

легально працюють магазини, які торгують піратською продукцією, а державні установи, в тому числі навчальні заклади, що продукцію купують. Чи потрібно при цьому дивуватись, що найпопулярнішими є ресурси Інтернету, звідки можна "стягнути" реферати та курсові роботи і видати їх за власні?

Офіційно закупити все програмне забезпечення, яке зараз використовується, нереально, хоч різного роду дилери намагаються переконати нас і отримати свої відсотки. Та і не потрібно нічого купувати, бо крім світу комерційних програм є ще величезна кількість наукових організацій, які виготовляють та вільно поширюють програмне забезпечення, яке нічим не поступається комерційному. Фінансування таких організацій здійснюється за класичною схемою: державні установи та промислові компанії надають кошти на розробки, результати яких публікуються і стають доступні всім. Так розроблені операційна система Linux, яка нічим не поступається Windows, та тисячі навчальних, прикладних та ігрових програм для неї. Так зроблений Офіс602 та сотні інших програм для Windows. Можна брати, користуватися, отримувати підтримку авторів. Але ми не беремо, хоч для шкільних потреб ці програми нічим не гірші від комерційних. Всі підручники і навчальні плани ми будемо на базі найновіших комерційних розробок, хоч на практиці діти не в стані зрозуміти десятої частки можливостей цих програм, а викладений в посібниках матеріал стосується (за винятком назви) будь-якої програми цього класу.

Окремо слід сказати про навчальні програми. Це звучить дуже прикро, але україномовних навчальних програм практично нема, бо держава за них нічого не платить, а взяти нема де. Мрії про використання комп'ютерів на уроках так і залишилися мріями. Якщо на старій техніці ентузіасти ще розробляли простенькі навчальні програми, то зараз розраховувати на те, що наші вчителі та викладачі вузів без оплати самі розроблять навчальні програми, так само марно, як сподіватися, що селяни самі виготовлятимуть собі трактори. Програмування — це важка кваліфікована праця, яка має бути належно оплачена. Коли б була можливість оплатити авторам хоч 5-10 гривень за навчальний заклад, який використовує програму (що в 100 разів менше від розвинених країн), то за кілька років ми б мали сотні розробок. Але зараз навіть програми, які автори пропонують безкоштовно, ніхто не розглядає, бо цими питаннями не займається жоден з чисельних підрозділів Міністерства освіти і науки України. Автори можуть лише в обхід Міністерства розмістити розробки в Інтернет, що я, наприклад, зробив на сторінці [www.lviv.ua/algo/](http://www.lviv.ua/algo/).

Сподіваюся, що учасники конференції погодяться з такими пропозиціями.

1. Вважати використання піратського програмного забезпечення в державних навчальних закладах неприпустимим. При розгляді та затвердженні навчальних планів, авторських програм, підручників та посібників, які базуються на комерційному програмному забезпеченні, вказувати ціни програм, джерела їх придбання та фінансування.

2. Сформувані та популяризувати засобами Інтернет базу даних про програмні засоби, які можуть бути використані навчальними закладами. В кожному конкретному випадку винесення рішення про купівлю програмного забезпечення за кошти державного чи місцевого бюджету проводити порівняльний аналіз пропонованих комерційних програм з безкоштовно

поширюваними аналогами.

3. Розробити систему фінансування розробників навчальних програм та перекладу безкоштовно поширюваних програм українською мовою. Вважати успішно завершеними лише ті проекти цієї галузі, результати яких доступні для використання по мережі Інтернет. Подальше фінансування проектів пов'язувати з реальним використанням програмних продуктів в навчальному процесі.

