова З.С., Абляимова Э.И., Меджитова Л.М., Сейтвелиева С.Н., Темненко В.А.]; под общ. ред. З.С. Сейдаметовой. — Симферополь: «ДИАИПИ», 2012. — 204 с.
5. Nigel Horspool TouchDevelop programming on the Go [Електронний Ресурс] / [Nigel Horspool, Judith Bishop, Arjmand Samuel and others] // Режим доступу до книги: <https://www.touchdevelop.com/book#single>.

ЗАДАЧІ ХХVI ВСЕУКРАЇНСЬКОЇ ОЛІМПІАДИ З ІНФОРМАТИКИ ТА РЕКОМЕНДАЦІЇ ЩОДО ЇХ РОЗВ'ЯЗУВАННЯ

Бондаренко Віталій Вікторович,

*асистент факультету кібернетики Київського Національного
університету ім. Тараса Шевченка,*

Ягієв Шаміль Ігорович

менеджер проектів компанії «Арісент Україна».

ЗАВДАННЯ ДРУГОГО ТУРУ

1. «Конструктор» (Данило Мисак)

На свій перший день народження Меггі Сімпсон, персонаж мультсеріалу «Сімпсони», отримала в подарунок конструктор: ігровий набір, що складається з паличок різної довжини. Кінці паличок можна скріплювати, причому з'єднані так палички можуть утворювали довільний ненульовий кут, крім розгорнутого (180°). Меггі хоче скласти опуклий многокутник, використавши якомога більшу кількість паличок із конструктора як сторони цього багатокутника.

Завдання. Напишіть програму `set`, що за розмірами паличок у конструкторі визначить, чи вдасться Меггі скласти з паличок опуклий многокутник, і якщо вдасться, то визначить, яку найбільшу кількість паличок вона зможе для цього використати.

Вхідні дані. У першому рядку вхідного файлу `set.dat` указано кількість N паличок у наборі, $2 \leq N \leq 10^5$. У другому рядку записано N натуральних чисел, менших за 10^9 (не обов'язково попарно різних) — довжини паличок.

Вихідні дані. Вихідний файл `set.sol` повинен містити єдине число — найбільшу кількість паличок з набору, з яких можна скласти опуклий многокутник, або число 0, якщо скласти опуклий многокутник не вдасться.

Оцінювання. Набір тестів складається з 3 блоків, для яких додатково виконуються такі умови:

- 40% балів: $2 \leq N \leq 15$.
- 30% балів: $15 < N \leq 3000$.
- 30% балів: $3000 < N \leq 10^5$.

Крім того, у тестах на 25% балів правильна відповідь не перевищує 4.

Приклади вхідних та вихідних даних

set.dat	set.sol
4	3
5 1000 5 5	
3	0
1 2 3	

Рекомендації щодо розв'язання

Під опуклим многокутником будемо розуміти строго опуклий многокутник, тобто такий, усі кути якого строго менші за 180° . Доведемо спершу допоміжне твердження, що є узагальненням нерівності трикутника для випадку опуклого N -кутника.

Лема 1. *Нехай задано $N \geq 1$ відрізків додатних довжин d_1, d_2, \dots, d_N . Із них можна скласти опуклий многокутник тоді й лише тоді, коли довжина найбільшо-*

го відрізка менша за суму довжин решти $N-1$ відрізків, тобто коли $2 \max\{d_1, d_2, \dots, d_N\} < d_1 + d_2 + \dots + d_N$.

Доведення. Нехай із заданих відрізків вдалося скласти опуклий N -кутник $A_1A_2 \dots A_N$. Хай, без утрати загальності, A_1A_N — найдовший з усіх відрізків. Послідовно використовуючи $N-2$ рази нерівність трикутника, матимемо

$$A_1A_2 + A_2A_3 + A_3A_4 + \dots + A_{N-1}A_N > A_1A_3 + A_3A_4 + \dots + A_{N-1}A_N > A_1A_4 + \dots + A_{N-1}A_N > \dots > A_1A_N.$$

Отже, нерівність з умови леми справджується.

Доведемо тепер зворотне твердження: якщо нерівність для відрізків довжин d_1, d_2, \dots, d_N справджується, то з них можна скласти опуклий багатокутник. Легко бачити, що при $N=1$ та $N=2$ нерівність виконуватися не може. Отже, $N \geq 3$. Не втрачаючи загальності, припустимо, що найдовшим із відрізків є d_N , тобто $d_i \leq d_N, 1 \leq i \leq N$.

Нехай на площині зафіксували коло радіуса $R \geq d_N/2$ із центром у точці O , а також деяку декартову систему координат, причому коло дотикається до прямої $x=0$ (вісь ординат) у точці $(0, 0)$ (початок координат). Визначимо на колі точки A_1, A_2, \dots, A_N так:

- A_1 — точка $(0, 0)$;
- A_2 — перша за рухом годинникової стрілки точка після A_1 на колі така, що $A_1A_2 = d_1$;
- A_3 — перша за рухом годинникової стрілки точка після A_2 на колі така, що $A_2A_3 = d_2$;
- ...
- A_N — перша за рухом годинникової стрілки точка після A_{N-1} на колі така, що $A_{N-1}A_N = d_{N-1}$.

Оскільки за побудовою діаметр кола не менший за кожен із відрізків d_1, d_2, \dots, d_{N-1} , конструкцію визначено коректно. Приклад розташування точок зображено на рис. 1.

Доведемо, що, взявши коло достатньо великого радіуса, ми зможемо забезпечити одночасне виконання таких двох умов:

- 1) $\angle A_1OA_2 + \angle A_2OA_3 + \dots + \angle A_{N-1}OA_N < 180^\circ$ (звідси, зокрема, випливатиме, що ламана $A_1A_2 \dots A_N$ не має самоперетинів);
- 2) $A_1A_N > d_N$.

Щоб довести цей факт, розглянемо довільні дві послідовні хорди $A_{k-1}A_k$ та A_kA_{k+1} , $2 \leq k \leq N-1$ (рис. 2). Нехай $\angle A_{k-1}OA_k = 2\alpha$, а $\angle A_kOA_{k+1} = 2\beta$.

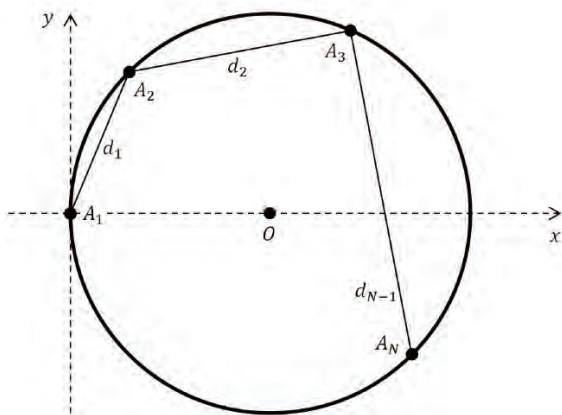


Рис. 1

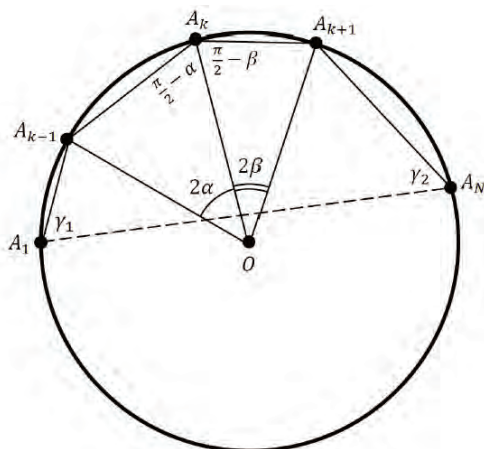


Рис. 2

Оскільки $OA_{k-1}=OA_k=OA_{k+1}=R$, трикутники $A_{k-1}OA_k$ та A_kOA_{k+1} рівнобедрені, а тому

$$\begin{aligned} \angle A_{k-1}A_kO &= \pi/2 - \alpha, \quad \angle A_{k+1}A_kO = \pi/2 - \beta, \\ \angle A_{k-1}A_kA_{k+1} &= \angle A_{k-1}A_kO + \angle A_{k+1}A_kO = \pi - \alpha - \beta. \end{aligned}$$

Як відомо, $A_{k-1}A_k = 2R \sin \alpha$, звідки

$$\alpha = \arcsin \frac{A_{k-1}A_k}{2R} = \arcsin \frac{d_{k-1}}{2R} \rightarrow 0, R \rightarrow \infty.$$

Аналогічно $\beta \rightarrow 0, R \rightarrow \infty$. Тому

$$\begin{aligned} \angle A_1OA_2 + \angle A_2OA_3 + \dots + \angle A_{N-1}OA_N &\rightarrow 0, R \rightarrow \infty; \\ \angle A_{k-1}A_kA_{k+1} &= \pi - \alpha - \beta \rightarrow \pi, R \rightarrow \infty. \end{aligned}$$

Отже, виконання умови 1) при достатньо великих значеннях R доведено. Щоб довести виконання умови 2), розглянемо багатокутник $A_1A_2 \dots A_N$ при значеннях R , що задовольняють умову 1). Сума кутів N -кутника складає, як відомо, $(N-2)\pi$. Якщо позначити кути $A_NA_1A_2$ і $A_1A_NA_{N-1}$ через γ_1 і γ_2 відповідно, матимемо

$$\begin{aligned} \gamma_1 + \gamma_2 &= (N-2)\pi - \sum_{k=2}^{N-1} \angle A_{k-1}A_kA_{k+1} \rightarrow 0, R \rightarrow \infty \Rightarrow \\ &\Rightarrow \max\{\gamma_1, \gamma_2\} \rightarrow 0, R \rightarrow \infty. \end{aligned}$$

Позначимо через $\varphi_i, 1 \leq i \leq N-1$, кут, який утворюють прями A_iA_{i+1} та A_1A_N . Якщо прями паралельні, цей кут дорівнює 0. Якщо точка перетину прями

лежить на промені A_NA_1 , то кут φ_i не перевищує γ_1 (дорівнює γ_1 , коли $i=1$, та менший за γ_1 , коли $i>1$). А якщо точка перетину прями лежить на промені A_1A_N , то кут φ_i не перевищує γ_2 (дорівнює γ_2 , коли $i=N-1$, та менший за γ_2 , коли $i<N-1$). Отже, $\varphi_i \leq \max\{\gamma_1, \gamma_2\}, 1 \leq i \leq N-1$. А тому

$$\begin{aligned} A_1A_N &= \sum_{i=1}^{N-1} A_iA_{i+1} \cos \varphi_i \geq \sum_{i=1}^{N-1} A_iA_{i+1} \cos \max\{\gamma_1, \gamma_2\} \rightarrow \\ &\rightarrow \sum_{i=1}^{N-1} A_iA_{i+1} = \sum_{i=1}^{N-1} d_i > d_N, R \rightarrow \infty. \end{aligned}$$

Отже, виконання умови 2) при достатньо великих R також доведено.

Візьмемо деяке достатньо велике число $R_1 > d_N/2$, для якого умови 1) і 2) виконані, і почнемо неперервно зменшувати радіус кола. При цьому точки A_2, A_3, \dots, A_N «ковзатимуть» по колу в напрямку годинникової стрілки. Припинимо процес «стискання» кола, щойно справдиться хоча б одне з двох тверджень:

- радіус кола стане рівним $d_N/2$ або
- точка A_N збігатиметься з точкою A_1 .

Ураховуючи природу першої умови зупинки, рано чи пізно ми обов'язково припинимо процес. Позначимо радіус кола, на якому відбулася зупинка, через R_0 . Нехай функція $f: [R_0, R_1] \rightarrow [0, +\infty)$ визначає відстань між точками A_1 та A_N при заданому радіусі $R \in [R_0, R_1]$. Незалежно від того, яка з умов послужила причиною зупинки, виконується нерівність $f(R_0) \leq d_N$. Справді, якщо $R_0 = d_N/2$, то відстань між будь-якими двома точками на колі не перевищує $2R_0 = d_N$. А якщо $A_N = A_1$, то взагалі $f(R_0) = 0$. У той же час із визначення числа R_1 маємо $f(R_1) > d_N$. Оскільки функція f неперервна, з цих співвідношень випливає, що існує число $R \in [R_0, R_1]$, для якого $f(R) = d_N$. Зважаючи на те, що це значення R ми «пройшли» до того, як точка A_N уперше збіглася з A_1 , можемо стверджувати, що при відповідному радіусі кола замкнена ламана $A_1A_2 \dots A_N$ — уписаний (а отже, опуклий) N -кутник без самоперетинів із довжинами сторін, що дорівнюють d_1, d_2, \dots, d_N . Лему доведено.

Повернімося тепер до самої задачі — нехай N знову позначає кількість заданих у вхідному файлі довжин, а через M позначимо найбільше значення, якого може набувати довжина одного відрізка (в даному випадку $M = 10^9 - 1$).

Ідейно найпростіший спосіб розв'язати задачу — перебрати всі підмножини заданої множини відрізків і, скориставшись твердженням леми, для кожної підмножини визначити, чи задає вона сторони опуклого багатокутника. Час виконання такого алгоритму дорівнює $O(N \cdot 2^N)$; або $O(N^2 \cdot 2^N)$, якщо перевіряти виконання нерівності багатокутника не лише для найбільшої, а для всіх його сторін. Перебірні алгоритми набирають 40 % від загальної кількості балів.

Задачу можна розв'язувати так. Спершу відсортуємо заданий масив чисел. Позначимо послідовність

чисел, утворену в результаті, як $d_1 \leq d_2 \leq \dots \leq d_N$. Підрахуємо суму $S_N = d_1 + d_2 + \dots + d_N$. Якщо подвоєне найбільше число в масиві d_N є меншим за цю суму (тобто задовольняє нерівність многокутника), то відповідь — N . Інакше незалежно від того, які відрізки стануть у підсумку сторонами многокутника, відрізка довжини d_N бути серед них не може, адже нерівність многокутника це число задовольнити ніяк не зможе. Тому відкинемо його та, перерахувавши суму $S_{N-1} = d_1 + d_2 + \dots + d_{N-1} = S_N - d_N$, повторимо ті самі дії для числа d_{N-1} : якщо $2d_{N-1} < S_{N-1}$, то з відрізків довжин d_1, d_2, \dots, d_{N-1} можна утворити опуклий многокутник, а тому відповідь — число $N-1$. А якщо $2d_{N-1} \geq S_{N-1}$, то відрізок довжини d_{N-1} не може бути стороною многокутника, тож слід відкинути число d_{N-1} і перерахувати суму $S_{N-2} = S_{N-1} - d_{N-1}$. Такі операції повторюємо доти, доки не знайдемо відповідь або не відкинемо всі відрізки. Останнє означатиме, що із заданих відрізків скласти опуклий многокутник неможливо.

Під час підрахунку суми чисел треба зважити на те, що вона може виявитися досить великою і, на відміну від самих чисел, не вміститься у чотирибайтову змінну. Тому слід або використати відповідний тип даних, або припиняти підрахунок суми, коли стає зрозуміло, що вона більша за $2M$, а отже, перевищує подвоєну довжину будь-якого із заданих відрізків.

Складність наведеного алгоритму лінійна, якщо знехтувати сортуванням масиву на початку виконання. Отже, залежно від реалізації сортування можна досягти ефективності $O(N^2)$ і набрати 70% балів або вкластися в $O(N \log N)$ і заробити повний бал.

Інший ефективний спосіб розв'язати задачу ґрунтується на твердженні такої леми.

Лема 2. Перед тим як вищенаведений алгоритм завершить свою роботу, він відкине щонайбільше $\lceil \log_2 M \rceil + 2$ відрізки (де $\lceil \log_2 M \rceil$ позначає найбільше ціле число, що не перевищує $\log_2 M$).

Доведення. Нехай алгоритм відкинув l відрізків завдовжки $d_N, d_{N-1}, \dots, d_{N-l+1}$, $l \leq N$.

Якщо $l < N$, то маємо $d_1 \geq 1, d_2 \geq 1, \dots, d_{N-l} \geq 1$, а далі, враховуючи, що наступні числа алгоритм відкинув,

$$d_{N-l+1} \geq d_1 + d_2 + \dots + d_{N-l} \geq N-l,$$

$$d_{N-l+2} \geq d_1 + d_2 + \dots + d_{N-l} + d_{N-l+1} \geq$$

$$\geq (N-l) + (N-l) = 2(N-l),$$

$$d_{N-l+3} \geq d_1 + \dots + d_{N-l} + d_{N-l+1} + d_{N-l+2} \geq$$

$$\geq (N-l) + (N-l) + 2(N-l) = 2^2(N-l),$$

...

$$d_N \geq (1+1+2+2^2+\dots+2^{l-2})(N-l) = 2^{l-1}(N-l).$$

Тому $M \geq d_N \geq 2^{l-1}(N-l) \geq 2^{l-1}$, звідки $l \leq \lceil \log_2 M \rceil + 1$.

