

- спрямованість на актуалізацію знань і умінь;
- забезпечення можливості самостійної евристичної діяльності.

Кожну тему курсу фізики доцільно забезпечити системою задач різних типів, яка повинна відповідати таким вимогам:

- складатись із задач, розрахованих на різні рівні вимог щодо навчальних досягнень учнів;
- враховувати всі основні закономірності, які вивчаються в даній темі;
- представляти собою сукупність задач із зростаючою складністю, причому складність задачі повинна визначатись кількістю пізнавальних кроків, які необхідно здійснити для її розв'язання;
- визначати типологію методів пізнання, які використовуються в фізиці;
- забезпечувати можливість творчо-пошукової діяльності учнів, яка передбачає: самостійне застосування раніше набутих знань і умінь в нових умовах; бачення проблеми у заданій ситуації; визначення нових функцій об'єкта дослідження та усвідомлення його структури; пошук альтернативних шляхів розв'язання задачі.

Застосування технології евристичного навчання при розв'язуванні задач з фізики забезпечує формування в учнів наступних умінь і навичок:

- проектування власних дій в умовах, коли інформація є обмеженою;
- узагальнення і абстрагування;
- цілісного осмислення інформації;
- розумового експерименту як засобу пізнання;
- самостійного конструювання навчальних завдань;
- регуляції і стимулювання продуктивності пізнавальної діяльності;
- здійснення самоконтролю і самоаналізу.

Для успішної організації навчальної діяльності за технологією евристичного навчання необхідно виконання таких умов:

- усвідомлення функцій педагогічного керування в організації навчання;

- застосування способів регуляції навчальної діяльності;
- якісної підготовки навчальних творчо-пошукових задач з урахуванням рівня їх складності;
- використання інформаційного синтезу;
- побудова вірогідних гіпотез і спостереження за їх розвитком у ході пошукової діяльності;
- створення нових ситуацій інноваційного середовища;
- забезпечення особистісної значущості пізнавальної діяльності та її результатів для учнів.

Досвід показує, що застосування технології евристичного навчання фізики дає значний педагогічний ефект, що підтверджується підвищенням рівня знань учнів з фізики. Підсумком евристичного навчання фізики, яке спирається на особистісний підхід до учнів, є розвиток творчого ставлення до навчальної діяльності, виховання в учнів аналітико-синтезованого мислення, самостійності при розв'язанні навчальних проблем, усвідомлення власних можливостей, що є необхідною умовою здійснення особистісно-орієнтованого навчання учнів.

Список використаних джерел:

1. Бех І. Особистісно-орієнтований підхід у вихованні // Професійна освіта: педагогіка і психологія: Українсько-польський щорічник. — Ченстохова-Київ, 2000. — С.331-350.
2. Бондаревская Е.В. Гуманистическая парадигма личностно ориентированного образования // Педагогика. — 1997. — № 4. — С.11-17.
3. Благодаренко Л.Ю., Грищенко Г.П., Шут М.І. Педагогічні класи фізико-математичного профілю як форма поглибленого вивчення фізики // Тези доповідей Всеукраїнської науково-практичної конференції "Стратегічні проблеми формування змісту курсів фізики та астрономії в системі загальної середньої освіти". — Львів, 2002. — С.64-65.
4. Друзь З.В. Система нестандартних завдань як засіб активізації навчальної діяльності школярів // Матеріали докладів Всеукраїнської науково-практичної конференції «Активізація учебной деятельности школьников». — Кривой Рог, КрПИ, 1995. — С.119-121.

Отримано: 19.05.2004.

УДК 681.3

А.В.Гаєвська, В.С.Щирба

Кам'янець-Подільський державний університет

ФОРМУВАННЯ СТРУКТУРИ ЛАБОРАТОРНОГО КУРСУ З ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ VISUAL BASIC

В роботі аналізується досвід впровадження технології візуального програмування в навчальний процес. Разом з набором індивідуальних завдань встановлюється доцільність їх розв'язання для швидкого освоєння особливостей програмування в середовищі Visual Basic.

In activity the experience of an intrusion of technologies of visual programming in educational process is analysed. Together with a set of personal problems the expediency of their solution for fast development of features of programming in medium Visual Basic is established.

Визначальна відмінність інформатики від інших дисциплін, що вивчаються у вищій школі, полягає в динамізмі змін предмету вивчення [2]. Не становлять виключення і питання пов'язані з розробкою прикладного програмного забезпечення.

На наш погляд, епоха повального захоплення комп'ютером як універсальним персональним "комбайном" різного роду редакторів повертає в бік комп'ютерного моделювання, а це, в свою чергу, вимагає розгляду методичних питань пов'язаних з вивченням програмування. Звичайно, мова не йде про примітивні

програми, на зразок розв'язання квадратного рівняння. Актуальними сьогодні є програми дослідження візуальних, в тому числі динамічних об'єктів.