## ОСНОВНІ ВИМОГИ, ПОДАНІ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

*Прийма С.М.*

*Мелітопольський державний педагогічний університет*

Незважаючи на відносну доступність якісного програмного забезпечення, яке в цілому відповідає єдиним дидактичним вимогам вивчення курсу інформатики, у вчителя дуже часто виникає необхідність використання своїх знань та вмінь з програмування для розв'язання деяких нестандартних прикладних задач, тобто без звернення до послуг професійних програмістів. Серед таких завдань можна, наприклад, вирізнити різноманітні засоби більш ефективного використання обчислювальної техніки в сфері адміністративного керування школою. Але найчастіше, особливо в умовах реформування шкільної освіти, вдосконалення її змісту та методів, завдання полягає в створенні або внесенні деяких змін до навчального програмного забезпечення на замовлення інших вчителів чи адміністрації навчального закладу для використання на конкретному предметі, уроці. Адже не завжди готові програмні засоби, що відповідають всім вимогам та пройшли апробацію, задовольняють конкретного викладача. Тому й виступає вчитель інформатики в ролі парапрограміста, що здатен формалізувати прикладну задачу користувача та довести її до стану працюючої програми [1]. В більшості випадків для розв'язання саме таких завдань використовують системи візуального програмування, системи швидкого створення додатків (так звані RAD-системи). Даний клас програмного забезпечення дозволяє вирішувати прикладні задачі користувача за допомогою набору засобів візуального програмування, обмежуючи обсяг ручного написання програмного коду. Однак автори таких навчальних програмних засобів недостатню увагу не завжди приділяють системі ознак (призначенню, принципам керування), за якою програми навчального характеру повинні вирізнятися за своєю дидактичною функцією серед іншого програмного забезпечення. Незважаючи на те, що будь-яка програма — це відображення інтелекту автора, його досвіду, матеріально-ідеальний носій його мислення, вона має виконувати множину функцій, типових для більшості програм даного класу з точністю до вибору клавіш-команд, формуючи при цьому саме моторні навички виконання рутинних операцій. Кінцевий користувач не повинен гаяти час на вивчення особливостей роботи з навчальною програмою та її інтерфейсом, адже вони виконують лише допоміжні функції, хоча і дозволяють ознайомитися із найбільш ефективними методами роботи з програмним засобом.

Метою даної публікації є позначення проблеми необхідності дотримання основних вимог, поданих до інтерфейсу користувача, демонстрування важливості та актуальності даного питання під час відбору та створення програмно-

педагогічних засобів навчання.

Що ж все ж таки ми маємо на увазі, коли говоримо про інтерфейс користувача? Яких вимог необхідно дотримуватися при його розробці?

В загальному випадку, інтерфейсом називається сукупність програмно-апаратних засобів, а також погоджень та обмежень, що забезпечують взаємозв'язок компонентів комп'ютера між собою, комп'ютера та периферійного обладнання, комп'ютера з користувачем [2]. Надалі, використовуючи поняття інтерфейсу, матимемо на увазі саме засоби взаємодії комп'ютера (програмного забезпечення) з користувачем, що об'єднанні в поняття інтерфейсу користувача.

Існує кілька типів інтерфейсу користувача, серед яких розрізняють зручний, інтуїтивно зрозумілий, розвинутий, жорсткий та примітивний інтерфейс [3].

Різновидом інтерфейсу є графічний інтерфейс користувача (Graphical User Interface-GUI), що припускає тип екранного уявлення, при якому користувач може виконувати певні операції, вказуючи на піктограми або пункти в переліках меню. Дії можуть виконуватися, як правило, за допомогою миші, або натисканням клавіш на клавіатурі. Рекомендації щодо розробки графічного інтерфейсу користувача (ГІК) засновані на психофізіологічних особливостях людини і підпорядковуються певним принципам. Ігнорування таких принципів призведе до застосування на відчуження, непопулярності серед користувачів [4].

Загальні рекомендації щодо розробки графічного інтерфейсу користувача.

Розглянемо деякі основні рекомендації, яких слід дотримуватися при розробці ГІК.

#### *Багатовіконні застосування.*

У більшості випадків додатки не можуть обмежитися одним вікном, тому, насамперед, потрібно вирішити питання керування вікнами. Є дві різноманітності моделі додатку: з інтерфейсом одного документа (SDI) та інтерфейсом множини документів (MDI). Дуже часто віддають перевагу інтерфейсу SDI, оскільки він передбачає наявність одного вікна та створення додаткових вікон за необхідністю [4]. З іншого боку, у MDI-додатків теж є свої переваги, адже бувають ситуації, коли вигідно відображати інформацію в кількох вікнах, що спільно використовують елементи інтерфейсу (наприклад, меню або інструментальні панелі).

Незважаючи на обраний тип додатка кожне вікно повинно бути старанно продумане і скомпоноване, мати деяку центральну тему, якій підпорядковується його композиція. Ще один принцип, якого треба дотримуватися при проектуванні вікон – стилістична єдність усіх вікон у додатку. Неприпустимо, щоб подібні за функціями елементи керування в різних вікнах називалися по-різному або розміщувалися в різних місцях вікон [4].