Якщо $l = N$, то маємо $d_1 \geq 1, d_2 \geq 1$ а далі

$$d_3 \geq d_1 + d_2 \geq 2.$$

$$d_4 \geq d_1 + d_2 + d_3 \geq 1 + 1 + 2 = 2^2.$$

$$d_5 \geq d_1 + d_2 + d_3 + d_4 \geq 1 + 1 + 2 + 2^2 = 2^3.$$

...

$$d_N \geq 1 + 1 + 2 + 2^2 + \dots + 2^{N-3} = 2^{N-2}.$$

Тому $M \geq d_N \geq 2^{N-2} = 2^{l-2}$, звідки $l \leq \lceil \log_2 M \rceil + 2$.

Лему доведено.

Отже, алгоритм, який не сортуватиме масив, а просто щоразу після відкидання елемента заново шукатиме найбільше число, буде мати час виконання порядку $O(N \log M)$. Така ефективність теж оцінюється повним балом.

Однак твердження леми 2 дозволяє побудувати навіть швидший метод розв'язання задачі, який учасникам зовсім не обов'язково було втілювати, щоб отримати повний бал, але який ми наведемо з теоретичного інтересу.

Не будемо сортувати масив, а натомість за лінійний час упорядкуємо елементи масиву так, щоб на останніх $m = \lceil \log_2 M \rceil + 3$ місцях стояли (у довільному порядку) саме ті числа, які б стояли на цих m місцях, якби масив було відсортовано в порядку неспадання. За лінійний час цю операцію дозволяють виконати так звані алгоритми вибору порядкових статистик (selection algorithms), деталі реалізації яких можна знайти в інтернеті. Після цього елементи на останніх m місцях слід відсортувати (за $O(m \log m)$ часу), а далі діяти так, начебто відсортовано увесь масив. Оскільки згідно з лемою 2 алгоритм відкине щонайбільше $m-1$ елемент, то до невідсортованої частини масиву він просто не дійде. Сумарний час виконання програми складатиме $O(N + \log M \log \log M)$, що на практиці означає лінійну (за кількістю чисел у вхідному файлі) ефективність.

Насамкінець кілька слів стосовно того, наскільки реально було написати розв'язок задачі учаснику, не знайомому до олімпіади із твердженням леми про нерівність многокутника. Думка автора така: виходячи з того, наскільки широко відомою є нерівність трикутника і наскільки просто обґрунтовується твердження про нерівність многокутника в один бік, учасник, пробуючи розв'язати задачу, цілком міг щонайменше висунути гіпотезу про нерівність многокутника. З огляду на засади проведення олімпіади, перевірка гіпотези не вимагала від учасника побудови доведення в інший бік: достатньо було написати програму, що ґрунтується на гіпотезі, здати її й подивитися на результат.

2. «Ковпачок» (Данило Мисак)

Нудьгуючи на одному з уроків, відмінник Петро П'яточкін придумав собі розвагу. Він замалював ручкою деякі клітинки прямокутного аркуша, вирваного із зошита, зняв із ручки ковпачок та поставив його на одну із зафарбованих клітин. Далі Петрик послідовно переставляє ковпачок з однієї замальованої клітинки на іншу замальовану клітинку, яка міститься в тому ж рядку або в тому ж стовпчику, що і попередня. Петрик вибрав деяку зафарбовану клітинку й хоче перемістити туди ковпачок із початкової клітини за якомога меншу кількість ходів.

Завдання. Напишіть програму **сар**, що за даними про розміри аркуша паперу, конфігурацію зафарбованих клітин, розміщення ковпачка та цільової клітини знайде найменшу можливу кількість переставлень, за які Петрик зможе перемістити ковпачок з по-

чаткової клітини до цільової, керуючись придуманими ним правилами.

Вхідні дані. У першому рядку вхідного файлу `cap.dat` записано два цілих числа: кількість рядків N і кількість стовпчиків M клітинок, із яких складається аркуш, $2 \leq M \leq N \leq 1000$. Кожен із наступних N рядків містить по M символів:

- **x** (маленька літера латинського алфавіту) — зафарбована клітина.
- **.** (крапка) — порожня клітина.
- **o** (маленька літера латинського алфавіту) — початкова клітина.
- **+** (плюс) — цільова клітина.

У вхідних даних задано рівно одну початкову та рівно одну цільову клітину.

Вихідні дані. Вихідний файл `cap.sol` повинен містити єдине число — найменшу кількість переміщень ковпачка, які потрібно зробити Петрику задля досягнення мети. Якщо ж відповідно до заданих правил не можна досягти цільової клітини, то слід вивести -1 .

Оцінювання. Набір тестів складається з 3 блоків, для яких додатково виконуються такі умови:

- 30% балів: $2 \leq M \leq N \leq 10$;
- 30% балів: $10 < M \leq N \leq 100$;
- 40% балів: $100 < M \leq N \leq 1000$.

Приклади вхідних та вихідних даних

cap.dat	cap.sol
3 2 x+ xx o.	2
4 4 .o.x x.x .x.x x.+.	-1

Рекомендації щодо розв’язання

Аркуш паперу можна подати як граф, вершинами якого є зафарбовані клітинки, а ребро між двома вершинами проведено тоді й лише тоді, коли відповідні їм клітинки містяться в одному рядку або в одному стовпчику. Природний підхід до розв’язання задачі — здійснити пошук у ширину на такому графі, починаючи з вершини, що відповідає початковій клітинці A , та знайти довжину найкоротшого шляху з неї до цільової клітини B . Однак час виконання такого алгоритму буде занадто великим, адже пошук у ширину працює $O(V+E)$ часу, де V — кількість вершин, а E — кількість ребер у графі. У нашому випадку кількість вершин не перевищує NM , але кількість ребер може досягати $NM(N+M-2)/2$ (і дорівнює цьому числу, якщо зафарбовано всі NM клітин). Тож час виконання програми складе $O(NM(N+M))$. Це дасть 60% від загальної кількості балів.

Суттєво пришвидшити алгоритм можна, зауваживши таке: якщо на деякому кроці пошуку в ширину ми розглядаємо клітинку C , а раніше вже було розглянуто хоча б одну клітинку D , що стоїть у тому ж рядку, що й клітинка C , то немає сенсу розглядати ребра графа, які сполучають клітину C з іншими клітинками в її рядку. Справді, припус-

тимо, що D — перша з клітинок у даному рядку, пройдена алгоритмом. Якщо деяке ребро (C, E) сполучає C з клітинкою E , яка міститься в тому ж рядку, що й C , то або $E = D$, або у графі наявне ребро (D, E) (бо D та E містяться в одному рядку). А отже, вершину E вже було додано в чергу або навіть пройдено раніше — і розгляд ребра (C, E) не дасть жодного результату. Отже, можна не розглядати багато «зайвих» ребер — достатньо пам’ятати клітини, у яких рядках алгоритм уже встиг пройти.

Звичайно, ті самі міркування справедливі не тільки для рядків, але й для стовпчиків таблиці. Тому в кожному рядку буде розглянуто щонайбільше $M-1$ ребро, а в кожному стовпчику — не більше ніж $N-1$ (ребра, що виходять з деякої однієї клітини даного рядка або стовпчика). Усього алгоритм розгляне $O(NM)$ ребер, а тому й час виконання дорівнюватиме $O(NM)$. Це дасть повний бал.

Задачу можна розв’язувати дещо інакше. Знов побудуємо граф, що відповідає таблиці у вхідному файлі, але тепер зробимо це іншим способом. Новий граф матиме $N+M$ вершин і не більше ніж NM ребер. Нехай N вершин a_1, a_2, \dots, a_N графа відповідають рядкам $1, 2, \dots, N$, а інші M вершин b_1, b_2, \dots, b_M відповідають стовпчикам $1, 2, \dots, M$ таблиці. Ребро між вершинами a_i та b_j проведено тоді й лише тоді, коли на перетині i -го рядка та j -го стовпчика стоїть зафарбована клітинка. Інших ребер у графі немає.

Поставимо у відповідність ходу з клітинки (i, j_1) у клітинку (i, j_2) таблиці переміщення з вершини a_i у вершину b_{j_2} графа, а ходу з клітинки (i_1, j) у клітинку (i_2, j) — переміщення з вершини b_j у вершину a_{i_2} . Оскільки в оптимальному (найкоротшому) маршруті між клітинками A і B не може бути двох послідовних ходів у межах одного рядка або двох послідовних ходів у межах одного стовпчика (інакше їх можна було б замінити на один еквівалентний хід), то оптимальному маршруту між A і B відповідає деякий шлях по ребрах графа. При цьому якщо $A=(i_1, j_1)$ а $B=(i_2, j_2)$, то шлях починається або у вершині a_{i_1} , або у вершині b_{j_1} , а закінчується ребром, що сполучає вершини a_{i_2} та b_{j_2} (або в напрямку від вершини a_{i_2} до b_{j_2} , або в протилежному). І навпаки: кожному такому шляху по ребрах графа відповідає маршрут тієї ж довжини між клітинками A і B .

Отже, задача зводиться до пошуку довжини найкоротшого шляху, що починається в одній з вершин a_{i_1} або b_{j_1} , а закінчується ребром, що сполучає вершини a_{i_2} та b_{j_2} графа. Якщо позначити через $d(x, y)$ довжину найкоротшого шляху між вершинами x та y , шукана величина дорівнює

$$\min \{d(a_{i_1}, a_{i_2}), d(a_{i_1}, b_{j_2}), d(b_{j_1}, a_{i_2}), d(b_{j_1}, b_{j_2})\} + 1.$$

А якщо між відповідними парами вершин шляхів не існує, то немає й маршруту між клітинками A і B .

Зауважимо, що відстані між парами вершин можна підрахувати з допомогою двох пошуків у ширину (один з вершини a_{i_1} , а інший — із b_{j_1}). Утім, для зна-

ходження відповіді можна обійтися й одним пошуком у ширину. Для цього додамо до графа фіктивну вершину c й проведемо з неї два ребра: у вершини a_{i_1} та b_{j_1} . Тоді шукане значення — це довжина найкоротшого шляху з вершини c в одну з вершин a_{i_2} або b_{j_2} , тобто $\min \{d(c, a_{i_2}), d(c, b_{j_2})\}$.

Отже, маємо альтернативний алгоритм розв'язання задачі з таким же порядком часу виконання $O(NM)$.

3. «Тетрис» (Сергій Нагін)

Тінкі-Вінкі грає у модулярний тетрис. Поле складається з N стовпчиків, у кожному з яких може міститися від нуля до трьох кубиків. Після того, як у стовпчику опиняється четвертий кубик, усі чотири кубики зникають. За один хід гравець може вибрати довільну кількість від 1 до N послідовних стовпчиків, на які впаде по одному кубику, як зображено на рисунку. Тінкі-Вінкі хоче, починаючи з наявної конфігурації кубиків на полі, якомога скоріше досягти певної цільової конфігурації.

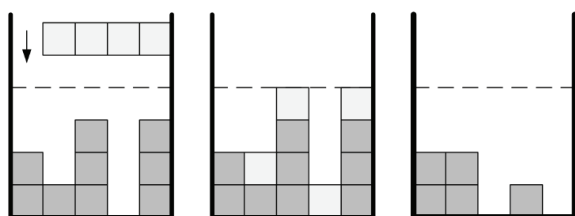


Рис. 3

Завдання. Напишіть програму **tetris**, яка за інформацією про кількість стовпчиків на полі, початкову та цільову конфігурації кубиків визначить найменшу кількість ходів, які має зробити Тінкі-Вінкі.

Вхідні дані. У першому рядку вхідного файлу **tetris.dat** міститься ціле число N ($1 \leq N \leq 1000$) — кількість стовпчиків на полі тетриса. У другому рядку записано N цілих чисел від 0 до 3, які задають початкову конфігурацію кубиків на полі. У третьому рядку записано N цілих чисел від 0 до 3, які задають кінцеву конфігурацію кубиків. Початкова та кінцева конфігурації не збігаються.

Вихідні дані. Єдиний рядок вихідного файлу **tetris.sol** повинен містити єдине ціле число — мінімальну можливу кількість ходів Тінкі-Вінкі для досягнення цільової конфігурації.

Оцінювання. Набір тестів складається з 4 блоків, для яких додатково виконуються такі умови:

- 30 % балів: $N \leq 8$.
- 20 % балів: $N \leq 1000$ та правильна відповідь не перевищує 10.
- 25 % балів: $N \leq 100$.
- 25 % балів: немає додаткових обмежень.

Приклади вхідних та вихідних даних

tetris.dat	tetris.sol
5	1
2 1 3 0 3	
2 2 0 1 0	
4	5
0 1 2 3	
3 2 1 0	

Рекомендації щодо розв'язання

Позначимо початкову конфігурацію як A , а кінцеву — як B .

Розв'язок за $O(4^N \cdot N^3)$. Нехай кожній конфігурації кубиків відповідає вершина графа, а між конфігураціями, між якими можна зробити перехід за один хід, є орієнтоване ребро. Запустимо стандартний пошук у ширину від вершини, що відповідає конфігурації A , до вершини, що відповідає конфігурації B . Отримана найкоротша відстань і буде відповіддю.

Розв'язок за $O(N^3)$. Зрозуміло, що зміна порядку операцій не змінює результуючої конфігурації. Впорядкуємо ходи гравця — відрізки кубиків — за лівим краєм.

Будемо використовувати динамічне програмування: нехай $dp(pref, opened)$ — мінімальна кількість операцій, яка потрібна, щоб прирівняти конфігурації в перших $pref$ символах за умови, що на теперішньому кроці буде відкрито $opened$ відрізків від деяких попередніх операцій. Перехід можна зробити так:

- закінчимо деяку кількість попередніх операцій;
- почнемо деякі інші операції, ліві кінці яких будуть збігатися з теперішньою позицією;
- після перших двох пунктів числа в обох конфігураціях на теперішній позиції повинні збігатися.

Більш формально динаміку можна записати так ($deleted$ — кількість закінчених операцій; new — кількість розпочатих операцій):

- $dp(pref, opened) + new \rightarrow dp(pref + 1, opened - deleted + new): deleted \leq opened; new \leq 3;$
- $(A[pref+1] + opened - deleted + new) \bmod 4 = B[pref+1].$
- $dp(0, 0) = 0.$

Розв'язок за $O(N^2)$. Додамо оптимізацію до попереднього розв'язку — обмеження $deleted \leq 3$. Зрозуміло, що на жодному кроці не треба скасовувати більше ніж три операції, інакше їх не треба було б починати.

Неасимптотичні оптимізації. Замість операцій $\bmod 4$ (Pascal) та $\% 4$ (C++) можна використовувати операції $\text{and } 3$ (Pascal) та $\& 3$ (C++). Дана оптимізація може скоротити час виконання в 10 разів.

Також можна помітити, що ми або скасовуємо деякі операції, або створюємо нові, але не робимо цього одночасно. Скориставшись цим, можна зменшити час виконання в 4 рази.

Також можна перебирати тільки досяжні стани динаміки, це теж може скоротити час виконання в 4 рази.

4. «Камелот» (Роман Єдемський)

Король Артур вирішив зібрати лицарів задля термінової військової наради. На землях, якими правив Артур, розташовувалися оборонні фортеці, побудовані у формі кола, — такі вважалися найбільш неприступними. Певні фортеці були розташовані всередині інших, що забезпечувало їм ще більшу захищеність. Як тільки лицарі отримували повідомлення про набіг ворога, вони збиралися в фортеці, щоб захистити її.

ють наказ від Артура, вони вирушають у дорогу зі свого маєтку в супроводі охорони. Якщо шлях лицаря проходить ззовні фортеці всередину чи навпаки, лицар повинен заплатити данину за перетин брами. Кожна фортеця встановлює розмір данини за прохід однієї людини, тобто лицарю потрібно заплатити за себе та своїх охоронців.

Артур бажає вибрати таке місце проведення наради, щоб мінімізувати сумарні витрати лицарів, адже вони будуть відшкодовані з державної скарбниці. Місце проведення наради може розташовуватися будь-де на землях Артура, крім границь фортець. Крім того, Артур може зменшити свої витрати, скасувавши данину не більш ніж у K вибраних ним фортецях. На свій шлях з Камелота до місця наради Король нічого не витрачає, а всі лицарі завжди вибирають найдешевший маршрут.

Завдання. Напишіть програму `camelot`, яка за інформацією про карту земель Артура, розміри данини кожної з фортець, кількість охоронців у лицарів і кількість фортець, де данина може бути скасована за наказом Короля, знайде мінімальну кількість грошей, яку він має витратити, щоби провести нараду. Карту земель може бути подано як площину з колами, що задають фортеці, і точками, що задають маєтки лицарів.