Програми для ЕОМ можуть бути різними за складністю. Існує багато програм, які задумані, розроблені і використовуються однією і тією ж людиною. Як правило, це програми, що використовуються один єдиний раз для розв'язання тимчасово назрілої проблеми. Значну частку становлять програмні продукти допоміжного та довідкового характеру.

Не можна відразу створювати програми складної структури. Потрібно побудувати послідовність відносно не складних програм прикладного характеру, написання яких забезпечить поетапне освоєння середовища візуального програмування, а прикладний характер задач гарантуватиме зацікавленість їх розв'язанням. Тут під прикладним характером ми розуміємо призначення програми розв'язувати різне коло задач користувачів персональних комп'ютерів.

Одним із перспективних напрямків розвитку технологій програмування вважають використання об'єктно-орієнтованого програмування. Це системи нового покоління, основним поняттям яких виступає візуально заданий об'єкт. Тобто в системах об'єктно-орієнтованого програмування широко використовується графічний інтерфейс.

Розробка будь-яких програм-додатків системи Windows на базі ідеології візуального програмування дозволяє використовувати різноманітні набори об'єктів для подання графічної інформації. Доступність і гнучкість різноманітних систем введення, обробки і подачі графічної інформації породило в багатьох програмістів захоплення розробкою моделей (найчастіше динамічних), що спостерігаються візуально.

Особливо хочеться відзначити роботи з моделювання об'ємних зображень. Аналізуючи запропоновані способи одержання візуальних моделей можна виділити такі підходи: одержання на плоскому екрані особливих проєкцій тривимірних об'єктів, таким чином, щоб у спостерігача виникло відчуття спостереження тривимірного простору; одержання псевдооб'ємних моделей (у процесі побудови таких моделей використовуються особливості людського зору).

Будь-яку прикладну програму розробляють для якомога ширшого кола користувачів і, отже, вона повинна задовольняти загальним вимогам:

- *функціональність програм*, тобто повнота задоволення ними широкого кола потреб користувача;
- *наочний, зручний, інтуїтивно зрозумілий і звичний інтерфейс*, як спосіб взаємодії програми з користувачем;
- *надійність програми*, тобто стійкість її до помилок користувача, відмови обладнання і т. д., та "розумні" її дії в цих ситуаціях;
- *простота освоєння програми* навіть користувачами-початківцями. Для цього використовуються інформативні підказки, вбудовані довідники і детальна документація.

Таким чином, перш ніж приступати до освоєння програмування, потрібно загострити увагу на формуванні культури програміста, виробленні правильного (прийняттого для інших користувачів) стилю та почерку написання програми.

Проблема встановлення вимог до комп'ютерних програм існує з часу впровадження обчислювальної техніки. Однак, ще й досі проблема науково-методичного обґрунтування вимог до розроблення та використання комп'ютерних програм повністю не розв'язана.

І все-таки, виходячи з досвіду роботи, принципів психології та естетики, враховуючи кращі зразки наочних посібників та комп'ютерних програм, можна сформулювати цілий ряд порад тим програмістам, хто планує писати прикладні програми.

Завдання лабораторного курсу з програмування в середовищі Visual Basic [1; 4] — навчити будувати невеличкі програми-додатки з додержанням цих вимог за допомогою сучасних технологій візуального програмування з метою освоєння основних прийомів роботи в цьому середовищі.

Вивчення курсу ми розпочинаємо з лекційних занять, де студенти чи не вперше знайомляться з класичною (традиційною) теорією кольорів. Окрім кольорового круга та принципів його формування їм пропонується виклад основних формул гармонії кольорів: ко-

льорова схема на базі сусідніх кольорів, на базі протилежних кольорів, на базі природних поєднань кольорів, на базі кольорів різного ступеня насиченості.

Для підсилення тезису про значення правильного поєднання кольорів пропонується провести психологічний тест. Студенти порівнюють як сприймає людина один і той самий червоний квадрат на різних фонах: чорному, білому, жовтому, зеленому, синьому та фіолетовому.

На чорному фоні він напружено світиться (навіюється страх перед уявою темряви та пожежі чи крові). На білому фоні він виглядає чітко і точно (акцент до світлого фону). На жовтому він виглядає більш-менш привабливо (акцент подібності споріднених кольорів). На зеленому, синьому він виглядає дуже яскраво (акцент протистояння, дисонанс). А на фіолетовому він практично губиться (відчувається бажання відкинути це зображення).