#### *Колірне рішення додатку.*

Колір є потужним засобом впливу на психіку людини. Саме тому використовувати його треба дуже обережно. Невдале колірне рішення може призводити до швидкого стомлення користувача, який працює з додатком, до розпорошення його уваги, до частих помилок. Ось чому необхідно намагатися використовувати обмежений набір кольорів і приділяти увагу їх правильному сполученню, пам'ятаючи, що колір не повинен використовуватися як основний

засіб передачі інформації.

Не можна також забувати, що сприйняття кольору дуже індивідуальне, тому не варто нав'язувати своє бачення кольору, навіть якщо воно бездоганне. Треба надати користувачеві можливість самостійного настроювання на найбільше прийнятну для нього гаму [4].

#### *Вимоги до шрифту тексту.*

Змінювати шрифт для вашого додатка треба з певною обережністю. Справа в тому, що шрифт, встановлений на вашому комп'ютері, не обов'язково повинен бути і на комп'ютері користувача. Для попередження ситуації неправильного відображення тексту необхідно додавати до свого додатка ще і файли використаних шрифтів і пояснити користувачеві, як він повинен встановити їх на своєму комп'ютері, якщо вони там відсутні. Інший вихід із положення – ввести в додаток можливість вибору шрифту користувачем [4].

#### *Основні принципи організації меню.*

Практично будь-який додаток повинен мати меню, оскільки саме меню дає найбільш зручний доступ до функцій програми. Основна вимога до меню – їх стандартизація, що полегшує користувачу роботу з додатком.

За звичай, склад меню залежить від конкретного додатка. Але розміщення найпоширеніших команд повинно бути стандартизованим. Головне меню повинно також постачатися інструментальною лінійкою, швидкі кнопки якої дублюють команди меню, що найчастіше використовуються. На цих кнопках треба використовувати, по можливості, і звичні графічні зображення.

Групи функціонально пов'язаних команд відокремлюються у меню роздільниками. Назви команд меню повинні бути звичними користувачеві, стислими і зрозумілими, містити не більше двох слів, оскільки це перевантажує екран і уповільнює вибір користувача.

Назви команд, пов'язані з викликом діалогових вікон, повинні закінчуватися крапками, які вказують на необхідність встановлення в діалоговому вікні додаткових параметрів. Команди, до яких відносяться каскадні меню, повинні закінчуватися стрілкою, що вказує на наявність дочірнього меню даної команди.

У кожній назві команди має бути виділений підкресленням символ, що відповідає клавіші швидкого доступу до команди. Багатьом командам можуть бути поставлені у відповідність «гарячі» клавіші, що дозволяють звернутися до команди даного пункту меню, навіть не заходячи в меню [4].

#### *Підказування і контекстно-залежні довідки.*

Додаток повинен якомога більше полегшувати роботу користувача, забезпечуючи його системою підказок, що допомагають зорієнтуватися в додатку. Ця система має містити вказівники, що спливають у момент, коли користувач затримує курсор миші над деяким елементом вікна додатку; більш розгорнуті підказки, розміщені в панелі стану або в іншому, відведеному для цієї мети місці екрана; вмонтовану систему контекстно-залежної оперативної довідки, що викликається за допомогою клавіші F1.

#### *Висновок.*

Отже, при відборі готових та розробці нових навчальних програмних засобів необхідно всіляко підтримувати одноманітність позначень та дій,

наближатися до обгрунтованості індуктивних узагальнень та задовольняти потребу користувача в орієнтації, застосовувати аналогію.

Розглядаючи місце теми вивчення вимог до інтерфейсу користувача в загальному курсі інформатики для студентів – майбутніх вчителів основ інформатики та обчислювальної техніки – хотілося б зазначити, що невинуватим є вивчення даного питання під час опанування систем програмування [1]. Дотримуючись одного з основних принципів вивчення інформатики-принципу циклічності – слід, на нашу думку, викладати даний матеріал після ознайомлення із застосуваннями, що мають розвинутий інтерфейс користувача, але вже в новому, зміненому контексті, проходячи через ряд узагальнень та конкретизацій. Оптимальним рішенням цього питання, на наш погляд, є введення курсу "Основи візуального програмування" в межах загальної програми з курсу інформатики. Саме даний курс дозволить, окрім основної спрямованості традиційних мов програмування, зосередити увагу на розгляді вищевказаних питань, надаючи розробнику широкі можливості швидкого та якісного проектування інтерфейсу користувача.