Вхідні дані. Перший рядок вхідного файлу `camelot.dat` містить три цілих числа N, M і K ($2 \leq N \leq 35\,000$, $1 \leq M \leq 35\,000$, $0 \leq K \leq N$), де N — кількість фортець на землях Артура, M — кількість лицарів, викликаних на нараду, а K — кількість фортець, у яких Артур може скасувати данину. Наступні N рядків задають фортеці й містять по чотири цілих числа x, y, R, C ($-10^6 \leq x \leq 10^6$, $-10^6 \leq y \leq 10^6$, $1 \leq R \leq 2 \cdot 10^6$, $1 \leq C \leq 10^5$), де (x, y) — координати центра фортеці на мапі, R — радіус кола, що задає фортецю, а C — розмір данини з людини. Наступні M рядків задають інформацію про лицарів і містять по три цілих числа x, y, L ($-10^6 \leq x \leq 10^6$, $-10^6 \leq y \leq 10^6$, $1 \leq L \leq 10^5$), де (x, y) — координати маєтку лицаря на мапі, а L — кількість охоронців, що подорожують разом із лицарем, включаючи самого лицаря. Вхідні дані гарантують:

- жодні два кола, що задають фортеці, не мають спільних точок;
- жодні дві точки, що задають маєтки лицарів, не збігаються та не лежать на колах.

Вихідні дані. Єдиний рядок вихідного файлу `camelot.sol` повинен містити ціле число — мінімальну суму грошей, які Король Артур має витратити, щоб зібрати лицарів.

Оцінювання. Набір тестів складається з 6 блоків, для яких додатково виконуються такі умови:

- 10% балів: $N \leq 1000$, $M \leq 1000$ і немає фортець, що розташовані на території інших фортець;
- 10% балів: $N \leq 35\,000$, $M \leq 35\,000$ і немає фортець, що розташовані на території інших фортець;
- 10% балів: $N \leq 1000$, $M \leq 1000$, $K = 0$;
- 20% балів: $N \leq 35\,000$, $M \leq 35\,000$, $K = 0$;
- 15% балів: $N \leq 1000$, $M \leq 1000$;
- 35% балів: немає додаткових обмежень.

Приклад вхідних та вихідних даних

camelot.dat	camelot.sol
4 9 1	12
6 10 2 1	
5 4 2 1	
10 7 1 200	
7 7 7 1	
5 3 10	
6 10 1	
7 10 1	
10 7 1	
10 10 1	
9 11 1	
9 12 1	
13 1 1	
14 1 1	

Пояснення. Щоб досягти оптимальності витрат, Королю Артуру необхідно скасувати данину у фортеці 3 і зібрати нараду у фортеці 2.

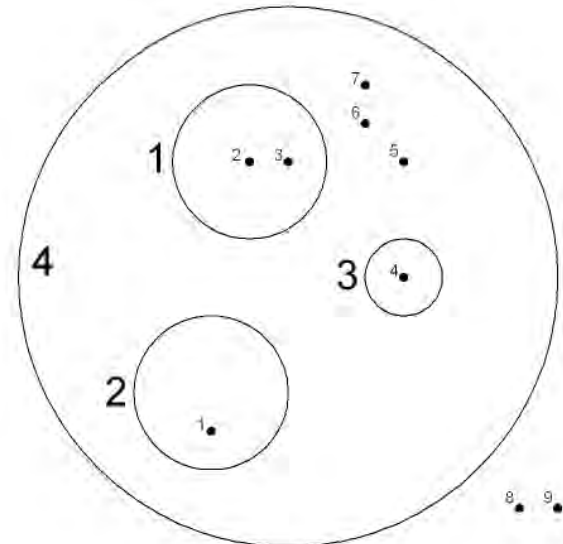


Рис. 4

Рекомендації щодо розв'язання

Значимо, що кола, які не перетинаються, утворюють дерево, причому кола відповідають ребрам, а області зв'язності — вершинам. Тому задачу можна переформулювати так: у кожній вершині дерева перебуває деяка кількість людей, що можуть переходити з вершини до вершини по ребрах, сплачуючи деяку суму грошей за цю операцію. Дозволено зробити ціну переходу нульовою не більш ніж для K ребер. Треба зібрати всіх людей в одній вершині, витративши на це якомога менше грошей.

Отже, задачу можна розбити на дві підзадачі: геометричну й оптимізаційну. *Геометрична підзадача* полягає у побудові дерева і визначенні початкової кількості людей у кожній вершині.

Геометрична підзадача: наївний розв'язок за $O(N^2 + NM)$. Для кожного кола знайдемо коло мінімального радіуса, що містить дане, і позначимо область зв'язності, що лежить між цими двома колами, як предка у дереві до вершини, що відповідає області зв'язності внутрішнього кола. У випадку, якщо немає кола, що містить задане, вважатимемо його предком деякий глобальний корінь нашого

дерева — вершину, що відповідає області зв'язності, яка лежить поза межами всіх кіл.

Геометрична підзадача: метод скануючої прямої (sweep line) за $O((N+M) \log(N+M))$. Для полегшення подальшого викладу будемо вважати, що на додаток до всіх кіл, що задані у вхідному файлі, ми маємо деяке коло із центром у точці $(0, 0)$ і радіусом таким, що всі інші точки та кола лежать усередині нього. Під час «сканування» ми не будемо розглядати події початку і закінчення цього кола, натомість дуги цього кола будуть постійно міститися у допоміжних структурах даних.

Будемо «сканувати» прямою площину зліва направо, підтримуючи у деякому бінарному збалансованому дереві верхні і нижні дуги кіл, що перетинають пряму в її поточному розташуванні. Ці дуги будемо впорядковувати у дереві за ординатою точки перетину з прямою. Кожну з можливих подій будемо опрацьовувати так.

Якщо скануюча пряма **натрапила на нове коло**, ми знаходимо у дереві дуг дугу, найближчу (або зверху, або знизу — неважливо) до точки дотику прямої і кола, й перевіряємо, чи коло, якому належить ця дуга, є таким, що містить нове коло. У випадку, якщо це так, позначаємо його «батьком» (предком) нового кола; якщо ні, то, як нескладно зрозуміти, батьком нового кола є батько кола, якому належала знайдена дуга. Після того як вершина — батько нового кола визначена, додамо дві його дуги до дерева дуг.

Якщо скануюча пряма **виходить за межі кола**, видалимо з дерева дуг вершини, що відповідають дугам цього кола.

Якщо скануюча пряма **натрапила на точку**, знаходимо коло мінімального радіуса, яке містить точку, у спосіб, аналогічний до наведеного вище методу знаходження батька нового кола: точку можна вважати колом радіуса 0.

Зауваження щодо реалізації дерева дуг:

- у мові C++ можна використовувати стандартний контейнер `set` із динамічним компаратором, що порівнює дуги за значенням y -координат точок перетину дуг зі скануючою прямою в її поточному розташуванні. При цьому є нюанс: дуги, що належать одному колу, треба порівнювати безпосередньо (нижня під верхньою), не порівнюючи y -координати цих дуг. Це пов'язано з тим, що в першій і в останній моменти, коли пряма перетинає коло, вона буде перетинати обидві дуги в одній і тій самій точці.

Існує також розв'язок, що не використовує динамічний компаратор: він порівнює дуги за відносним розташуванням кіл, до яких ці дуги належать. Такий розв'язок є більш ніж удвічі швидшим за розв'язок із динамічним компаратором.

До *оптимізаційної підзадачі* також є кілька підходів.

Оптимізаційна підзадача: випадок $K=0$. У цьому випадку в Артура немає можливості скасовувати плату за перехід по ребрах. Тоді єдине, що залежить від Артура, вибір вершини збору. Маємо наївний розв'язок з асимптотикою $O(N^2)$: перебрати варіанти ве-

ршини збору і для кожного варіанта за допомогою пошуку в глибину знайти ціну збору в даній вершині. Зробити це можна, рахуючи дві допоміжні величини: кількість людей у піддереві й сумарну ціну збору всіх людей з піддерева у корені цього піддерева. Такий розв'язок оптимізується до лінійного за допомогою підтримання під час пошуку в глибину описаних величин не тільки у піддеревах, а також і в наддереві (тобто в дереві без піддерева).

Оптимізаційна підзадача: розв'язання з допомогою структур даних. Нехай зафіксовано деяку вершину збору. Зорієнтуємо всі ребра так, як по них будуть йти люди до вершини збору. З цих міркувань для кожного ребра можна оцінити те, яку знижку отримає Артур, якщо зробить ціну проходу по даному ребру нульовою. Ця величина дорівнює кількості людей у відповідному піддереві, помноженій на ціну переходу. Так отримаємо розв'язок за $O(N^2 \log N)$: перебираємо можливі варіанти вершини збору й за допомогою пошуку в глибину вираховуємо знижки для кожного ребра; далі вибираємо K ребер, для яких знижки найбільші.

Для оптимізації цього розв'язку до $O(N \log N)$ подивимось, як змінюється множина знижок при перенесенні вершини збору до сусідньої вершини: змінюється лише знижка на ребро, через яке це перенесення здійснювалося. Отже, необхідно запровадити деяку структуру даних, що підтримує набір чисел (разом із дублікатами) і може швидко опрацьовувати запити двох типів:

- замінити деякий елемент x на y ;
- знайти K максимальних елементів.

Зауважимо, що перша операція еквівалентна послідовності операцій: додати елемент y до набору, видалити елемент x із набору. Цю структуру даних можна промодельовувати за допомогою контейнера `map` зі стандартних бібліотек C++, підтримуючи поточне значення K -го найбільшого елемента та кількість елементів, строго менших за K -й найбільший елемент. Альтернативою до цієї структури даних є дерево відрізків зі стисканням координат або розріджена його модифікація.

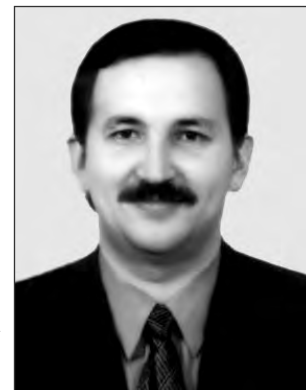
Оптимізаційна підзадача: розв'язання з допомогою жадібного алгоритму. Розглянемо деяке ребро (U, V) дерева. Нехай його ціна — $cost$. Позначимо кількість людей у піддереві вершини U через L , а у піддереві вершини V — через R . Доведемо, що до ціни збору це ребро додає $\min\{L, R\} \cdot cost$ грошей при оптимальному виборі вершини збору. Справді, нехай, наприклад, $L < R$, а вибрана вершина збору X міститься у піддереві L . Хай зібрати всіх людей з L у X коштує A , а зібрати всіх з R у V коштує B . Позначимо ціну проходу однієї людини по шляху з V у X через W . Тоді ціна збору наради у вершині X складає $A + B + W \cdot R$, а у вершині V — $A + B + W \cdot L$, що, очевидно, менше. Отже, X не є оптимальним вибором вершини збору.

Зауважимо, що оцінка $\min\{L, R\} \cdot cost$ не залежить від того, чи ми скасовували плату за перехід через інші ребра. Тому достатньо вирахувати дану величину для всіх ребер та вилучити K найбільших значень — сума решти і є відповіддю.

ДРУГА ВСЕУКРАЇНЬСЬКА ОЛІМПІАДА З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кузічев Микола Миколайович,

*керівник інформаційно-аналітичного центру управління освіти та науки
Дніпропетровської міської ради.*



У Дніпропетровську з 31 березня по 4 квітня 2013 року відбувся IV етап Всеукраїнської олімпіади з інформаційних технологій серед учнів. Починаючи з I-го етапу у 2012/2013 навчального року, в ній взяли участь 70920 учнів, із яких 29807 — з міських шкіл, 30320 — із сільських шкіл та 10793 зі спеціалізованих шкіл.

У IV етапі взяли участь 87 переможців обласних етапів (у минулому році — 57). Серед учасників олімпіади 12 дівчат та 75 хлопців, 1 учень сьомого класу, 7 учнів дев'ятого класу, 33 десятикласники та 46 учнів одинадцятих класів, з яких 25 — з міських шкіл, 8 — із сільських шкіл та 54 зі спеціалізованих шкіл. Переможцями олімпіади стали: Данило Омельченко (Дніпропетровськ) та Олег Павлюк (Харківська область). Дипломи I ступеня отримали 7 учасників, II ступеня — 15 учасників, III ступеня — 24 учасники.

У 2011 році дванадцятирічна авторська праця дніпропетровських освітян переросла у спільний інноваційний проект Міністерства освіти та науки України, Інституту інноваційних технологій і змісту освіти, Методичного центру управління освіти та науки Дніпропетровської міської ради.

Цього року олімпіада проводилася у 2 тури. Учасникам було запропоновано комплексне завдання і набір із чотирьох окремих задач, на виконання яких відведено 4 години у першому турі й 3 години у другому турі відповідно. Для розв'язання завдань учасники використовували програми Word, Excel, PowerPoint та Access.

Завдання першого туру значні за обсягом функцій, які треба було реалізувати і мали комплексний характер. Вони були розроблені з метою перевірки вміння учасників працювати з інформацією, аналізувати її, правильно використовувати для отримання кінцевого результату. Учасники мали використати як знання можливостей офісних систем, так і вміння логічно мислити в процесі застосування функцій та інструментів прикладних програм. Для чіткого визначення з умовами і результатами, які треба отримати, учасникам було запропоновано шаблони для виконання завдань.

Проблемним залишається питання формування змісту завдань із включенням тем відповідно до державної програми. З одного боку, необхідно включити до олімпіади всі розділи інформатики, з іншого, — неможливо вмістити весь спектр тем у час, відведений для двох практичних турів. Тому в моделі цього року було вирішено залишити тільки практичні завдання по роботі з інформацією в текстовому процесорі, табличному процесорі, системі управління базами даних, редакторі презентацій.

Завдання з використанням текстового процесора націлене на створення технології роботи з текстовою

інформацією і містило елементи роботи з графічною інформацією. Учасники виконали це завдання різними способами.

Слід зазначити, що досить складні завдання у табличному процесорі вимагали від учасників не тільки спеціальних знань і вмінь роботи з функціями Excel, але і значного часу для їх розв'язування. Уміння створити алгоритм рішення і застосувати його до подібних частин завдання є необхідним елементом в опрацюванні надлишкової інформації (наприклад, однотипні завдання на роботу зі зведеними таблицями і графіками). Оскільки досвід виконання олімпіадних завдань такого рівня в школах України вчителями й учнями тільки напрацьовується, досить багато учасників, які виконували завдання Excel, не змогли раціонально розподілити час на виконання завдання і, як наслідок, не встигли виконати інші задачі комплексного завдання.

За умовою завдання Access необхідно було на основі вхідних таблиць сформувати структуру бази даних, а подальша робота виконувалася не з розробленою структурою, а з вхідними таблицями, про що в умові авторами було наголошено. Оскільки постановка цієї задачі передбачала виконання двох різних типів робіт, учасники мали відповідно до умови виконати відповідні дії. Однак, судячи з перевірки робіт, досить багато учасників проігнорували вимоги завдання.

Завдання другого туру, які включали в себе завдання на знання й вміння використовувати можливості офісних систем були більше направлені на визначення вміння логічно мислити. Також здійснювалася перевірка знання спеціальних функцій і технологій застосування офісних систем. Ці завдання не носили комплексний характер і не були пов'язані з одна з іншою. На наступну олімпіаду заплановано створення тестового прикладу для перевірки роботи створених учасниками розв'язків, що зменшить тривалість і складність перевірки журі розв'язків учасників.

Найближчим часом планується створення офіційного сайту і збірника завдань олімпіади. Зважаючи на те, що серед усталених міжнародних предметних олімпіад відсутня олімпіада з інформаційних технологій, сподіваємось на підтримку Інституту інноваційних технологій і змісту освіти щодо спільного відпрацювання методології цієї олімпіади з подальшою роботою щодо ініціювання статусу міжнародної. Автори олімпіади впевнені, що у співпраці з учителями інформатики України і раціональним пропозиціям щодо організації олімпіади з інформаційних техноло-

гій набуде позитивного досвіду і стане повноправною міжнародною учнівською олімпіадою.

І ТУР

У найближчих планах дніпропетровців створення міського парку ракет. У ньому планується розмістити оригінальні ракети і космічні апарати в центрі міста. У місті, де працює Національний центр аерокосмічної освіти, діє планетарій, такий парк не може бути просто статичним подіумом для ракет. Цій ідеї може додати сучасності і креативу створення інтелектуальної довідкової системи, якою зможе скористатись будь-хто з відвідувачів: на великих екранах побачити 3D фільми про космічну галузь, на окремих електронних стендах ознайомитись з науковими досягненнями та інженерними розробками.

Учаснику олімпіади пропонується стати на один день учасником творчої групи зі створення моделі сучасної пошукової системи з історії світової космонавтики і ракетобудування.

Учаснику олімпіади необхідно розв'язати такі задачі виключно засобами MS OFFICE відповідно до умов задач. Використання VBA **заборонено!**

Умова задачі «Історія ракет-носіїв»

(Завдання виконується виключно засобами MS EXCEL, результат роботи учасника зберегти у файл **РН_історія.xlsx**).

Ракета-носіїв (РН, також ракета космічного призначення, РКН) — апарат, що діє за принципом реактивного руху (ракета) і призначений для виведення корисного навантаження в космічний простір. Першою у світі справжньою ракетою-носієм, що доставила в 1957 році вантаж на орбіту, була радянська Р-7 («Супутник»). Далі США і ще кілька країн стали т. зв. космічними державами, почавши використовувати власні ракети-носії, а три країни (пізніше також і Китай) створили РН для пілотованих польотів.

