

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
МІНІСТЕРСТВА ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

Р.В. Клопов

**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У ФІЗИЧНОМУ
ВИХОВАННІ ТА СПОРТІ: ОСОБЛИВОСТІ
РОЗРОБКИ СПЕЦІАЛІЗОВАНОГО
КОМП'ЮТЕРНОГО ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**

Методичні рекомендації для магістрів факультету фізичного виховання
галузі 0102 «фізичне виховання і спорт ».

Затверджено вченою радою ЗНУ
(протокол засідання № 11 від 29.03.2011)

Запоріжжя 2011

УДК 378.147: 796 : 004 (076)

ББК Ч 486.29 я79

К 505

Інформаційні технології у фізичному вихованні та спорті: особливості розробки спеціалізованого комп'ютерного програмного забезпечення: методичні рекомендації для магістрів факультету фізичного виховання галузі 0102 «фізичне виховання і спорт» / Уклад. Р.В. Клопов – Запоріжжя: ЗНУ, 2011. – 41 с.

Методичні рекомендації з курсу «Інформаційні технології у фізичному вихованні та спорті» розроблені згідно з навчальною програмою для магістрів факультету фізичного виховання як складова навчально-методичного комплексу дисципліни спрямовані на формування знань, вмінь та навичок застосування інформаційних технологій у майбутній професійній діяльності. Методичні рекомендації містять: теоретичну інформацію щодо загальних принципів розробки програмного забезпечення, аналіз сучасного інструментарію, за допомогою якого здійснюється розробка прикладного програмного забезпечення, особливості розробки спеціалізованого програмного забезпечення в галузі фізичного виховання та спорту, а також перелік рекомендованої літератури. Призначені для магістрів галузі 0102 «фізичне виховання і спорт».

Рецензенти: к.п.н., доцент А.В.Сватьєв

Відповідальний за випуск Р.В.Клопов

ЗМІСТ

ВСТУП	4
1. ЗАГАЛЬНІ ПРИНЦИПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	5
1.1. Поняття комп'ютерного забезпечення	5
1.2. Програмне забезпечення інформаційних технологій	8
1.3. Життєвий цикл програмного забезпечення	11
1.4. Поняття технології програмування	14
1.5. Сучасні системи програмування	20
2. ОСОБЛИВОСТІ РОЗРОБКИ СПЕЦІАЛІЗОВАНИХ ПРОГРАМ В ГАЛУЗІ ФІЗИЧНОГО ВИХОВАННЯ ТА СПОРТУ	24
ГЛОСАРІЙ.....	35
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39

ВСТУП

В сучасних умовах стрімкого розвитку інформаційних технологій система освіти України зазнає істотних трансформацій, пов'язаних з бурхливим розвитком телекомунікації: широким впровадженням мережі Інтернет, розвитком обчислювальної техніки, впровадженням новим технологій накопичення й обробки великих обсягів інформації тощо.

Вища фізкультурна освіта не може бути осторонь інформатизації системи вищої освіти, тому застосування сучасних інформаційних технологій у навчальному процесі з урахуванням специфіки професійної підготовки майбутніх фахівців фізичного виховання та спорту на сьогодні актуально.

Методичні рекомендації розроблені для навчально-методичного комплексу дисципліни «Інформаційні технології у фізичному вихованні та спорті» магістрів усіх спеціальностей галузі знань «Фізичне виховання і спорт» на основі навчальної програми курсу «Інформаційні технології у фізичному вихованні та спорті».

Викладений матеріал сприяє формуванню знань, вмінь і навичок застосування спеціалізованого програмного забезпечення у майбутній професійній діяльності і навчанні впродовж всього життя та може бути використаний для самостійної та індивідуальної роботи студента.

1. ЗАГАЛЬНІ ПРИНЦИПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Поняття комп'ютерного забезпечення

Технічну основу забезпечення різноманітних інформаційних технологій складають засоби комп'ютерної, телекомунікаційної та організаційної техніки.

Засоби комп'ютерної техніки є основою всього комплексу технічних засобів інформаційних технологій. Вони, в першу чергу, призначені для автоматичної обробки великих масивів різних видів інформації, що застосовується в різних сферах людської діяльності.

Засоби телекомунікаційної техніки забезпечують обмін інформацією між її споживачами. Застосування цих засобів наразі також цілком базується на використанні комп'ютерної техніки.

Засоби організаційної техніки призначені для механізації та автоматизації управлінської діяльності в усіх її проявах.