Інший (наступний) тест ілюструє взаємодію різних кольорів. На малюнку можна побачити відразу дві подібні взаємодії:

1. Сіро-блакитний колір набуває трохи зеленуватого відтінку на червоній частинці картинки.
2. Смужка сіро-блакитного здається більш вузькою на темно-червоній частині і більш широкою на світлій частині.

Для ілюстрації цього матеріалу використовується комп'ютерна програма "Колір", розроблена нами в стилі строгого наукового викладу, який сприяє зосередженості студентів.

Для навіювання творчих роздумів відразу ж після цього пропонується проглянути програму "Графічний редактор Paint", яка розроблена в стилі приємних емоцій і призначена для роботи молодших школярів. Позитивом даної програми є і те, що в студентів знімається втома та напруга після перегляду попередньої програми. Слід відзначити, що практично ніхто з них не звертає увагу на зміст матеріалу, а обговорення ведеться виключно по дизайну програми.

Отримавши заряд приємних емоцій, студенти з задоволенням обговорюють недоліки наступної програми "Комп'ютерні віруси", відзначаючи при цьому "бурхливу" фантазію автора-програміста. На хвилі цього піднесення пропонується зафіксувати в конспекті найбільш розповсюджені помилки не професійних дизайнерів.

Очевидно немає потреби викладати їх в рамках цієї публікації, оскільки ми обговорюємо лише методичку подачі матеріалу, а не його зміст.

Першу програму студентам пропонується створити не в середовищі Visual Basic, а редактором презентацій Power Point. Інтерфейс редактора набагато простіший, а арсенал методів роботи — більш потужний, ніж в середовищі Visual Basic. Вони мають більше можливостей проявити політ фантазії, творчу особистість, зосередженість над дотриманням вимог комп'ютерного дизайну. Завдання викладача — постійно наголошувати на цьому, не йти на поступки, пом'якшення вимог. Важливим є і правильний підбір змісту інформації, яку повинна відображати програма, котру розробляє студент. Від цього залежить емоційний заряд студента, бажання довести написання програми до її завершення.

Психологічно важко сприймається подальший перехід з Power Point в середовище Visual Basic, оскільки процес оформлення дизайну тут набагато складніший і вимагає більших зусиль та витрат часу. Не потрібно критикувати за це Visual Basic. Він мудро дозволяє використовувати і поєднувати в одному проєкті розробки інших середовищ. Тому немає потреби повторювати їх інструментарій та можливості.

Досвід роботи показує, що перші кроки варто розпочати з простих операцій нанесення об'єктів на форму, встановлення їх властивостей, а кодову частину доцільно обмежити простими командами завершення роботи та переходу на іншу форму. Тут же потрібно проілюструвати суть ідеології системи Windows, яка базується на подіях.

На наш погляд, найкраще підходить для досягнення цієї мети задача розробки титульної сторінки програми за зразком титульної сторінки курсової роботи. Крім стандартних написів обов'язково потрібно помістити хоча б дві командні клавіші, одна з яких відповідатиме за завершення роботи, а інша — за продовження роботи (перехід на іншу форму). Раніше ми пропонували обов'язково встановлювати також клавішу виклику інструкції, але в доречності строгого виконання цієї вимоги так і не переконалися. Натомість, більшого ефекту досягаємо від використання об'єктів Share та Image з метою оформлення робочого вікна малюнками чи фотографіями, оскільки є можливість попрацювати над правильним поєднанням кольорів.

Змінюючи розміри форми і забезпечивши її розміщення на весь екран, студенти відчувають потребу у виклику події, яка забезпечить завершення роботи чи перехід на іншу форму. Тут вони зрозуміють різницю між різними подіями і однотипними подіями для різних об'єктів та навчаться писати фрагменти коду. Вперше студенти відчувають призначення вікна проекту. Наступний крок — приєднання нових форм до проекту та включення їх в активну роботу.

Після цього логічно напрошується поглиблення знань студентів в напрямку написання коду. Ми обрали для цього задачу розробки меню, яке запускати те чи іншу прикладну програму. Використання нових графічних об'єктів тут не передбачається, але виникає потреба в створенні масиву об'єктів, зокрема масиву надписів. Кодова частина збагачується обробкою події наведення курсору мишки на об'єкт, а це, в свою чергу, вимагає оголошення локальної дії для ряду змінних в межах коду однієї події. Використання масиву об'єктів зумовлює оголошення змінної локального характеру для обробки індексу, але в межах форми. Також в цьому випадку практично не обійтися без використання розгалуження типу перемикача, тобто оператора Select Case.