Для створення у парку ракет тематичного інформаційного стенду необхідно надати модель інформаційної системи, яка дозволить:

- переглядати аналітичну інформацію будь-яких 10 рядків, що розташовані поспіль, з таблиці, яка знаходиться на аркуші «орбітальні запуски 57_64»;
- проводити графічний аналіз інформації про запуски за класами ракет-носіїв по окремих країнах, окремих класах ракет і окремих ракет, що входять до певного класу;
- автоматично переглядати дані з успішності запусків у розрізі країн, категорій (ознак) успішності, дат, як за роками, так і за кварталами і місяцями;
- створювати динамічну демонстрацію вантажопідйомності ракет-носіїв;
- аналізувати розподіл кількості штучних супутників Землі по різних країнах.

Учасник олімпіади повинен візуалізувати інформацію про запуски ракет-носіїв засобами електронних таблиць. У файлі **Історія ракет.doc** надані інструкції для створення інформаційної панелі.

Увага! Кількість допоміжних аркушів для розміщення проміжних розрахунків не регламентується. У разі зміни вхідних даних у таблицях-розв'язках має змінюватися вихідна інформація.

Умова задачі «Пілотовані польоти»

(Завдання виконується виключно засобами MS ACCESS, дозволяється використання MS Word та MS Paint. Результат роботи учасника зберегти у файл **пілотовані польоти.accdb**)

За багато років до космічної ери люди мріяли про подорожі до зірок. У минулому сторіччі людина вперше здійснила подорож у космос. З того далекого 1961 року експедиції на навколосезну орбіту стали буденністю. Сучасна наука і техніка вже дозволяє досить тривалий час астронавтам проводити дослідження на космічних станціях.

Для знайомства відвідувачів парку ракет з історією пілотованих польотів планується створення декількох терміналів з довідковою системою про пілотовані польоти до космосу.

Учаснику олімпіади необхідно підготувати модель бази даних, за допомогою якої можна отримувати різноманітну інформацію за статистикою польотів.

Умова задачі «Лінза»

(Завдання виконується виключно засобами MS Word, MS PowerPoint. Результат роботи учасника зберегти у файли **Лінза.docx** та **Рух лінзи.pptx**)

У парку ракет на інформаційному стенді «Цивілізації» планується розмістити великий екран для виведення на нього текстів із джерел різних часів і народів про явища, події чи артефакти, пов'язані з космосом (з перекладом на сучасні мови).

Око людини природно сприймає інформацію не в прямокутному полі, й не однорідною за якістю зображення у різних областях огляду (файл **zip.jpg**), а вивчення складного тексту (давнього, складного за конфігурацією символів тощо) вимагає концентрації зору. Тому для якісного використання великого екрану авторам проекту парку запропоновано розробити спеціальний програмний засіб для детального ознайомлення і вивчення окремих фрагментів текстів. Однією з його функцій є виділення фрагменту тексту у вікні вибраної геометричної форми. Учаснику необхідно відповідно до зразка (**рух лінзи.wmv**) надати оргкомітету для подальшого замовлення програмістам модель інструмента «збільшувальне скло», яке може виділяти будь-який круглий фрагмент тексту без збільшення. Інструкції для виконання у файлі **око.docx**.

Увага! У розв'язку завдання дозволяється використовувати тільки файли з початковими даними, які розміщено у каталозі *Для учасника*, файли-розв'язки мають розміщуватись у каталогах відповідних задач.

Вхідні дані у завданні подані без зміни інформації з джерел зі збереженням авторського права (для підписів об'єктів і заголовків діапазонів надано словники відповідностей). Перевірка розв'язку учасника передбачає зміну вхідних даних і перевірку результату обчислень зі зміненими даними.

Усі завдання необхідно виконати за **4 години** і передати файли-розв'язки на підпис члену журі у каталозі з назвою **Прізвище_Імя_Регіон_Клас** (наприклад: **Петров_Олесь_Сумська_область_11** або **Сидоренко_Анна_Київ_10**).

Під час олімпіади учасник має право задавати запитання виключно у письмовому вигляді протягом першої години роботи. Запитання має бути сформульоване так, щоб на нього можна було відповісти однозначно **Так** чи **Ні**.

У випадку, якщо одна з цих відповідей може стати підказкою або відповідь міститься у тексті завдання учасник отримує відповідь **Без коментарів**.

У роботі залишати *відомості, які ідентифікують особу учасника* **Заборонено** (за виключенням назви каталогу з файлами-розв'язками учасника).

Розв'язок олімпіадного завдання вважається зданим тільки після перевірки членом журі в присутності учасника олімпіади наявності його файлів-рішень і підпису учасником у протоколі прийому робіт (підпис учасника ставиться після відмітки про наявність файлів-розв'язків кожної задачі).

II тур

Учаснику олімпіади необхідно розв'язати такі задачі виключно засобами MS OFFICE відповідно до умов задач. Використання VBA **заборонено!**

Умова задачі «Макрос» (20 балів)

Завдання виконується виключно засобами MS Word. Результат роботи учасника зберегти у файл з ім'ям Макрос.

У документі не більше 2000 символів, серед яких — довільна кількість пробілів. Створіть в режимі автозаписування макрос, який видаляє в документі всі зайві пробіли. У результаті має залишитися:

- а) між словами, не відокремленими знаками пунктуації, — по одному пробілу;
- б) після знаків пунктуації не в кінці абзацу — по одному пробілу;
- в) перед знаками пунктуації, крім тире, — жодного пробілу;
- г) перед тире — один пробіл;
- д) в кінці й на початку абзацу — жодного пробілу.

Макрос має викликатися сполученням клавіш **Ctrl+1** і міститися в файлі, що здається журі (не в шаблоні Normal.dot!). У кінці файлу слід також описати дію макросу у текстовому вигляді.

Зразок документа для застосування макросу міститься у файлі **word.doc**.

Умова задачі «Батьківство» (30 балів)

(Завдання виконується виключно засобами MS EXCEL, результат роботи учасника зберегти у файл **Батьківство.xlsx**)

У діапазоні A1:B10 вказано відомості про батьківство: у клітинки A2:A10 введено імена батьків, а у клітинки B2:B10 — імена їхніх синів. Різні люди мають різні імена. Користувач вводить у клітинку D2 ім'я батька, а в клітинках E2, E3, ... мають відобразитися імена його синів, причому між цими іменами не повинно бути порожніх клітинок (рис. 1).

Умова задачі «Залізниця» (40 балів)

(Завдання виконується виключно засобами MS PowerPoint. Результат роботи учасника зберегти у файл з ім'ям **Залізниця.pptx**).

Створити презентацію «залізниця», як у файлі **railway.exe**. Має зображуватися рух паровоза залізничною колією з 2 розгалуженнями. Біля розгалужень слід зобразити стрілки, які користувач може перемикаати клацанням. Результат перемикання має відбиватися на зображенні залізничної колії. У результаті перемикань паровоз спрямовується по одному з трьох маршрутів. Зображення паровоза під час руху має повертатися відповідно до вигинів залізничної колії. Рух паровоза має починатися після натискання кнопки **Поїхали**. Користувач під час одного перегляду презентації може перемикаати стрілки і запустити паровоз довільну кількість разів.

	A	B	C	D	E
1	Батько	Син		Батько	Сини
2	Гриць	Андрій		Гриць	Андрій
3	Пилип	Остап			Іван
4	Гриць	Іван			Потап
5	Микола	Микита			
6	Семен	Гриць			
7	Ілля	Пилип			
8	Микола	Василь			
9	Микита	Сашко			
10	Гриць	Потап			
11					

Рис. 1

чною колією з 2 розгалуженнями. Біля розгалужень слід зобразити стрілки, які користувач може перемикаати клацанням. Результат перемикання має відбиватися на зображенні залізничної колії. У результаті перемикань паровоз спрямовується по одному з трьох маршрутів. Зображення паровоза під час руху має повертатися відповідно до вигинів залізничної колії. Рух паровоза має починатися після натискання кнопки **Поїхали**. Користувач під час одного перегляду презентації може перемикаати стрілки і запустити паровоз довільну кількість разів.

Умова задачі «Розклад занять» (60 балів)

(Завдання виконується виключно засобами MS ACCESS. Результат роботи учасника зберегти у файл **Розклад занять.accdb**).

1) Створити базу даних за предметною галуззю «Розклад занять». Про вчителів відомі їхні імена, прізвища та номери паспортів, про предмети — назви і типи (природничий, гуманітарний тощо), про класи — назви і кількості учнів. Потрібно також зберігати відомості про те, який учитель, у який день тижня, на якому уроці, у якому класі, який предмет викладає. Один учитель може викладати кілька предметів, а в одному класі один той самий предмет — кілька вчителів.

Слід створити всі необхідні таблиці і зв'язки між ними з підтримкою обмежень цілісності й увести в кожну таблицю по 2–4 записи, зокрема 10-А клас, викладачів Петрова й Іванова, які викладають у цьому класі алгебру.

2) Створити запит «Визначити прізвища вчителів, які викладають у 10-А класі».

3) Створити запит «Визначити назви предметів, які вчитель Петров не викладає в 10-А класі».

Увага! У розв'язку завдання дозволяється використовувати тільки файли з початковими даними, які розміщено у каталозі **Для учасника**.

Перевірка розв'язку учасника передбачає зміну вхідних даних і перевірку результату обчислень зі зміненими даними.

Усі завдання необхідно виконати за 3 години і передати файли-розв'язки на підпис члену журі у каталозі з назвою **Прізвище_Ім'я_Регіон_Клас** (наприклад: **Петров_Олесь_Сумська_область_11** або **Сидоренко_Анна_Київ_10**).

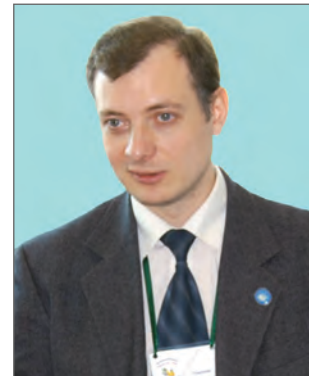
ЗАДАЧІ ХІ ВСЕУКРАЇНСЬКОГО ТУРНІРУ ЮНИХ ІНФОРМАТИКІВ

Мельник Валентин Іванович,

Учитель інформатики Полтавського обласного ліцею-інтернату для обдарованих дітей із сільській місцевості при Кременчуцькому педагогічному училищі імені А.С. Макаренка, заслужений учитель України,

Мотурнак Євген Володимирович,

учитель КЗО «Навчально-виховний комплекс №100 «Загальноосвітній навчальний заклад — ліцей» Дніпропетровської міської ради, переможець конкурсу «Учитель року 2013» у номінації «Інформатика», заслужений учитель України.



Формат турніру ми докладно розглядали у статті в №4 журналу «Комп'ютер у школі та сім'ї» за 2012 рік. Тож розглянемо більш докладно цього річн завдання турніру.

Завдання I етапу Всеукраїнського турніру юних інформатиків 2013

Планета Земля, колыска людства, уже досить переповнена. Усе частіше про колонізацію інших космічних тіл люди говорять не як про теоретичну, а про практичну можливість. Але умови цих космічних тіл зовсім не пристосовані для людини. Тому першопрохідцями, тими, хто все підготує до приходу людей, будуть роботи.

Нині роботи займають все більше місце у нашому житті. Спочатку дослідницькі, потім промислові, військові і, нарешті, побутові. Роботи-іграшки, роботи-друзі, роботи-пилососи — усе це ми можемо побачити на полицях магазинів. Розвиток усіх цих видів роботів відбувається паралельно. Вони працюють за однаковими принципами і взагалі між ними немає жодної різниці. Чи ні?

Відчуйте себе розробником робототехнічної системи. Виробництво роботів є досить коштовним, тому більшість попередніх досліджень варто здійснювати на моделях. Пропонується створити комп'ютерну модель робота згідно представлених завдань.

Завдання турніру носить дослідницький і творчий характер і тому є відкритим. Команди-учасниці деталізують завдання, виходячи зі своєї концепції розв'язання. Цю концепцію вони мають викласти у вигляді технічного завдання і надати журі разом з іншою технічною документацією у файлі **tech_zavd.docx (doc)**.

Кожне наступне завдання є ускладненням попереднього. Але команди-учасниці можуть комбінувати вимоги різних завдань у своєму програмному продукті. Усі математичні моделі, створені у рамках проекту, мають бути описані і збережені у файлі **mat_mod.docx (doc)**.

Завдання 1

1.1. Розробити програму — виконавець алгоритмів. Під виконавцем ми розуміємо комп'ютерну модель робота.

1.2. Програма повинна містити редактор алгоритму, яка підтримує систему команд для керування виконавцем, що рухається площиною. Систему команд учасники розробляють самостійно, виходячи з функціональної повноти та ефективності.

1.3. Реалізувати в програмі двовимірну візуалізацію виконання алгоритму, заданого користувачем.

1.4. Забезпечити збереження алгоритму в файлі.

1.5. Забезпечити зчитування алгоритму з файлу.

1.6. Надати користувачеві засоби налагодження алгоритму.

Завдання 2

2.1. Побудувати модель руху робота в тривимірному просторі поверхнею довільного рельєфу. За необхідністю розширити систему команд.

2.2. Забезпечити завантаження рельєфу поверхні з файлу зображення в сірих тонах у форматі BMP. Висота поверхні визначається яскравістю пікселя. Користувачеві надати можливість керувати масштабом відображення горизонтальних та вертикальних вимірів поверхні.

2.3. Забезпечити тривимірну візуалізацію моделі руху робота заданою поверхнею за побудованим користувачем алгоритмом.

2.4. Надати можливість зміни ракурсу, наближення та віддалення камери.

Завдання 3

3.1. Додати в модель можливість модульної побудови робота. Приклади модулів: шасі, сенсор доотику, сенсор відстані, сенсор орієнтації у просторі, акселерометр, сервомотор, маніпулятор та інші. Модель повинна враховувати взаємне положення модулів, їх орієнтацію у просторі.

3.2. Розробити програму — виконавець алгоритмів для модульного робота. Визначити відповідну систему команд.

3.3. Розробити формат файлу для збереження даних про робота. Забезпечити запис та зчитування цих даних.

3.4. Програма повинна містити редактор алгоритму, що підтримує систему команд для керування моделлю робота, що рухається поверхнею довільного рельєфу та може взаємодіяти з різними об'єк-

тами. Система команд повинна містити можливість опитування датчиків, керування сервомоторами, іншими приладами.

3.5. Врахувати можливість виходу з ладу модулів робота та його часткове руйнування.

Завдання 4

4.1. Додати у модель можливість роботи робота в умовах різних планет. Змодельовати систему живлення робота.

4.2. Змодельовати систему керування роботом із Землі або орбітальної станції.

4.3. Врахувати характеристики поверхні, якою рухається робот: вологість, слизькість, осипання ґрунту, буксування.

4.4. Врахувати вплив навколишнього середовища: умови освітлення, атмосферні явища, стихійні лиха тощо.

Завдання 5. Автономний робот

5.1. Запрограмувати робота на виконання завдань пунктів 5.2.–5.4. для довільного рельєфу.

5.2. Починаючи з точки з координатами (x_0, y_0) , переміститися в точку (x_1, y_1) .

5.3. Починаючи з точки з координатами (x_0, y_0) , забрати в точці (x_1, y_1) об'єкт та доставити його в точку (x_2, y_2) . Об'єкт має форму прямокутного паралелепіпеда та має характеристики: розміри rx, ry, rz ; одиничний вектор напрямку OY локальної системи координат об'єкта в системі світових координат (nx, ny, nz) .

5.4. Починаючи з точки з координатами (x_0, y_0) , відшукати об'єкт та перемістити його в точку (x_1, y_1) .

5.5. Забезпечити функціонування робота у заданий період часу N (днів, годин, хвилин, секунд) в умовах агресивного середовища.

Розбір завдання I туру

Позаяк основою завдання є проблема реального життя, для вирішення якої необхідно звернутися до одного з методів наукового дослідження — моделювання. За фабулу в 2013 році була взята робототехніка як галузь, що швидко розвивається уже нині, а з плином часу потреба у спеціалістах у цій галузі буде тільки зростати.

У завданні 1 необхідно було створити систему виконання алгоритмів з двовимірною візуалізацією. Як приклади можна було розглянути такі системи, як Logo, Roo, Kodu.

Така система повинна складатися з трьох основних частин:

1. Редактор алгоритмів.
2. Підсистема аналізу і виконання алгоритмів.
3. Візуалізація виконання алгоритму.

Найскладнішою тут є підсистема аналізу й виконання алгоритмів. Але, насправді, складність її програмування тісно пов'язана з підходом, який було використано у редакторі алгоритмів. Якщо було використано стандартну систему текстового запису алгоритмів розробленою мовою програмування, то, дійсно, необхідно було створити повноцінний синтаксичний аналізатор. А його створення — це досить нетривіальне завдання. Інший підхід — використання графічної системи запису алго-

ритмів є само собою доволі складним, але полегшує процедуру аналізу й виконання алгоритму.