Сучасний комп'ютер – це електронний прилад, який призначений для автоматизації різноманітних операцій з даними. Гнучкість автоматизації досягається завдяки тому, що операції з даними виконуються за заздалегідь заготовленими і легко змінюваними *програмами*. Універсальність комп'ютерів базується на тому принципі, що будь-які типи даних представляються в них за допомогою універсального двійкового кодування.

Наразі існує досить багато різновидів комп'ютерів. Єдиної їх класифікації не існує. Так, наприклад, досить часто комп'ютери відносять до одного з чотирьох поколінь.

Комп'ютери першого покоління (1948–1958 рр.) були побудовані на базі електронно-вакуумних ламп. Вони мали незначну потужність, великі розміри і низьку надійність, оскільки електронні лампи за своїм принципом роботи часто виходять з ладу.

Комп'ютери другого покоління (1959–1967 рр.) були побудовані із

застосуванням напівпровідникового пристрою – транзистора. На відміну від попереднього типу обчислювальної техніки вони мали значно вищу продуктивність і менші розміри. Крім того, транзистори на відміну від ламп, мали значно більший ресурс роботи.

Комп'ютери першого та другого покоління часто називають терміном мейнфрейм (mainframe), який означає велику багатокористувацьку електронну обчислювальну машину (ЕОМ).

Комп'ютери третього покоління (1968–1973 рр.) було побудовано із застосуванням великих інтегральних схем (чипів), які утворювалися з сотень або тисяч мікротранзисторів, розташованих на невеликій електронній схемі (мікросхемі). На відміну від мейнфреймів такі комп'ютери мали значно менші розміри і зазвичай більш високу продуктивність. Цей тип комп'ютерів прийнято називати мініЕОМ.

Комп'ютери четвертого покоління (починаючи з 1974 р.) побудовані із застосуванням надвеликих інтегральних схем (мікрочипів), що утворені з мільйонів мікротранзисторів. Такі комп'ютери прийнято називати мікроЕОМ. У наш час цей термін вважається застарілим, натомість замість нього вживають термін персональний комп'ютер (ПК).

Оскільки комп'ютери першого-третього поколінь є застарілими і зараз майже не зустрічаються, для класифікації ПК, починаючи з 1999 р. застосовується міжнародний сертифікаційний стандарт – специфікація PC99. Відповідно до неї виділяють такі категорії персональних комп'ютерів:

- масовий ПК (consumer PC);
- офісний ПК (office PC);
- мобільний, переносний ПК (mobile PC);
- робоча станція (workstation PC);
- розважальний ПК (entertainment PC).

Масовий ПК призначений для домашнього використання. Його можливості та продуктивність повинні задовольняти більшість пересічних користувачів.

Офісний ПК – це окрема категорія персональних комп'ютерів, яка характеризується відповідністю визначеним, досить жорстким критеріям і оригінальними особливостями. По-перше, такі комп'ютери повинні забезпечувати надійну й безперебійну роботу протягом тривалого часу, а, по-друге, бути адаптованими для роботи з так званими офісними додатками (текстовими та табличними процесорами, базами даних тощо). По-третє, офісні ПК повинні мати можливість легко з мінімальними фінансовими та часовими витратами модифікуватися для забезпечення роботи в нових умовах або з новим програмним забезпеченням.

Переносний ПК – це окремий вид портативних комп'ютерів, призначений для застосування в дорожніх умовах. За своїми характеристиками такі комп'ютери бувають досить різними, але на відміну від двох звичайних ПК їх відрізняє досить велика вартість.

Робоча станція – це комп'ютер, що задовольняє яким-небудь особливим вимогам, що висуваються до потужного комп'ютера для конкретного виду інтелектуальної професійної діяльності. Зазвичай – це спеціалізовані комп'ютери з відмовостійким обладнанням і потужними графічними можливостями. Робочі станції, наприклад, застосовуються програмістами для *розробки програмного забезпечення*. Іншим прикладом може слугувати спеціалізований комп'ютер для обробки графічних зображень – графічна робоча станція. І так далі.

Розважальні ПК – це підвид масових комп'ютерів, які мають потужне графічне та мультимедійне обладнання і призначені для високоякісного відтворення тривимірної графіки та звуку.

Також досить часто сучасні персональні комп'ютери класифікують за їх розмірами:

- настільні (desktop, nettop);
- портативні (notebook, netbook);
- кишенькові (palmtop).