#### *Література:*

1. Бочкин А.И. Методика преподавания информатики: Учеб.пособие.-Мн.:Выш.шк.,1998 - 431с.:ил.
2. Голубев А.П., Персональный лабораторный интерфейс//Научно-методический журнал «Информатика й образование».-1996.-№6.-с.88-92.
3. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Общая информатика. Учебное пособие для средней школы.-М.:АСТ-ПРЕСС:Инфорком-Пресс, 1998.-592 с.
4. Архангельский А.Я. Программирование в Delphi5 -М.: ЗАО «Издательство БИНОМ», 2000 г.-1072 с.:ил.

### ОРГАНІЗАЦІЯ САМОСТІЙНОГО ВИВЧЕННЯ ТЕМИ "МНОГОГРАННИКИ І ЇХ ВЛАСТИВОСТІ" ЗА ДОПОМОГОЮ ПЕОМ

*М.В. Дорошенко, Н.В. Летнянчин*

Широке впровадження електронно-обчислювальних машин (ЕОМ) в різноманітні сфери нашого життя потребує вирішення актуального завдання – привести зміст освіти у відповідність з можливостями науково-технічного прогресу і характером сучасної навчальної, наукової і виробничої діяльності.

Одним з напрямків вдосконалення системи освіти є використання навчально-контролюючих програм, які забезпечують організацію навчання з високим ступенем індивідуалізації, можуть функціонувати в режимі діалогу, надають можливість систематично проводити перевірку засвоєного матеріалу на основі розроблених контролюючих завдань, оцінку і облік знань учнів, створювати історію навчання, здійснювати ефективний зворотній зв'язок з кожним учнем, виконувати оцінку ефективності та оптимізації методик навчання. Контролюючі завдання виконують корегуючу, навчальну і діагностичну функцію. Корегуюча функція полягає в тому, що фіксуються і локалізуються помилки, а контролююча програма допомагає учневі ліквідувати їх. Навчальна роль контролюючих завдань полягає в тому, що на основі виявлених нею помилок і утруднень можна визначити кількість і характер завдань

і вправ, необхідних для ліквідації прогалин у знаннях учнів з даної теми. На основі аналізу результатів виконання контролюючих завдань можна зробити мотивований висновок про складність пропонованих завдань, швидкість їх виконання, повноту і міцність засвоєння знань та про індивідуальні можливості учня. В цьому полягає діагностична функція контролюючих завдань.

Можливість високоякісного відтворення графічного зображення в сучасних ПЕОМ дозволяє ефективно використовувати навчальні програми при самостійному вивченні геометрії в старших класах середньої школи.

Метою даної роботи є розробка тестових завдань для перевірки знань учнів з теми "Многогранники і їх властивості" та розробка відповідної навчально-контролюючої програми, оскільки за характером організації традиційний контроль знань здебільшого не забезпечує потрібної оперативності, стійкості та інваріантності оцінки, якості засвоєння навчальної інформації, її достовірності. Крім того уникаються такі фактори, як страх перед учителем та необ'єктивність оцінювання своїх знань.

Реалізації поставленої мети здійснювалась у двох напрямках:

систематизація теми "Многогранники і їх властивості" та розробка навчально-контролюючої програми. Систематизація курсу завершилась підбором навчального матеріалу та розробкою системи контролюючих завдань відповідно до принципів програмованого навчання, яка включала 20 контролюючих вправ, представлених у вигляді тестів.

Розроблена система – навчальна програма із зворотнім зв'язком, тобто вона може не тільки навчати учня, а й перевіряти отримані ним знання. Навчальна програма "Многогранники і їх властивості" дозволяє використовувати комп'ютер для навчання і подання нового матеріалу, самопідготовки та отримання різноманітних довідок, а також для контролю знань.

Програма розроблена з використанням об'єктно-орієнтованої мови програмування DELPHI - 4 і реалізована на комп'ютерах PENTIUM-160 з використанням модульної організації її структури і дозволяє працювати в режимі "навчання і контролю", а також створювати індивідуальну та групову історію навчання.

Інтерфейс користувача здійснюється за допомогою Головного меню навчальної програми, яке дозволяє вибрати для роботи потрібний модуль.

Процес навчання реалізується в модулі "Лекції" шляхом видачі на екран дисплея навчального матеріалу, представленого у вигляді "кроків навчання".

При виборі Правильні многогранники із пункту головного меню Навчання ми дістаємо вікно, у якому можна наочно проглянути загальний вигляд правильних многогранників в різних проекціях.

В програмі реалізовано облік знань учнів на основі обробки результатів тестового контролю.