Розроблена система команд оцінювалася з позицій ефективності та функціональної повноти. З одного боку, команд не має бути багато, з іншого, вони мають забезпечувати створення будь-якого алгоритму, як то — лінійного, розгалуженого, циклічного, і використання підпрограм, а також роботу з даними.

Окрема увага приділялася можливостям налагодження алгоритму, пошуку помилок, покрокового виконання.

Традиційно, завдання 2 — це перехід до тривимірної візуалізації. Наразі можна було як скористатися відомими бібліотеками тривимірної візуалізації, так і створити власну. З огляду на те, що завдання візуалізації тривимірних ландшафтів стало вже стандартним у комп'ютерному моделюванні, існує досить багато засобів його виконання. Але слід зауважити, що такі бібліотеки як OpenGL, DirectX призначені лише для роботи з графікою, фізичне моделювання має міститись в програмі користувача. А, наприклад, Unity 3D, містить і обмежені можливості фізичного моделювання. Слід розуміти природу цих обмежень.

Unity 3D створена насамперед для розробки комп'ютерних ігор, а не справжніх наукових інструментів, тому результат її використання тільки схожий на реальність, і навіть схожий не у всіх випадках. Складне фізичне моделювання, якого вимагає завдання 3, можна було б реалізувати, використавши такі професійні інструменти, як MatLab, але тут уже складність інструмента перевищує складність завдання.

Завдання 3 вимагало створення докладної моделі робота, який складався б з певних частин — модулів. Кожен з модулів реального робота має певне розташування відносно інших модулів робота, яке може змінюватись в разі рухомого модуля. Також модуль має масу, момент інерції, розміри, певну міцність та інші характеристики. Формат даних для збереження робота мав містити всю цю інформацію. Робот у зібраному стані — це цілісний фізичний об'єкт, який має центр мас, момент інерції, засоби руху поверхнею та інше. Взаємодія модулів між собою і з навколишнім середовищем і є предметом моделювання. Наприклад, якщо маніпулятор робота витягується вперед і підхоплює вантаж, центр мас робота зміщується, він може втратити рівновагу та перекинутись.

Завдання 4 — це перехід до більш реалістичної моделі, яка враховує систему живлення робота. Система живлення складається з трьох частин:

- підсистема отримання енергії;
- підсистема зберігання енергії;
- підсистема живлення модулів.

Кожен з модулів може бути джерелом енергії, контейнером для її зберігання, а також споживачем. У реальних робототехнічних системах енергетичний баланс дуже важливий. Падіння напруги акумулятора в процесі роботи може привести до зниження потужності сервомоторів, і, як наслідок, робота не

зможе підніматися по схилу, на який раніше легко заїжджав.

Також мають бути враховані характеристики ґрунту (вплив на рух), освітлення (вплив на датчики і сонячні батареї) та інші параметри зовнішнього середовища. Зокрема, слід було приділити увагу можливості руйнування робота під впливом навколишнього середовища.

Завдання 5 пропонувало низку тестових прикладів, на яких можна було перевірити функціонал робота і розробленої системи команд.

Загалом, детальні критерії оцінювання завдання І туру з боку досконалості й ефективності реалізації мали такий вигляд, як у табл. 1.

Таблиця 1

Завдання 1	
1.1	Наявність реалізованої програми-виконавця алгоритмів
1.2	Наявність опису системи команд
	Реалізація роботи з даними
	Реалізація лінійних алгоритмів
	Реалізація розгалужених алгоритмів
	Реалізація циклічних алгоритмів
1.3	Реалізація підпрограм
1.3	Реалізована двовимірна візуалізація
1.4	Можливість зберігати алгоритм у файлі
1.5	Можливість зчитувати алгоритм з файлу
1.6	Можливість покрокового виконання
	Можливість поблокового виконання
	Можливість перегляду значень параметрів робота користувачем
Завдання 2	
2.1	Наявність реалізованої програми-виконавця алгоритмів в 3D
	Візуалізація моделі робота в 3D
2.2	Можливість завантаження даних про рельєф з BMP
	Можливість задання горизонтального масштабу
	Можливість задання вертикального масштабу
2.3	Створена система 3D-візуалізації
	Використана готова бібліотека (OpenGL, DirectX,...)
	Використана бібліотека власного створення
2.4	Можливість зміни ракурсу перегляду
	Можливість наближення/віддалення камери
	Відсутність помилок відтворення зображення
Завдання 3	
3.1	Створена модель робота, що має модульну структуру
	Створені модулі введення даних (датчики)
	Створені модулі виведення даних (сервомотори, маніпулятори)
	Створені модулі керування

Продовження таблиці 1

3.1	Є можливість руху модулів відносно один одного
	Існує можливість редагування роботу в програмі
3.2., 3.4	Система команд враховує керування модулями
	Система команд включає введення даних з датчиків
	Система команд керування сервомоторами/іншими пристроями
3.3	Система команд включає перевірку систем робота
	Існує можливість запису робота до файлу
3.5	Існує можливість зчитування робота з файлу
	Врахована можливість руйнування робота
3.5	Врахована можливість руйнування окремих модулів
Завдання 4	
4.1	Змодельована система живлення робота в умовах автономності
	Враховані енерговитрати кожної дії робота
4.2	Змодельоване дистанційне керування роботом (із затримкою)
	Змодельовані можливі перешкоди зв'язку з роботом
4.3	Враховані вологість/слизкість ґрунту
	Враховане осипання ґрунту
	Враховане буксування
4.4	Врахований вплив умов освітлення
	Врахований вітер/поток рідини
	Враховані температура/атмосферний тиск/електромагнітні явища
Завдання 5	
5.1	Врахована взаємодія з об'єктами середовища
5.2	Виконане завдання для гладкого рел'єфу
	Виконане завдання для лабіриту
5.3	Виконане завдання для гладкого рел'єфу
	Виконане завдання для лабіриту
5.4	Виконане завдання пошуку
5.5	Робот виживає в умовах агресивного середовища

Завдання II етапу Всеукраїнського турніру юних інформатиків 2013 Індіана Джонс і пастка Майя

Усім вам добре відомі древні пророцтва племені Майя про кінець світу. На щастя, це пророцтво не збулось (або поки не збулось), але історія зовсім не про це...

Якось, відомий на весь світ шукач скарбів, Індіана Джонс потрапив до загадкової печери майя. Як веліли закони жанру кам'яні двері за ним зачинились і шляху назад не було. На щастя в печері горіло кілька ліхтарів і з малюнків на стінах професор Джонс зрозумів, що йому треба зробити, щоб відкрити шлях до подальших пригод!

Виявилось, що печера є прямокутником $N \times M$ клітинок. У деяких клітинках розташовані важкі каменюки, які Індіана Джонс не може ні посунути,

ні підняти. В інших клітинках стоять кілька тотемів язичьких богів, які він може зсунути з місця. Також на підлозі він знайшов відмітки, на яких, судячи з настінних малюнків, і стояли ці тотеми. Для виходу з печери, нашому герою необхідно повернути всі тотеми на свої місця (вважається що всі місця і всі тотеми однакові). Він може виконувати лише такі дії:

- переходити в сусідню вільну клітинку (по горизонталі або по вертикалі);
- якщо у сусідній з Індіаною клітинці стоїть тотем і наступна в цьому напрямку клітинка вільна (там немає ні тотему, ні каменю, ні межі печери), то він може пересунути тотем у вільну клітинку.

Завдання

А. Допоможіть професору Джонсу написати програму, що дасть йому можливість знайти правильну послідовність дій для виходу з печери.

Б. Проявіть творчий підхід до задачі, спробуйте додати до своєї програми додаткові можливості та функції (можливо, дати можливість зміни певних правил).

В. Напишіть штучний інтелект, який за заданим початковим розміщенням виведе Індіану Джонса з пастки, приготованої Майя.

Технічні умови

Передбачити, що інформація про печеру зберігається у текстовому файлі в такому форматі:

- перший рядок містить два числа N, M — розміри печери. У наступних N рядках розміщено по M чисел, розділених пропусками, що відповідають клітинкам печери;
- 0 — вільне місце;
- 1 — камінь;
- 2 — тотем;
- 3 — відмітка;
- 4 — початкове положення Індіани.

Для перевірки надати папку з усіма необхідними для роботи програми файлами і документацією. Ім'я папки має відповідати назві команди.

Розбір завдання II туру

Як бачимо, завдання є ні чим іншим, як фабульним переформулюванням класичної комп'ютерної гри Сокобан. Якщо в першому турі перевірялася здатність команд-учасниць створити складну фізичну модель, то завдання другого туру вимагає лише акуратності програмування і чіткого планування роботи в умовах обмеженого часу.

Саме в цьому турі найяскравіше проявляється робота команди. Ролі, які можна запропонувати, як основні — це координатор, програміст, дизайнер, алгоритміст та секретар.

Координатор — слідкує за часом, розподіляє завдання, визначає пріоритети в роботі і приймає остаточне рішення. Це роль капітана, якщо він не є, наприклад, основним програмістом у команді. У завдання

програміста входить реалізація ідей команди у вигляді програмного коду й інтеграція в нього продуктів інших учасників — графіки, текстів, звуку тощо. Дизайнер займається розробкою графічного контенту і системи навігації. Секретар займається створенням документації, тестує створені фрагменти програми і розробляє довідкову систему. Алгоритміст має розробити ефективні алгоритми, які запрограмує програміст. Найбільш складною частиною його роботи є розробка алгоритму штучного інтелекту.

Наведемо детальні критерії, за якими оцінувалися повнота і досконалість реалізації програмного продукту (табл. 2).

Таблиця 2

Повнота реалізації	
1	Наявність програми
2	Наявність початкової заставки
3	Наявність довідки в програмі
4	Можливість зчитування конфігурації печери з файлу
5	Можливість керування головним героєм
6	Герой не проходить крізь камені, межі печери та тотеми
7	Тотем пересуваються за умовою
8	Розміщення всіх тотемів на відмітках відкриває вихід з печери
9	Додані додаткові рівні гри
10	Наявність редактора печер
11	Можливість запису печери в файл
12	Існує можливість зберігання/завантаження поточного стану гри
13	Графічне оформлення відповідає фабулі
14	Програма містить екран переможця
15	Наявна система штучного інтелекту для головного героя
16	Створене головне меню програми
	Творчий доробок
Досконалість реалізації	
1	Система штучного інтелекту розв'язує тривіальну задачу
2	Система штучного інтелекту розв'язує нетривіальну задачу
3	Програма працює без критичних збоїв
4	Програма перевіряє коректність введення початкових даних
5	Програма працює з комфортною швидкістю
6	Відсутні помилки відображення графічної інформації
7	Відсутні інші помилки

Завдання заочного туру Всеукраїнського турніру юних інформатиків наступного року буде доступне у вересні-жовтні. Слідкуйте за інформаційними листами.

Бажаємо успіхів у підготовці до ТЮІ-2014!



ІНФОРМАЦІЙНЕ УПРАВЛІННЯ ЯКІСТЮ ОСВІТИ В БАГАТОПРОФІЛЬНОМУ ЛІЦЕЇ

Моцик Богдан Васильович,
аспірант Інституту педагогіки НАПН України.



Багатопрофільний ліцей є навчальним закладом нового типу, який має певну специфіку як щодо цілей і змісту навчання, так і організації навчально-виховного процесу. Управління таким навчальним закладом, порівняно із звичайним загальноосвітнім навчальним закладом (ЗНЗ), ускладнене внаслідок наявності видів і засобів діяльності, які відсутні в інших закладах загальної середньої освіти.

Для проведення дослідження щодо визначення організаційно-педагогічних основ діяльності нами виокремлено два основних напрями: перший — виявлення і класифікація умов функціонування багатопрофільних ліцей, і другий — розкриття їх сутності й з'ясування специфіки управління якістю освіти у цих закладах. Такий підхід передбачав вивчення мети й змісту діяльності багатфункціональних ліцей, специфічних умов їх функціонування. Ми розглядали ліцей вказаної категорії як соціальні, матеріальні системи, які безпосередньо пов'язані із середовищем і мають певні статичні й динамічні характеристики. З іншого боку, такі ліцей — це складні, динамічні, соціально-педагогічні системи, характерними ознаками яких є наявність сукупності взаємопов'язаних елементів, що утворюють певну ієрархічну структуру і визначеність взаємозалежності елементів, відношень підпорядкування.

Перелік і зміст внутрішніх документів у ліцеях визначається Інструкцією Міністерства освіти і науки України про ведення ділової документації. Попри це, ліцей можуть самостійно розробляти Положення про власні структурні підрозділи чи інші аспекти діяльності (положення про центр дистанційного навчання, положення про науково-природничу лабораторію, положення про зимову залікову сесію) на підставі державних Положень або інструкцій і відповідно до діючого законодавства.

Керівникові загальноосвітнього навчального закладу необхідно знати джерела вироблення і надходження інформації до суб'єктів управління і навчально-виховний процес (НВП) багатопрофільного ліцею, наявні освітні інформаційні ресурси, можливості, які надають засоби ІТ для пошуку, класифікації та перетворення інформації; уміти формувати інформаційний ресурс у межах закладу й управляти ним.

Особливе значення інформаційний ресурс має під час розгляду суспільних інституцій як складної, відкритої, динамічної соціально-інформаційної системи, оскільки системотвірні властивості цього ресурсу відіграють визначальну роль у формуванні і саморозвитку цієї системи. Аналіз наукових праць, у яких висвітлено особливості інформаційних процесів як складників системи інформаційного забезпечення управління і практики управління якістю освіти в ЗНЗ,

дав змогу з'ясувати джерела інформації, розробити, обґрунтувати та здійснити їх класифікацію за такими основами як суб'єкти вироблення інформації і суб'єкти використання інформації, способи створення і надходження інформації до багатопрофільного ліцею.

Інформатизація діяльності ЗНЗ, яким є багатопрофільний ліцей, полягає в систематичному і доцільно організованому використанні суб'єктами управління матеріальних ресурсів і кадрового потенціалу на основі використання даних, отриманих з інформаційних ресурсів. Роль інформації, яка нині є стратегічним ресурсом нарівні з матеріальними, природними, людськими, фінансовими, енергетичними суттєво змінилась. Нині недостатньо мати певне матеріальне ресурсне забезпечення організації, необхідно знати, як найефективніше його використовувати і мати інформацію щодо актуальних технологій управління, інформаційних і комунікаційних технологій [1].

Застосування викладених вище міркувань до проблеми інформаційного управління ЗНЗ й у більш конкретному сенсі — інформаційного забезпечення управління якістю освіти у багатопрофільному ліцеї, може базуватись, на нашу думку, на підходах, сформульованих Т.О. Лукіною в роботі, присвяченій аналізу стану зовнішнього оцінювання результатів розвитку освіти як основи управління якістю освіти [2, с. 63–71].

У табл. 1, створеній на основі даних Т.О. Лукіної, нами виокремлено (курсивом) заходи, які вимагають використання інформаційних ресурсів для свого проведення або доведення даних, на підставі яких приймаються управлінські рішення, до керівників ЗНЗ.

Нині в Україні наявність у загальному доступі необхідних для інформаційного забезпечення управління ліцеєм цифрових ресурсів обласного рівня швидше виняток, ніж правило, а загальнодержавний ресурс, яким є портал МОН України, достатньо наповнений управлінською інформацією тільки рівня нормативно-правового забезпечення.

Інформаційне забезпечення управління якістю освіти на рівні навчального закладу є необхідною складовою управління. Перш, ніж проектувати і створювати систему управління якістю освіти на рівні навчального закладу, зокрема багатопрофільного ліцею, необхідно сформулювати цілі її застосування у межах місії навчального закладу.