1.2. Програмне забезпечення інформаційних технологій

Як уже зазначалося, для автоматизації обробки інформації із застосуванням комп'ютера необхідне програмне забезпечення (ПЗ). Залежно від функцій, що виконуються програмним забезпеченням, його можна поділити на дві групи: системне й прикладне. Системне ПЗ організовує процес обробки інформації на комп'ютері й забезпечує нормальне робоче середовище для прикладних програм.

До складу системного ПЗ входять:

- операційні системи;
- програми технічного обслуговування;
- транслятори;
- сервісні програми тощо.

Операційна система (ОС) – це сукупність програм, що управляє апаратними й програмними ресурсами комп'ютера, а також забезпечує виконання команд користувача. В наш час найбільшого поширення отримали багатозадачні системи (такі, що можуть одночасно виконувати багато програм) з графічним інтерфейсом користувача. Серед найбільш поширених операційних систем можна виділити Microsoft Windows (версії XP, Vista, Seven) та Linux (дистрибутиви Debian, Ubuntu, Gentoo тощо).

Сервісне програмне забезпечення – це сукупність програмних продуктів, що надають користувачам додаткові послуги в роботі з комп'ютером і розширюють можливості ОС. Прикладом сервісного ПЗ є спеціалізовані програми, що забезпечують роботу апаратних пристроїв комп'ютера – *драйвери*.

Програми технічного обслуговування призначені для діагностики і виявлення помилок у процесі роботи комп'ютера або обчислювальної системи в цілому. Вони включають засоби діагностики і тестового контролю правильності роботи комп'ютера та його окремих частин, а також автоматичного усунення помилок і несправності. Досить часто сервісне та технічно-діагностичне ПЗ входить до складу сучасних ОС, як їх невід'ємна

складова. До таких програм можна, наприклад, віднести відомий програмний пакет Norton Utilities.

Транслятором називається програма, що здійснює переклад тексту програми з однієї мови програмування на іншу. Розрізняють два типи трансляторів: *компілятори* та *інтерпретатори*. Компілятор здійснює переклад тексту програми на деякій мові програмування в машинний код. Інтерпретатор – виконує покроковий переклад програми на деякій алгоритмічній мові в машинний код і тут же його виконує.

Прикладне програмне забезпечення призначено для вирішення конкретних завдань користувача. Воно дає змогу розробляти і виконувати завдання (додатки) користувача, наприклад, при роботі з інформаційними системами, в навчальному процесі, в управлінні персоналом тощо. Прикладне програмне забезпечення працює під управлінням системного ПЗ. До складу прикладного ПЗ зазвичай входять:

- прикладні програми загального призначення;
- прикладні програми функціонального призначення.

Програми загального призначення – це універсальні програмні продукти, що призначені для автоматизації розробки й експлуатації функціональних завдань користувача в цілому. До цього класу ПЗ зазвичай відносяться:

- текстові редактори;
- графічні редактори;
- електронні таблиці;
- системи управління базами даних тощо.

Тобто все те, що зазвичай прийнято зараховувати до офісного програмного забезпечення.

До прикладних *програм функціонального призначення* належать програмні продукти, орієнтовані на автоматизацію функцій користувача, наприклад, у конкретній сфері його діяльності. До цього класу, наприклад, належать пакети програм з бухгалтерського обліку, з обробки цифрових

зображень, верстання друкованої продукції, управління персоналом, географічних інформаційних систем тощо.

Найбільш важливим елементом програмного забезпечення будь-якої обчислювальної системи є операційна система. Фактично застосування сучасних комп'ютерів без ОС неможливе. У наш час існує велика кількість ОС для персональних комп'ютерів, для робочих станцій, для серверів тощо. Всі ОС можна умовно поділити на два класи: однозадачні й багатозадачні.

Однозадачні ОС у один момент часу можуть виконувати одну програму (задачу). Іншими словами вони дають змогу виконуватися програмам тільки послідовно. Найбільш відомим прикладом однозадачної ОС є MS DOS фірми Microsoft.

На відміну від однозадачних багатозадачні ОС можуть виконувати програми одночасно. Якщо на комп'ютері встановлено декілька процесорів (обчислювальних пристроїв), то обробка даних в багатозадачних ОС може виконуватися в так званому паралельному режимі. Якщо ж процесор один, то ОС організує роботу всіх програм у так званому псевдопаралельному режимі, коли у користувача вдається враження одночасної роботи всіх запущених ним програм.