Розв'язання двох попередніх задач дає мінімальний рівень навичок програмування в середовищі Visual Basic. Можна вважати, що з основними особливостями програмування в ньому завдяки переднім задачам студенти уже ознайомилися. Тепер потрібно поглибити їх знання за рахунок розв'язання задач прикладного характеру. Слід відзначити, що ми свідомо уникали на попередньому етапі поняття про методи в середовищі Visual Basic (мова йшла лише про властивості об'єктів, події та функції і оператори). Розробляючи завдання для наступних лабораторних занять, ми ставили перед собою три задачі:

- включити в роботу практично весь арсенал стандартних об'єктів Visual Basic;
- не ускладнювати без потреб кодову частину програми;
- стимулювати роботу студентів шляхом реалізації так званої позитивної емоційності, яка має велике значення для прояву зацікавленості програмою.

Розглядалося декілька варіантів продовження послідовності завдань з врахуванням цих задач. На даному етапі найбільш актуальною виявилася саме третя задача — посилення позитивної емоційності. Студентам запропонували розробити власний калькулятор.

Зауважимо відразу, що не потрібно ускладнювати задачу. Достатньо обмежитися чотирма арифметичними діями і обчисленням квадратного кореня (останнє викликане лише бажанням відмітити різницю між синтаксисом для даної функції в Бейсіку і Паскалі). Серед недоречностей і типових (досить частих) помилок відмітимо лише логічну помилку, яка полягає в діленні на нуль та обчисленні квадратного кореня з від'ємного числа.

Задачу розробки комп'ютерної моделі калькулятора можна модифікувати в залежності від рівня під-

готовки групи. Наприклад, можна пропонувати обробляти вираз, що містить дужки, збільшити кількість функцій тощо. Разом з тим, не бажано цим захоплюватися, бо ускладнення призведе лише до збільшення коду і втрати зацікавленості програмою. Краще більше попрацювати над дизайном моделі.

Наступна задача, яка розширить уявлення про інструменти та ідеологію Visual Basic, може полягати в розробці власної записної книжки (умовно назвемо її "адресна книга" або "телефонний довідник").

По-перше, ми зацікавлюємо студентів встановленням взаємозв'язку між проектом і базою даних, що відкриває перспективу використання даної програми для написання цілого ряду цікавих прикладних програм.

По-друге, на прикладі оновлення даних чи поповнення записів знайомимо їх з такою важливою категорією як методи.

По-третє, ілюструється приклад використання цілого ряду нових інструментів: текстове поле, списки, зв'язний елемент. Встановлюючи шлях до файлу бази даних, демонструємо можливість розширення панелі інструментів на прикладі діалогового вікна відкриття файлу.

Проте, зауважимо, що дана задача не викликала особливого захоплення серед студентів і потрібно ще попрацювати над модифікацією її постановки. Причина, на нашу думку, полягає в необхідності розв'язання допоміжної задачі формування бази даних.

Значне піднесення емоційності викликали задачі побудови об'ємних та динамічних зображень [3] з використанням API-функцій графічної бібліотеки (додаткові можливості Visual Basic). Фрагмент демонстраційної програми, яка дозволяє будувати відрізки, прямокутники та овали зафарбовуванням замкнених областей модифікується на побудову об'ємного зображення окремих літер. Раніше пропонувалося змодельовати динаміку повертання зображення куба навколо довільної осі. Причиною неприйняття такої задачі стало слабе володіння математичним апаратом. Розв'язати її змогли не всі студенти. Тому від такої постановки задачі довелося відмовитися. Непогане враження на студентів справила задача динамічного повороту тексту у вигляді пропелера.

На завершення студентам пропонується змодельовати будь-яку динамічну ігрову програму з арсеналу, наприклад, мобільних телефонів (тетріс, змійовик тощо). Хотілося б відзначити не лише величезне емоційне піднесення і захоплення студентів, завдяки їм при написанні подібних програм, але й проведений великий об'єм самостійної роботи по поглибленню знань з програмування в середовищі Visual Basic при реалізації таких задач.

Таким чином, логічно виправдана послідовність задач прикладного характеру дозволяє за лічені години практичної роботи досить глибоко вникнути в особливості візуального програмування і зацікавити студентів потужним і разом з тим досить простим середовищем Visual Basic.

Список використаних джерел:

1. Браун С. Visual Basic 6: Учебный курс / Пер. с англ. — СПб.: ЗАО "Издательство "Питер", 1999. — 576 с.
2. Есипов А.С. Информатика и информационные технологии для учащихся школ и колледжей. — СПб.: БХВ-Петербург, 2004. — 480 с.: ил.
3. Рейнбоу В. Компьютерная графика. Энциклопедия. — СПб.: Питер, 2003. — 768 с.: ил.
4. Visual Basic 6.0: Пер. с англ. — СПб.: БХВ, 2000. — 992 с.: ил.

Отримано: 22.06.2004.