Місія має бути сформульована так, щоб її можна було використати як інструмент під час перевірки різноманітних операційних рішень щодо проблеми управління якістю, а також як критерій для визначен-

Умови і заходи, необхідні для здійснення ефективного державного управління якістю ЗСО

Умови	Заходи
Структурні перетворення	<ol style="list-style-type: none"> 1. Реорганізація організаційної структури системи державного управління ЗСО, орієнтація його на управління якістю освіти; встановлення та впорядкування різних видів зв'язків – внутрішніх (<i>між окремими елементами, рівнями та підсистемами</i>) і зовнішніх – (<i>між компонентами внутрішньої структури системи і навколишнім середовищем</i>): іншими освітніми системами, громадськими організаціями, освітнім середовищем тощо. 2. Створення розгалуженої національної системи моніторингу якості ЗСО та єдиної системи інформаційного забезпечення органів державного управління ЗСО. 3. Розвиток інфраструктури системи освіти в регіонах
Законодавче забезпечення	<ol style="list-style-type: none"> 1. Створення відповідної законодавчої, нормативно-правової бази реформування системи ЗСО. 2. Розроблення й прийняття державного стандарту якості ЗСО. 3. Оновлення системи освітньої статистики, яке передбачає: <ul style="list-style-type: none"> • перегляд переліку показників і критеріїв якості ЗСО; • формування національної системи індикаторів якості ЗСО для оцінювання досягнутої якості й міжнародних порівнянь. 4. Удосконалення технології збирання, передавання й узагальнення статистичних даних
Процесуальні зміни	<ol style="list-style-type: none"> 1. Модернізація механізмів державного контролю й управління якістю ЗСО (перегляд процедур і технологій проведення акредитації, ліцензування навчальних закладів, атестації навчальних установ і медпрацівників). Оновлення форм управління, зокрема розвиток державно-громадської форми управління якістю ЗСО, особливо на вищих організаційних рівнях управління освітою. 2. Упровадження обов'язкової спеціальної фахової підготовки працівників органів державного управління якістю ЗСО всіх рівнів, що передбачає розвиток і модернізацію системи підготовки, перепідготовки та підвищення кваліфікації державних службовців (<i>оновлення змісту освіти, організація навчання на основі новітніх інформаційних і управлінських технологій, комунікативних методик, різноманітних активних тренінгів, які дали б змогу реально переорієнтувати практику державного управління ЗСО на управління її якістю</i>). 3. Створення сучасних потужних електронних банків статистичної інформації, відкритих баз даних щодо функціонування ЗСО; механізму управління, аналізу та корекції інноваційних процесів у галузі ЗСО

ня власне поняття «якість». Формулювання місії, отже, має відповідати вимозі антиципації перевірки здійснення, виконання, досягнення цілі.

Розв'язання зазначеної проблеми дещо спрощується наявністю зовнішніх імперативів — Державного стандарту освіти, вимог навчальних програм, систем ДПА і ЗНО тощо. Зовнішні щодо закладу освіти чинники роблять знанневий складник мети навчання досить прозорим для контролю. Внаслідок цього критеріальний апарат перевірки знаннєвого і компетентнісного складників якості освіти може базуватися на відповідних стандартах освітньої галузі, навчальних програмах, тобто використовувати унормовані на загальнодержавному рівні критерії.

Складнішим є виявлення і застосовування критеріїв доцільності управлінських рішень щодо якості освіти в тактичному плані, оскільки, як уже було зазначено, багатопрофільний ліцей відрізняється від звичайного ЗНЗ наявністю завдань і видів діяльності, які не завжди повністю й вичерпно прописано у загальнодержавних нормативних документах.

Необхідність інформаційної підтримки складників процесу управління на рівні ЗНЗ у формі програмно-апаратних комплексів викликає появу нового класу АСУ — систем, призначених для управління навчально-виховним процесом у закладі освіти. Особливостями впровадження зазначених систем у процес управління якістю освіти на рівні ЗНЗ є необхідність вирішення дуже різнопланових задач, які з точки зору математики є недостатньо детермінованими, ухвалення рішення відбувається здебільшого з використанням неповної системи початкових і крайових умов. Позаяк навчально-виховний процес в цілому є досить інерційним об'єктом керування («час відклику», за термінологією теорії управління, становить від кількох місяців до десяти років, якщо мати на увазі завершення повної загальної середньої освіти), тому пря-

ме застосування управління за кінцевим результатом (отримання суб'єктом навчання якісної освіти) здебільшого неефективне. Необхідність виокремлення підсистем, які мають менший час відклику, спонукають до пошуків шляхів виокремлення в об'єкті управління складників (підсистем управління, керованих об'єктів нижчого, ніж заклад освіти, рівня ієрархії): «клас протягом навчального року», «клас протягом уроку», «учень протягом навчального тижня» тощо, якими можна керувати за кінцевим результатом.

Разом з тим, залишаються задачі управління, які можна і необхідно розв'язувати з використанням систем управління з розімкнутим колом зворотного зв'язку, або використовуючи підходи стратегічного управління, управління за результатами аналізу передісторії об'єкта управління. З іншого боку, специфіка управління навчальним закладом допускає розподіл функцій управління, що суттєво полегшує розв'язання зазначених вище проблем.

Як було показано вище, явище «управління якістю освіти у багатопрофільному середньому навчальному закладі з використанням інформаційних технологій» є достатньо складним, тому на його моделі (рис. 1) відображено тільки найсуттєвіші складові й зв'язки, які умовно можна виокремити, виходячи з їх часових параметрів. Аналіз явища показав, що розв'язання проблеми слід шукати на шляху комплексного застосування можливостей автоматизованої системи управління (АСУ), зокрема в частині збирання, інтерпретації та подання даних для їх використання з метою покращення якості освіти. Можливе застосування всіх видів управління (стратегічного, за результатом, за відхиленням параметрів процесу, за відхиленням проміжного результату) з урахуванням об'єктів керування в багатопрофільному ліцеї, але на різних за тривалістю часових проміжках.

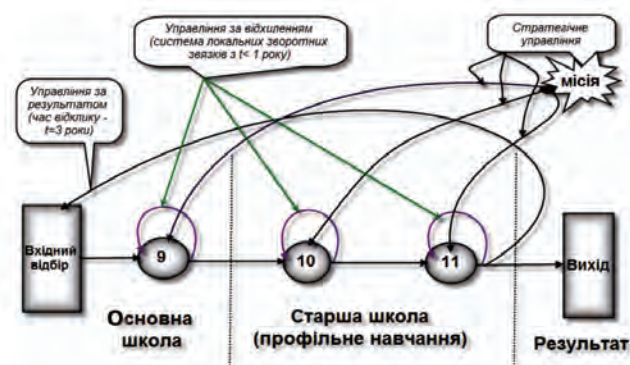


Рис. 1. Модель управління якістю освіти у ліцеї з використанням зворотних зв'язків різного рівня

Зокрема, управління НВП як складової підсистеми управління якістю освіти в ліцеї, з технологічної точки зору навчально-виховний процес можна уявити як керовану взаємодію його учасників і суспільного оточення.

Взаємодія супроводжується і підтримується інформаційними потоками, які з'єднують як безпосередніх учасників процесу, так і суспільне оточення із закладом навчання як цілісною системою, так і з учасниками навчально-виховного процесу як з особистостями.

Нові принципи, засоби, форми і методи управління закладом освіти потребують змін її змісту, потоків, масивів, форм одержання, опрацювання, передавання і використання інформації. Змінюється програма управління — відповідно змінюється й система інформаційного забезпечення. Розглянемо зміст інформації за визначеними блоками. Планувати й отримувати внутрішню соціально-педагогічну інформацію потрібно, виходячи з інформаційних потреб суб'єктів управління і навчально-виховного процесу закладу освіти, використовуючи специфічну для кожного об'єкта управління систему оцінювання стану системи і формування прямих впливів.

Орієнтовний перелік суб'єктів управління якістю освіти, які збирають, аналізують і приймають управлінські рішення в процесі управління якістю освіти в ліцеї розроблено, виходячи з аналізу діяльності ліцею. Визнано за необхідне виокремити такі суб'єкти управління:

- учні закладу освіти (ліцеїсти);
- органи учнівського самоврядування в ліцеї;
- учителі, педагоги-куратори;
- батьки;
- адміністративно-управлінська ланка;
- директор загальноосвітнього навчального закладу;
- заступники директора закладу освіти.

Теоретичною основою розробки змістової системи є системно-комплексний і кібернетичний підходи до управління закладом освіти і функціональний підхід до управлінської діяльності адміністративно-управлінської ланки. На основі цього розроблена модель інформаційного забезпечення управління якістю освіти ліцею, зображена на рис. 2.

У процесі дослідження створено бази даних, структуровані за допомогою інформаційних блоків за цілями і пріоритетними компонентами управління якістю освіти.

Освітньо-інформаційне середовище ліцею, незалежно від того, яку саме АСУ буде використано, має проектуватись з урахуванням поданих на рис. 2 складників й ієрархії моделі.

Перш за все, мають бути задоволені інформаційні потреби суб'єктів управління і НВП, зазначені в табл. 1. Незважаючи на те, що більшість сучасних АСУ, призначених для застосування на рівні закладу освіти, мають відповідну базу даних, слід подбати про їх актуалізацію й передбачити систему заходів, які б забезпечували неперервність цього процесу. Забезпечення інформаційних потреб суб'єктів управління і НВП засобами ІТ має враховувати необхідність подання актуальної інформації.

У системі інформаційного забезпечення управління якістю освіти ліцею (ІЗУЯОЛ) виокремлено чотири основних компоненти: концептуально-теоретичний, змістовий, організаційно-діяльнісний і техніко-технологічний.

Основним компонентом, на основі якого вибудовується вся система ІЗУЯОЛ, є концептуально-теоретичний, у якому визначено, на основі наукових підходів і суспільного замовлення, місію закладу освіти. Від чіткості і прозорості формулювання місії залежить можливість перевірки досягнення мети діяльності. Разом з тим, як було зазначено вище (див. рис. 1), формулювання місії закладу освіти не має бути надто жорстким, оскільки система ІЗУЯОЛ необхідно, як одну з умов ефективного управління, передбачає модифікування місії.

Не менш важливим як у стратегічному плані, так і з точки зору оперативного управління, прийняття тактичних рішень, є змістовий компонент, найважливішою вимогою до якого є забезпечення інформаційних потреб суб'єктів управління і НВП актуальною інформацією. Систематизація відомостей, які подаються суб'єктам управління, має виконуватись так, щоб мінімізувати їх надлишковість.

Організаційно-діяльнісний компонент підтримує власне процес управління на всіх часових проміжках і на всіх рівнях ієрархії. На рис. 1 це відображено зв'язками, які відображають управління за відхиленням і управління за результатом. Управління «за передбаченням», тобто стратегічне управління також здійснюється в межах і засобами зазначеного компонента, але, оскільки управління за повним результатом навчання має час запізнення результату від трьох років і більше, тому вплив сигналу зворотного зв'язку на якість освіти може відбуватись переважно шляхом удосконалення процесу і/або коригуванням місії, як показано на рис. 1.

Техніко-технологічний компонент забезпечує відстеження і фіксування проміжних результатів, локальний зворотний зв'язок. Час отримання повідомлення про відхилення процесу від норми — від кількох годин до десяти діб, час відклику на управляючий вплив (за результатами повторної перевірки) — кілька діб (якщо на навчальний предмет у робочому плані виділено один урок на тиждень, час відклику становитиме два тижні).

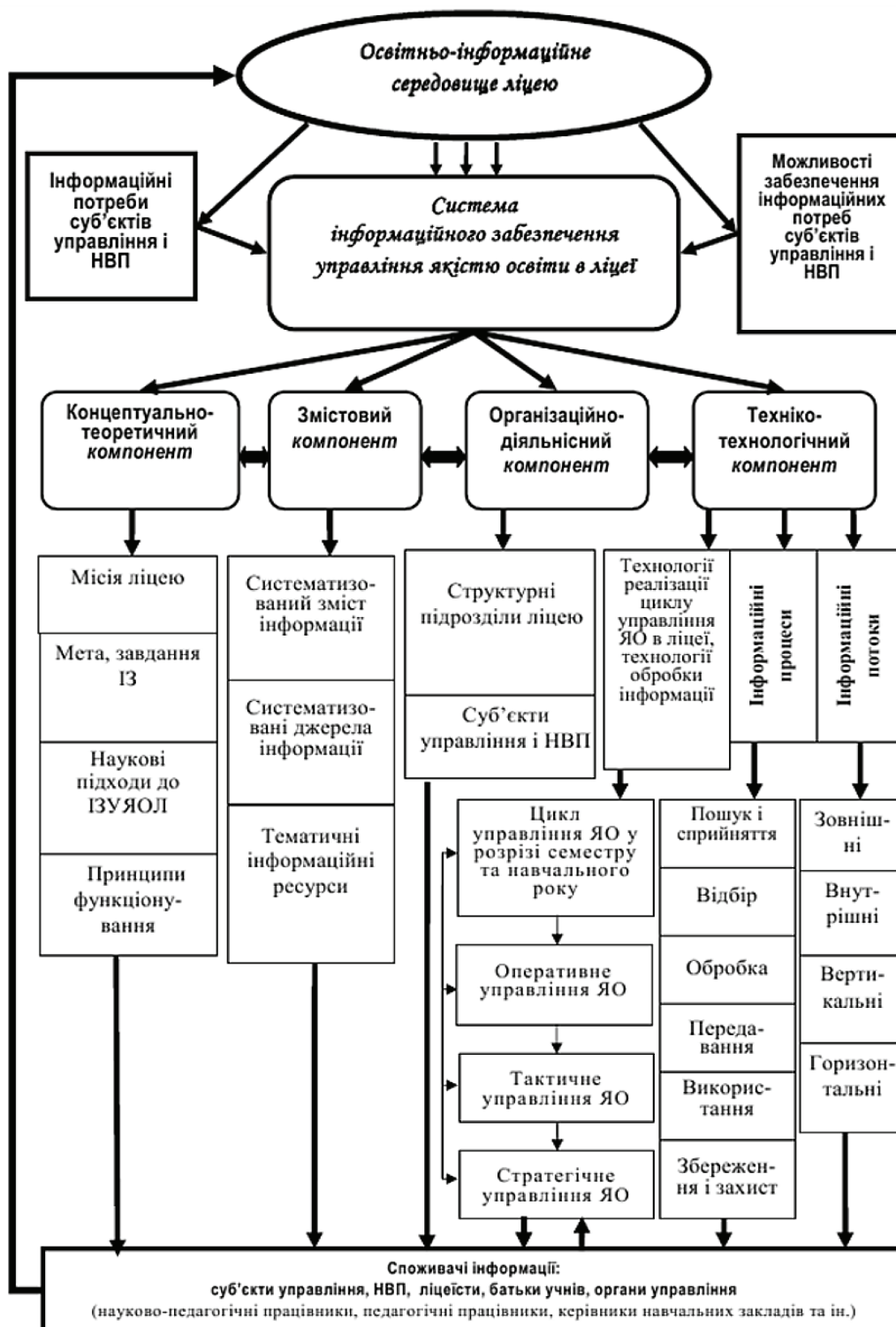


Рис. 2. Модель інформаційного забезпечення управління якістю освіти в багатoproфільному ліцеї

Важливість чіткого функціонування техніко-технологічного компоненту полягає в тому, що час отримання повідомлення про відхилення за локальним результатом становить від кількох діб до місяця, час відклику на управляючий вплив (за результатами повторної перевірки) — місяць і більше, залежно від моменту отримання повідомлення, яке є сигналом зворотного зв'язку.

Систематичний розгляд ІЗУЯОЛ й її застосування у НВП ліцею підтвердило припущення щодо доцільності її впровадження.

Література

1. Калініна Л. М. Стан інформаційного забезпечення управління якістю освіти загальноосвітнього навчального закладу / Л. М. Калініна, Б. В. Моцик // Педагогічний дискурс : зб. наук. праць / Ін-т педагогіки НАПН України, Хмельницька гуманітарно-педагогічна академія. — Хмельницький : ХГПА, 2010. — Вип. 8. — С. 81–86.
2. Лукіна Т. О. Державне управління якістю загальної середньої освіти в Україні [Текст] : дис... д-ра наук з держ. упр. : 25.00.02 / Тетяна Олександрівна Лукіна / Національна академія держ. управління при Президенті України. — К., 2005. — 526 с.

КОДУВАННЯ ЗВУКОВИХ ДАНИХ

Жугастрова Олена Вікторівна,

відповідальний секретар журналу «Комп'ютер у школі та сім'ї».

Анотація. У статті розглянута сутність аналогової і дискретної форми подання звуку, методи перетворення аналогової форми звуку в цифрову й навпаки, склад звукової системи комп'ютера і формати звукових файлів.

Ключові слова: звук, звукова плата, дискретна форма, амплітуда, частота, дискретизація, квантування.

Аналогова й дискретна форма звуку

Звукові дані нині зберігаються й опрацьовуються практично у будь-якому комп'ютері. Звук виникає внаслідок зміни тиску повітря з певною частотою. Частота вимірюється в герцах (гц), кілогерцах (Кгц) та інших одиницях (один гц дорівнює одному коливанню за секунду). Найкраща частота для сприйняття звуку людиною знаходиться в діапазоні від 20 гц до 4 Кгц, а максимальна частота відчуження звуку людиною досягає 20 Кгц. Звукові сигнали сприймаються й опрацьовуються мозком людини, на основі чого формуються певні поняття.

У комп'ютері звукова інформація представляється в аналоговій і дискретній формах за допомогою електричних сигналів. Для цього звукові хвилі перетворюються в електричні, наприклад за допомогою мікрофона, а електричні коливання перетворюються у звукові хвилі, наприклад, за допомогою звукових колонок. **Аналогова** форма — це така форма, коли електричні коливання з часом змінюються безперервно і має безліч значень. **Дискретна** форма — це форма, за якої електричний сигнал змінюється ступінчато і має кінцеву кількість значень. На рис. 1 наведена аналогова і дискретна форма електричного сигналу відповідного звуку. Аналогова форма подана у вигляді безперервної кривої лінії, а дискретна у ступінчатому вигляді.

Об'єкти природи, у тому числі людина, подають звук у вигляді аналогової (неперервної) хвилі. Такій хвилі відповідає коливання електричного сигналу, приклад якого наведено на рис. 2.