Оскільки багатозадачні ОС є більш продуктивними всі сучасні ОС, які застосовуються на більшості ПК, є багатозадачними. Найбільш відомими прикладами багатозадачних систем є Windows, Linux, OS/2, UNIX, FreeBSD та інші.

До найбільш поширених програм загального призначення відносяться так звані текстові редактори, що призначені для створення й редагування текстових документів різної складності: від простого набору тексту до видавничої верстки. Головне призначення текстових редакторів – автоматизація документообігу. Найбільш популярними текстовими редакторами в наш час є MS Word, що входить до поширеного комерційного пакету Microsoft Office, а також безкоштовні редактори Lotus Symphony, OpenOffice, StarOffice, LibreOffice тощо.

Другими за популярністю програмами загального призначення є електронні таблиці, які широко застосовуються для автоматизації вирішення різноманітних класів фінансово-економічних задач. Електронною таблицею називається програма, яка призначена для обробки даних у табличній формі. Дані в таблиці зберігаються у комірках, що знаходяться на перетині рядків і стовпців. У комірках можуть зберігатися числа, символьні дані та формули. Формули задають функціональну залежність значення одних комірок від вмісту інших. Зміна вмісту довільної комірки автоматично призводить до змін значень у всіх пов'язаних з нею комірках. Найбільш поширеними електронними таблицями є MS Excel з пакету Microsoft Office, а також Lotus, Quattro Pro та інші.

База даних (БД) це сукупність спеціальним чином організованих наборів даних, що зберігаються на електронних носіях. Управління базою даних включає введення даних, корекцію і маніпулювання даними (додавання, видалення, оновлення та ін.). Для забезпечення цих стандартних операцій з БД використовуються спеціалізовані програми – системи управління базами даних (СУБД). Із наявних СУБД найбільшу популярність отримали MS Access із пакету Microsoft Office, Paradox, Sybase, Interbase, MS SQL Server, Oracle, MySQL та інші.

1.3. Життєвий цикл програмного забезпечення

Розробка якісного програмного забезпечення повинна враховувати тривалий процес, який прийнято називати *життєвим циклом програмного забезпечення*. Цей процес розпочинається з початкової ідеї, містить написання та налагодження програми і зазвичай триває багато років, протягом яких у програмне забезпечення вносяться певні зміни й покращення (цей етап життєвого циклу програмного забезпечення прийнято називати *супроводом*). На фінальній стадії життєвого циклу програми відбувається процес виведення її з експлуатації.

Зазвичай життєвий цикл програмного забезпечення утворений такими етапами.

1. *Постановка задачі* – процес виявлення та формального опису вимог до майбутньої програми (дуже складний та важливий етап оскільки помилки, припущені в цей час інколи буває дуже складно виправити в майбутньому).

2. *Проектування* – процес побудови математичної моделі предметної області, для якої створюється програма, та побудови алгоритму її майбутньої роботи (для його подальшого кодування із застосуванням певної алгоритмічної мови).

3. *Оцінка ризику* – створення програмного забезпечення пов'язане з ризиком. Деякі проблеми притаманні всім проектам, а деякі характерні лише для певних розробок. Деякі з них можна передбачити, в той час інші – ні. Вони можуть впливати на графік і вартість виконання робіт, економічні успіхи і навіть на життя і здоров'я людей. Деякі небезпеки можна запобігти або зменшити, а деякі – ні. Для ідентифікації, оцінки та запобігання небезпечностей, що виникають при розробці програмного забезпечення, існують спеціальні методи. Результат оцінки ризику впливає на всі етапи життєвого циклу програмного забезпечення.

4. *Верифікація* – це процес перевірки правильності алгоритмів, покладених в основу програмного забезпечення. Хоча повністю задача верифікації ПЗ ще повністю не вирішена (на практиці неможливо стовідсотково довести, що програма не має помилок), варто зазначити, що існують певні процедури верифікації програм.

5. *Кодування* полягає в перекладі алгоритму на конкретну мову програмування (алгоритмічну мову) з подальшим виправленням синтаксичних помилок. Зазвичай прийнято вважати, що саме кодування і є власне програмуванням. Однак все ж слід розуміти, що кодування – це не саме головне, це лише один з етапів життєвого циклу програмного забезпечення.

6. *Тестування* – це етап, на якому здійснюється виявлення та виправлення логічних помилок в алгоритмі роботи програми. Тестування програмного забезпечення один з найскладніших етапів. Це – одночасно і наука, і мистецтво.