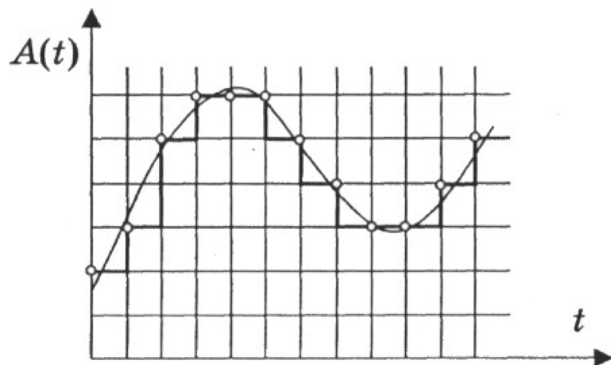


Рис. 1



Рис. 2

Електричний сигнал звукової хвилі характеризується багатьма параметрами, основними з яких є такі.

1. Амплітуда. Визначає гучність звуку. Чим більша амплітуда, тим більша гучність. На рис. 3 наведені звукові хвилі трьох різних амплітуд. Найбільшою є амплітуда на рис. 3в.

2. Частота коливань. Цієї фізичної характеристики звуку відповідає фізіологічна характеристика висоти тону. На рис. 4 наведені три хвилі різних частот. Найбільшою є частота хвилі, зображеної на рис. 4в.

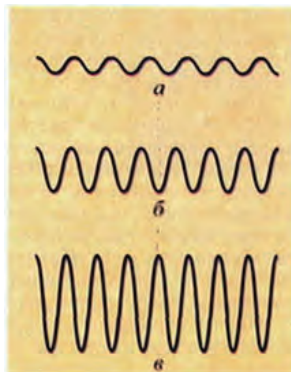


Рис. 3

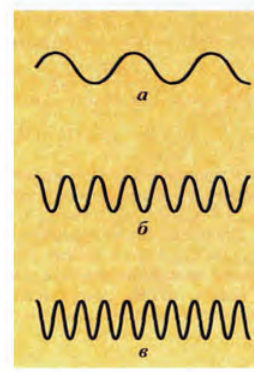


Рис. 4

3. Тембр (окраска звуку), визначається нотами різних інструментів.

Перетворення аналогового сигналу в цифрову форму й навпаки

Для збереження й опрацьовання звуку в комп'ютері його необхідно перетворити з аналогової форми в цифрову, тобто у певну сукупність двійкових чисел. Процес перетворення аналогового сигналу в цифрову форму називається **аналого-цифровим перетворенням** (оцифруванням). Алгоритм такого перетворення полягає у виконанні двох основних дій.

1. Виміри величини амплітуди аналогового сигналу через рівні проміжки часу. На рис. 5 наведено приклад виділення рівних проміжків часу сигналу певної форми. Процес такого перетворення називається

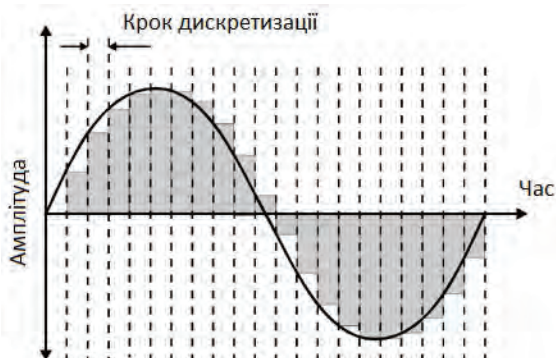


Рис. 5

дискретизацією, а проміжок часу, через який здійснюється вимір величини (амплітуди) сигналу, — кроком або частотою дискретизації.

2. Записування отриманих значень амплітуди в числової формі. Цей процес називається квантуванням. Приклад квантування сигналу наведено на рис. 6.

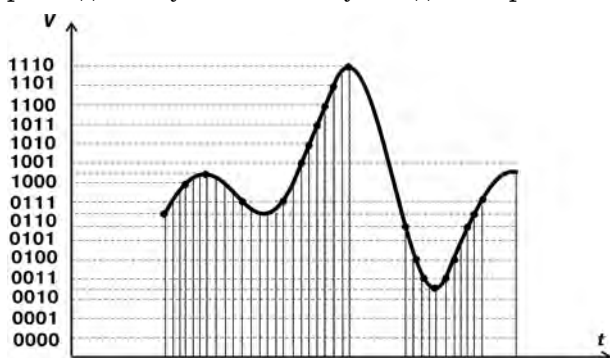


Рис. 6

З рисунка видно, що точність перетворення (квантування) амплітуди аналогового сигналу у цифрову форму залежить від двох основних факторів.

а) Від частоти дискретизації. Чим частіше вимірюється величина сигналу, тим точність квантування вища. У комп'ютерах поширеною є частота дискретизації 44 КГц, тобто амплітуда сигналу вимірюється 44000 разів за секунду. Така частота використовується, зокрема, для аудіо компакт-дисків. У сучасних комп'ютерах вона досягає 128 КГц і більше.

б) Від кількості двійкових розрядів, відведених для запису амплітуди сигналу. Чим більше використовується двійкових розрядів, тим точність квантування більша. На рис. 6 для квантування амплітуди сигналу відведено 4 двійкових розряди. Реально в комп'ютерах використовується 16 розрядів, що дозволяє фіксувати $2^{16}=65536$ рівнів квантування, або 24 розряди. Кожний біт кодового слова відповідає приблизно 6 дБ. У разі 16-розрядного квантування можна отримати діапазон гучності до 96 дБ.

Перетворення сигналу з цифрової форми в аналогову полягає в тому, що на основі цифрових даних здійснюється інтерполяція (згладжування) сигналу, за рахунок чого формується неперервний сигнал.

Технічна реалізація

Перетворення сигналу з аналогової форми у цифрову й навпаки здійснюється за допомогою звукових

карт, які виготовляються у вигляді окремих модулів, або розміщуються в чипсеті системної плати. Аналого-цифрові перетворювачі разом з цифро-аналоговими перетворювачами називають кодеками. Перша звукова карта була розроблена у 1989 році. Зовнішній вигляд однієї з них наведено на рис. 7.



Рис. 7

Крім звукової карти для відтворення звуку в комп'ютері застосовується мікрофон, звукові колонки, навушники та інші пристрої, які безпосередньо перетворюють звукові хвилі в електричні сигнали й навпаки. Ці пристрої входять до складу акустичної системи, які разом зі звуковою картою називають звуковою системою комп'ютера. Роз'єми звукової системи виведені на задню стінку системного блоку, приклад яких наведено на рис. 8.



Рис. 8

До лінійного входу може підключатися будь-яке джерело звукових сигналів, а до лінійного виходу — будь-який приймач звукових сигналів.

Зв'язки між основними пристроями звукової системи наведені на рис. 9.

Сигнали ступінчастої форми — це цифрові сигнали, а інші — аналогові. З рисунка видно, що з аку-

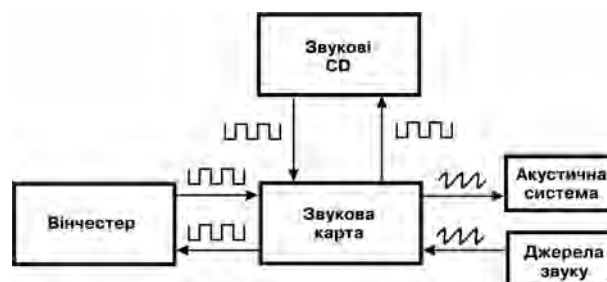


Рис. 9

стичної системи і в акустичну систему надходять аналогові сигнали. Наприклад, з мікрофона у звукову карту надходять аналогові електричні сигнали. Так само із звукової карти на колонки надходять електричні аналогові сигнали. Обмін звуковими даними між звуковою картою і вінчестером, а також між вінчестером і звуковими CD або DVD здійснюється тільки у цифровій формі.

Отже, у процесі функціонування звукової системи комп'ютера виникає постійна потреба у перетворенні електричних сигналів з аналогової форми у цифрову й навпаки. Цю функцію виконує звукова карта.

Формати звукових файлів

Для збереження звукових файлів потрібні великі обсяги пам'яті. Наприклад, для збереження файлу з 24-розрядною глибиною квантування і дискретизацією 44 Кгц звучанням протягом 3 хвилин потрібно $44000 \times 3 \times 60 \times 4$ байти = 31680000 байти = 31,68 МБ. Для зменшення обсягу пам'яті використовуються спеціальні програми, що здійснюють стиснення (компресію) звукових даних, за рахунок чого обсяг потрібної пам'яті можна зменшити до 12 разів. Однак стиснення даних з великим коефіцієнтом компресії може призвести до деякої втрати якості звуку. Спосіб стиснення даних визначає формат звукових файлів. Нині існує три основних типи форматів звукових файлів.

1. Формати без стиснення або з невеликим коефіцієнтом стиснення даних. Типовими форматами цього типу для ОС Windows є формат WAV, для ОС Mac — формат AIFF, а також внутрішні формати звукових редакторів, наприклад, Audacity. Перевага файлів цього формату полягає у високій якості збереження й відтворення звуку. Однак, цей формат потребує для збереження файлів великих обсягів пам'яті. Наприклад, для збереження стереозапису пісні протягом 6-ти хвилин з дискретизацією 44 Кгц і глибиною 16 біт потрібно приблизно 60 МБ.

2. Формати стиснення даних із втратами якості звуку. Це один з основних форматів звукових файлів. Окрім комп'ютерів стиснення з втратами використовується у потоковому аудіо в DVD, цифровому телебаченні й радіо, а також потоковому медіа в Інтернеті.

У файлах цього формату стиснення здійснюється, перш за все, тих компонентів звуку, що слабо сприймаються слухом людини. Мова йдеться про високі частоти або тихі звуки, що виникають одночасно або одразу після голосніших звуків. Такі компоненти звуку можуть зберігатися менш точно, або взагалі відкинуті.

Наприклад, компоненти звуку на частотах до 400 Гц і більше 14 Кгц можуть бути закодовані 4-ма бітами, а компоненти звуку від 400 Гц до 14 Кгц, які краще сприймає людина, — 16 бітами. У цьому випадку в середньому може бути потрібно 8 біт. Але це набагато краще, ніж кодувати увесь діапазон 16-ма бітами.

Основним параметром стиснення файлів цього типу є бітрейт, тобто ступінь стиснення файлу і якість його звучання. Розрізняють стиснення з постійним бітрейтом (CBR), змінним бітрейтом (VBR) і усередненим бітрейтом (ABR).

Одним із самих популярних форматів файлів цього типу є формат MP3 і WMA.

3. MIDI-формат. Незважаючи на те, що термін MIDI-формат звукових файлів розповсюджений в літературних джерелах, фактично він означає інтерфейс музичних інструментів. MIDI-програма забезпечує керування вбудованими синтезаторами й обмін даними між пристроями з цим інтерфейсом. MIDI-файли містять інформацію про довжину й гучність програвання нот, а також про інструменти, на яких необхідно виконувати ноти. Музичні інструменти оснащені спеціальними роз'ємами: MIDI IN, MIDI OUT, MIDI TRU. MIDI IN одного інструмента або звукової карти з'єднується з MIDI OUT іншого інструмента або мобільного телефону. Через роз'єм MIDI TRU здійснюється трансляція отриманих повідомлень.

Для відтворення MIDI-файлів використовуються два основних апаратних методи: метод частотної модуляції (FM-синтез) і метод таблиць хвиль (WT-синтез). Обидва методи реалізуються у звуковій карті. Назва методу синтезу співпадає з назвою синтезатора (FM-синтезатор, WT-синтезатор). Найчастіше для відтворення звуку використовуються програвачі Windows Media (рис. 10) і Winamp (рис. 11).

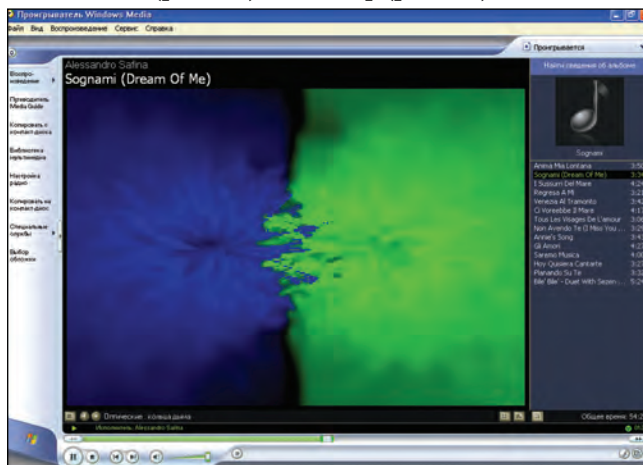


Рис. 10



Рис. 11

Література

1. Технология передачи данных. 7-е изд. /Г. Хелд. — СПб.: Питер, К.: Издательская группа ВНУ, 2003. — 720 с.: ил. — (Серия «Классика computer science»).
2. Гук М. Аппаратные средства IBM PC. Энциклопедия — СПб: Издательство «Питер», 2000. — 816 с.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

пр. Перемоги, 10, м. Київ, 01135, тел. (044) 481–32–21, факс (044) 236–1049
E-mail: ministry@mon.gov.ua, код ЄДРПОУ 38621185

Від 17.07.2013 №1/9-497

Міністерство освіти і науки, молоді та спорту Автономної Республіки Крим, департаменти (управління) освіти і науки обласних, Київської та Севастопольської міських державних адміністрацій, Інститути післядипломної педагогічної освіти.

Про використання Інструктивно-методичних матеріалів з питань створення безпечних умов для роботи у кабінетах інформатики та інформаційно-комунікаційних технологій загальноосвітніх навчальних закладів

Надсилаємо для використання у навчально-виховному процесі загальноосвітніх навчальних закладів Інструктивно-методичні матеріали «Безпечне проведення навчальних занять у кабінетах інформатики та інформаційно-комунікаційних технологій загальноосвітніх навчальних закладів».

Зазначені матеріали будуть розміщені на офіційних веб-сайтах Міністерства освіти і науки www.mon.gov.ua, Інституту інноваційних технологій і змісту освіти www.iitzo.gov.ua та надруковані у фахових виданнях.

Додаток: Інструктивно-методичні матеріали на 7 арк.

Заступник Міністра Б.М. Жебровський

Додаток до листа МОН від 17.07.2013 №1/9-497

Інструктивно-методичні матеріали

«Безпечне проведення навчальних занять у кабінетах інформатики та інформаційно-комунікаційних технологій загальноосвітніх навчальних закладів»

Відповідно до Закону України «Про освіту», Закону України «Про охорону праці» державна політика щодо охорони праці і безпеки життєдіяльності учасників навчально-виховного процесу загальноосвітнього навчального закладу (далі — навчального закладу) базується на принципі пріоритетності їхнього життя і здоров'я, повної відповідальності керівника навчального закладу за створення належних, безпечних і нешкідливих умов навчання і праці та запобігання нещасних випадків і професійних захворювань.

Запорукою збереження здоров'я і життя учасників навчально-виховного процесу під час роботи в кабінеті інформатики та інформаційно-комунікаційних технологій (далі — КПКТ) навчального закладу є дотримання вимог безпеки життєдіяльності (сукупність знань і навичок зі зміцнення та збереження здоров'я, життя, знання вимог пожежної, радіаційної, електробезпеки, попередження травматизму тощо).

1. Перелік нормативно-правових документів з питань охорони праці та безпеки життєдіяльності, що регулюють питання навчання в КПКТ

• Положення про організацію роботи з охорони праці учасників навчально-виховного процесу в установах і навчальних закладах, затверджене наказом Міністерства освіти і науки України від 01 серпня 2001 р. №563, зареєстроване в Міністерстві юстиції 20 листопада 2001 р. за №969/6160 (зі змінами) (далі — Положення про організацію роботи з охорони праці);

- Державні санітарні правила і норми влаштування, утримання загальноосвітніх навчальних закладів та організації навчально-виховного процесу ДСанПіН 5.5.2.008-01, затверджені постановою Головного державного санітарного лікаря України від 14 серпня 2001 р. №63 (далі — ДСанПіН 5.5.2.008-01);
- Державні санітарні правила та норми «Влаштування і обладнання кабінетів комп'ютерної техніки в навчальних закладах та режим праці учнів на персональних комп'ютерах ДСанПіН 5.5.6.009-98», затверджені постановою Головного державного санітарного лікаря України від 30 грудня 1998 р. №9 (далі — ДСанПіН 5.5.6.009-98);
- Положення про кабінет інформатики та інформаційно-комунікаційних технологій навчання загальноосвітніх навчальних закладів, затверджене наказом Міністерства освіти і науки України від 20 травня 2004 р. №407, зареєстроване в Міністерстві юстиції України 14 червня 2004 р. за №730/9329 (далі — Положення про кабінет інформатики);
- Правила безпеки під час навчання в кабінетах інформатики навчальних закладів системи загальної середньої освіти, затверджені наказом Держнаглядохоронпраці України від 16 березня 2004 р. №81, зареєстровані в Міністерстві юстиції України 17 травня 2004 р. за №620/9219 (далі — Правила безпеки під час навчання в кабінетах інформатики);
- Правила пожежної безпеки для закладів, установ і організацій системи освіти України, затверджені спільним наказом Міністерства освіти України

і Головного управління Державної пожежної охорони МВС України від 30 вересня 1998 р. №348/70, зареєстровані у Міністерстві юстиції України 17 грудня 1998 р. за №800/3240 (далі — Правила пожежної безпеки);

- Правила безпечної експлуатації електроустановок споживачів, затверджені наказом Комітету по нагляду за охороною праці України Міністерства праці та соціальної політики України від 09 січня 1998 р. №4, зареєстровані в Міністерстві юстиції України 10 лютого 1998 р. за №93/2533 (далі — Правила безпечної експлуатації електроустановок);
- Нормативи наповнюваності груп дошкільних навчальних закладів (ясел-садків) компенсуючого типу, класів спеціальних загальноосвітніх шкіл (шкіл-інтернатів), груп продовженого дня і виховних груп загальноосвітніх навчальних закладів усіх типів та Порядок поділу класів на групи при вивченні окремих предметів у загальноосвітніх навчальних закладах, затверджені наказом Міністерства освіти і науки України від 20 лютого 2002 р. №128, зареєстровані в Міністерстві юстиції України 06 березня 2002 р. за №229/6517;
- Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці, затверджене наказом Держнаглядохоронпраці України від 26 січня 2005 р. №15, зареєстроване в Міністерстві юстиції України 5 лютого 2005 р. за №231/10511 (НПАОП 0.00-4.12-05);
- Положення про порядок проведення навчання і перевірки знань з питань охорони праці в закладах, установах, організаціях, підприємствах, підпорядкованих Міністерству освіти і науки України, затверджене наказом Держнаглядохоронпраці України від 18 квітня 2006 р. №304, зареєстроване в Міністерстві юстиції України 07 липня 2006 р. за №806/12680 (далі — Положення про порядок проведення навчання з охорони праці);
- Положення про порядок розслідування нещасних випадків, що сталися під час навчально-виховного процесу в навчальних закладах, затверджене наказом Міністерства освіти і науки України від 31 серпня 2001 року №616, зареєстроване в Міністерстві юстиції України 28 грудня 2001 року за №1093/6284 (далі — Положення про порядок розслідування нещасних випадків).

2. Основні вимоги безпеки в кабінеті інформатики та інформаційно-комунікаційних технологій навчання

Основною метою створення КПКТ є забезпечення належних умов для проведення навчально-виховного процесу з розв'язання завдань, визначених цілями та змістом освітньої галузі «Технології» Державного стандарту базової і повної середньої освіти, затвердженого постановою Кабінету Міністрів України від 23 листопада 2011 р. №1392 та Державного стандарту початкової загальної освіти, затвердженого постановою Кабінету Міністрів України від 20 квітня 2011 р. №462.

КПКТ повинен розміщуватися в окремому приміщенні і мати допоміжне приміщення (лаборантську).

Приміщення КПКТ має відповідати вимогам:

- ДСанПіН 5.5.2.008-01;
- Правил безпечної експлуатації електроустановок;
- Правил пожежної безпеки.

Облаштування, обладнання, реконструкція КПКТ здійснюється відповідно до вимог ДСанПіН 5.5.2.008-01 та ДСанПіН 5.5.6.009-98.

Обладнання, яке створює навчальне середовище і використовується для проведення навчально-виховного процесу в КПКТ, має відповідати вимогам безпеки життєдіяльності.

Електротехнічне обладнання КПКТ повинно відповідати вимогам електробезпеки, передбаченим Правилами безпечної експлуатації, Правилами безпеки під час навчання в кабінетах інформатики, іншими нормативними документами.

У КПКТ дозволяється використовувати кіно-, відео- та мультимедійну апаратуру з урахуванням вимог Правил безпечної експлуатації електроустановок.

Сучасні технічні засоби навчання, до яких належить комп'ютер, мультимедійний проектор, проекційний екран, інтерактивна дошка тощо, мають бути заземленими за схемами, які розміщені в технічних описах та інструкціях, що додаються до пристроїв.

Технічне обслуговування та ремонт комп'ютерної техніки КПКТ проводять підприємства, які здійснюють сервісне обслуговування і мають право на проведення таких робіт відповідно до угод, які укладають навчальні заклади.

Основна система електроживлення повинна використовуватися тільки для підключення навчального комп'ютерного комплексу.

Відповідно до Правил пожежної безпеки КПКТ обладнується системою автоматичної пожежної сигналізації, переносними вогнегасниками та засобами аварійного відключення живлення.

Комплектність, типи та розташування шкільних меблів у КПКТ та лаборантській відповідає вимогам санітарно-гігієнічних правил і норм.

Засоби навчання, якими обладнано КПКТ, можуть використовуватися для навчання інших навчальних предметів у позаурочній роботі. Позакласні заняття з інформатики проводяться тільки в присутності вчителя.

Тривалість та періодичність навчальних занять, чисельність навчальних груп у КПКТ визначаються ДСанПіН 5.5.6.009-98.

Готовність КПКТ до навчального процесу перед початком навчального року перевіряє комісія, створена відповідно до Положення про організацію роботи з охорони праці.

Завідувач КПКТ розробляє паспорт кабінету, у якому зазначаються основні його параметри: освітлення, площа, наявність інженерних мереж (водопостачання, каналізація, вентиляція, тепломережа, електромережа), забезпечення меблями, обладнанням, підручниками, навчальними посібниками, приладам, науково-методичними і науково-популярними журналами тощо.

У кожному КПКТ мають бути інструкції до проведення практичних робіт, інструкції до програмних засобів та систем програмування тощо; інструкції з безпеки і журнал реєстрації інструктажів з безпеки життєдіяльності.

З метою попередження та запобігання нещасних випадків під час роботи в КПКТ необхідно дотримуватися вимог безпеки, що визначаються Правилами безпеки під час навчання в кабінетах інформатики.

КПКТ має бути обладнаний аптечкою з набором медикаментів, перев'язувальних засобів і приладь та ін-

формацією про місце знаходження і номер телефону найближчого лікувально-профілактичного закладу, де можуть надати кваліфіковану медичну допомогу.

У разі нещасного випадку з учнем під час роботи в КПКТ учитель (лаборант) повинен терміново повідомити про це керівника і службу охорони праці навчального закладу, організувати надання першої домедичної допомоги потерпілому, викликати медпрацівника навчального закладу, за потреби організувати виклик швидкої допомоги. Робота з розслідування нещасного випадку проводиться відповідно до Положення про порядок розслідування нещасних випадків.

3. Відповідальність і обов'язки завідувача КПКТ, учителя інформатики, лаборанта щодо забезпечення безпеки життєдіяльності учнів

Відповідно до Положення про організацію роботи з охорони праці відповідальність за створення належних, безпечних і здорових умов навчання і праці в КПКТ несе керівник навчального закладу.

Відповідальність за безпечний стан робочих місць, обладнання, приладів, інвентарю несуть завідувач КПКТ, учителі інформатики.

Відповідальність за стан електрообладнання, вентиляції, водопровідної та каналізаційної систем і сантехнічних споруд у КПКТ несуть особи, призначені наказом керівника навчального закладу.

Учитель інформатики контролює роботу лаборанта і надає йому практичну допомогу з метою попередження нещасних випадків.

Відповідно до Положення про організацію роботи з охорони праці завідувач КПКТ, учителі інформатики проходять один раз на три роки навчання з безпеки життєдіяльності з наступною перевіркою знань.

До обов'язків завідувача КПКТ з питань безпеки життєдіяльності входить:

- забезпечення безпечних умов для проведення уроків і позаурочної роботи;
- проведення інструктажів з безпеки життєдіяльності;
- контроль за дотриманням учнями в КПКТ правил поведінки, чистоти, порядку, правил безпечної експлуатації електротехнічного і іншого обладнання, дотриманням правил пожежної безпеки;

Робоче місце вчителя обладнується системою управління електроживленням навчального комп'ютерного комплексу, яка забезпечує його включення (відключення).

Лаборант КПКТ, який працює під керівництвом завідувача, учителя інформатики, допомагає їм в орга-

нізації і проведенні навчальних занять, демонстрацій, позаурочної роботи з навчального предмета.

4. Проведення інструктажів з питань безпеки життєдіяльності

Навчання з питань безпеки життєдіяльності в умовах роботи в КПКТ навчальних закладів проводиться за допомогою інструктажів відповідно до Положення про порядок проведення навчання з питань охорони праці, яким визначається порядок проведення, тематика та організація проведення інструктажів з безпеки життєдіяльності учнів.

Інструктажі з безпеки життєдіяльності з учнями проводить завідувач КПКТ (учитель інформатики).

Перед початком роботи учнів у КПКТ завідувач кабінету проводить первинний інструктаж з безпеки життєдіяльності, який знайомить їх з правилами поведінки в кабінеті. Запис про проведення первинного інструктажу робиться в спеціальному журналі реєстрації інструктажів з безпеки життєдіяльності, який зберігається в кабінеті (згідно з додатком).

Первинний інструктаж перед кожним практичним заняттям у КПКТ проводиться учителем інформатики за інструкціями з безпеки під час навчання відповідно до виконуваних робіт і реєструється в класному журналі на сторінці предмета у розділі про зміст уроку: «Інструктаж з БЖД». Учні, які інструктуються, не розписуються про такий інструктаж.

При порушенні учнями правил, норм, вимог нормативно-правових актів з питань безпеки життєдіяльності, що можуть призвести або призвели до травмування, аварії, пожежі та інших надзвичайних ситуацій, проводиться позаплановий інструктаж з безпеки життєдіяльності. Позаплановий інструктаж також проводиться на робочому місці при введенні в дію нових нормативно-правових актів з питань безпеки життєдіяльності, зміні умов виконання навчальних завдань, у разі скоєння нещасних випадків тощо.

Запис про проведення позапланового інструктажу з безпеки життєдіяльності здійснюється в спеціальному журналі реєстрації інструктажів (додаток).

Цільовий інструктаж з безпеки життєдіяльності з учнями проводиться у разі організації позанавчальних, позашкільних заходів (олімпіад, турнірів з предмета тощо), при ліквідації аварії, надзвичайної ситуації тощо.

Реєстрація проведення цільового інструктажу з безпеки життєдіяльності здійснюється у спеціальному журналі реєстрації інструктажів (додаток).

Додаток

назва загальноосвітнього навчального закладу

Розпочато: _____ 20__ р.
Закінчено: _____ 20__ р.

Журнал

реєстрації первинного, позапланового, цільового інструктажів з безпеки життєдіяльності учнів у кабінеті інформатики та інформаційно-комунікаційних технологій

№ з/п	Прізвище, ім'я та по батькові особи, яку інструктують	Дата проведення інструктажу	Клас	Назва інструктажу, назва інструкції	Прізвище, ім'я та по батькові особи, яка проводила інструктаж	Підпис особи, яка проводила інструктаж	Підписи особи, яку інструктували
1	2	3	4	5	6	7	8

Учні розписуються у журналі інструктажу, починаючи з 9 класу.



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

пр. Перемоги, 10, м. Київ, 01135, тел. (044) 481-32-21, факс (044) 236-1049
E-mail: ministry@mon.gov.ua, код ЄДРПОУ 38621185

Від 17.07.2013 № 1/11-11636
На № _____ від _____

Директору Інституту педагогіки
НАПН України Топузову О.М.

Про надання грифа

Міністерство освіти і науки України розглянуло матеріали комісії з математики Науково-методичної ради з питань освіти Міністерства освіти і науки України на навчальну програму «Інформатика. 5–9 класи. Для навчальних закладів з поглибленим вивченням предметів природничо-математичного циклу» (за ред. А.М. Гуржія і В.Ю. Бикова) і схвалює її до використання з грифом «Рекомендовано Міністерством освіти і науки України».

Заступник Міністра Б. М. Жебровський

Примітка: програма надрукована в журналі «Комп'ютер у школі та сім'ї» №6 за 2012 рік.



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

пр. Перемоги, 10, м. Київ, 01135, тел. (044) 481-32-21, факс (044) 236-1049
E-mail: ministry@mon.gov.ua, код ЄДРПОУ 38621185

Від 17.07.2013 № 1/11-11637
На № _____ від _____

Директору Інституту педагогіки
НАПН України Топузову О.М.

Про надання грифа

Міністерство освіти і науки України розглянуло матеріали комісії з математики Науково-методичної ради з питань освіти Міністерства освіти і науки України на підручник «Інформатика. 5 клас. Для навчальних закладів з поглибленим вивченням природничо-математичних дисциплін» (за ред. А.М. Гуржія) і схвалює його до видання з грифом «Рекомендовано Міністерством освіти і науки України».

Оригінал-макет підручника повинен пройти погодження і отримати дозвіл на друк в установленому порядку.

Контрольні примірники просимо надсилати за адресою: 03035, м. Київ, вул. Митрополита Василя Липківського, буд. 36, Інститут інноваційних технологій і змісту освіти.

Заступник Міністра Б.М. Жебровський

Успіхи загальноосвітніх навчальних закладів України у Всеукраїнських та Міжнародних олімпіадах з інформатики

Редакція науково-методичного журналу «Комп'ютер у школі та сім'ї» щорічно визначає 12 загальноосвітніх навчальних закладів, учні яких досягли найкращих результатів у Всеукраїнських і Міжнародних олімпіадах з інформатики. За диплом першого ступеню у Всеукраїнських олімпіадах нараховується 7 балів, за диплом другого ступеня — 5 балів і за диплом третього ступеня — 3 бали. За отриману золоту медаль на Міжнародних олімпіадах нараховується 11

балів, за срібну медаль — 9 балів і за бронзову медаль — 7 балів. Результати учнів Українського фізико-математичного ліцею Київського національного університету імені Тараса Шевченка не враховуються, тому що вони виступають командою на рівні області. У цьому році вперше найкращих результатів досягли учні Полтавського обласного ліцею-інтернату для обдарованих дітей із сільської місцевості при Кременчуцькому педагогічному училищі імені А. С. Макаренка.

Навчальні заклади	Дипломи Всеукраїнської олімпіади			Медалі Міжнародної олімпіади			Кільк. балів	Місце
	I ст. (7 б)	II ст. (5 б)	III ст. (3 б)	Зол. (11 б)	Сріб. (9 б)	Брон. (7 б)		
Полтавський обласний ліцей-інтернат для обдарованих дітей із сільської місцевості при Кременчуцькому педагогічному училищі	2	3	4		1	1	57	1
Харківський НВК №45 «Академічна гімназія» Харківської міської ради	3	3					36	2
Комунальний заклад освіти «Дніпропетровський ліцей інформаційних технологій при Дніпропетровському національному університеті»	1	2	1			1	27	3
Український фізико-математичний ліцей Київського національного університету імені Тараса Шевченка		3	3				24	-
Комунальний заклад освіти «Харківський фізико-математичний ліцей №27» Харківської міської ради	1		1			1	17	4
Запорізький НВК №7 Запорізької міської ради			5				15	5
Загальноосвітній спеціалізований санаторно-інтернатний заклад II–III ст. «Ерудит» для обдарованих дітей Донецької області		2					10	6
Фізико-математична гімназія №17 Вінницької міської ради			3				9	7
Львівський фізико-математичний ліцей-інтернат при Львівському національному університеті імені Івана Франка		1	1				8	8-12
Миколаївський муніципальний колегіум Миколаївської міської ради		1	1				8	8-12
Сумська спеціалізована школа №17 I–III ст. м. Суми		1	1				8	8-12
Старокостянтинівська гімназія Старокостянтинівської міської ради Хмельницької області		1	1				8	8-12
Ліцей «Наукова зміна» м. Києва		1	1				8	8-12



На четвертій сторінці обкладинки: **Фурко Роман Володимирович** — учень 11-го класу Полтавського обласного ліцею-інтернату для обдарованих дітей із сільської місцевості при Кременчуцькому педагогічному училищі ім. А. С. Макаренка, нагороджений срібною медаллю XXV Міжнародної олімпіади з інформатики

Підписано до друку 31.07.2013 р. Формат 60x84 1/8. Папір офсет. Друк офсет. Умовн. друк. арк. 5,88.

Умовн. фарбо-відб. 11,76. Обл.-вид. арк. 8,54. Видавець: ФО-П Жугастрова О.В. Зам. №13–221.

Віддруковано у друкарні видавництва «Фенікс». Свід. ДК 271 від 7.12.2000 р.

Адреса видавця: вул. Половецька, 12/42, к. 88, м. Київ, 04107, Україна.

E-mail: csf221@rambler.ru, www.csf.vashpartner.com.

Повне або часткове передрукування матеріалів журналу можливе тільки з письмового дозволу редакції.

Передплату на наш журнал можна оформити у будь-якому відділенні зв'язку. Наш індекс 74248

**XXVI ВСЕУКРАЇНЬСЬКА ОЛІМПІАДА З ІНФОРМАТИКИ
(17–21 березня 2013 р., м. Луганськ)**



Индекс 74248