7. *Уточнення рішення*. Результатом виконання попередніх етапів є працююча програма, яку інтенсивно тестували і налагоджували. Якщо програма дійсно вирішує поставлену задачу, виникає питання: навіщо уточнювати рішення? Краще за все вирішувати завдання при найбільш простих припущеннях, поступово ускладнюючи програму. Наприклад, можна припустити, що вхідні дані мають певний формат і є правильними. Створивши найпростіший варіант, можна доповнювати його більш складними процедурами введення і виведення даних, оснащувати додатковими можливостями та засобами для виявлення помилок. Таким чином, якщо застосовується підхід «від простого – до складного», етап уточнення рішення стає необхідним. Зрозуміло, що остаточне уточнення рішення не повинно призводити до повного перегляду програми. Кожне уточнення рішення є досить очевидним, особливо якщо програма має модульну структуру. Фактично поступове уточнення рішення являє собою основну перевагу модульного підходу до розробки програм. Крім того, після кожної, навіть найпростішої, модифікації програми, її потрібно знову ретельно протестувати.

8. *Виробництво*. Після завершення розробки програмного продукту він поширюється серед користувачів, інсталюється на їх комп'ютерах і застосовується на практиці.

9. *Супровід* – підтримка програми в експлуатації, коли користувачі програм можуть виявити помилки, що залишилися непоміченими при тестуванні. Крім того, з часом програмне забезпечення потрібно вдосконалювати, додаючи в нього нові функціональні можливості або модифікуючи його компоненти.

Як показує досвід, етапи життєвого циклу програмного забезпечення не

ізолювані один від одного і можуть не слідувати один за іншим. Крім того, слід зазначити, що всі етапи життєвого циклу програмного забезпечення повинні ретельно документуватися. Документування програми не є окремим етапом її життєвого циклу, як можна було б подумати, а супроводжує її протягом усього життя.

1.4. Поняття технології програмування

Технологія програмування – це сукупність методів і засобів розробки (написання) програмного забезпечення та порядок застосування цих методів і засобів.

На ранніх етапах розвитку програмування, коли програми писалися у вигляді послідовностей машинних команд, будь-які технології програмування були відсутні. Перші кроки в розробці технології полягали в представленні програми у вигляді послідовності операторів. Написанню послідовності машинних команд передувало складання операторної схеми, що відбиває послідовність операторів і переходи між ними. Операторний підхід дозволив розробити перші програми для автоматизації складання програм – так звані *складові програми*.

Зі збільшенням розмірів програм стали виділяти їх відокремлені частини і оформляти їх в якості так званих *підпрограм*. Частина таких підпрограм об'єднувалася в бібліотеки, з яких підпрограми можна було включати в робочі програми і потім викликати з робочих програм. Це поклало початок новій технології програмування – *процедурному програмуванню*, коли велика програма представлялася у вигляді певної сукупності підпрограм (або *процедур*). Одна з підпрограм обов'язково є головною і з неї розпочинається виконання програми.

У 1958 році були розроблені перші мови програмування Фортран (FORTRAN) та Алгол-58 (ALGOL-58). Програма на мові Фортран складалася з головної програми і деякої кількості процедур – підпрограм і *функцій*.

Програма на Алгол-58 і його подальшої версії Алгол-60 представляла собою єдине ціле, але мала блокову структуру, що містить головний блок і вкладені блоки підпрограм і функцій. Компілятори для Фортрану забезпечували роздільну трансляцію процедур і наступне їх об'єднання в робочу програму, перші компілятори для Алгол припускали, що транслюється відразу вся програма, роздільна трансляція процедур не забезпечувалася.

Процедурний підхід зажадав структурування майбутньої програми, поділу її на окремі процедури. При розробці окремої процедури про інші процедури було потрібно знати тільки їх призначення та спосіб виклику з головної програми або іншої підпрограми. З'явилася можливість переробляти окремі процедури, не зачіпаючи решти програми, скорочуючи при цьому витрати праці та машинного часу на розробку і модернізацію програм.

Наступним кроком у поглибленні структурування програм стало так зване *структурне програмування*, при якому програма в цілому і окремі процедури розглядалися як послідовності канонічних структур: *лінійних ділянок, циклів і розгалужень*. З'явилася можливість читати і перевіряти програму як послідовний текст, що підвищило продуктивність праці програмістів при розробці, налагодженні програм, а також при супроводі чужих програм (що є дуже трудомістким і складним процесом). З метою підвищення структурності програми були висунуті вимоги до більшої незалежності підпрограм, підпрограми повинні зв'язуватися з іншими програмними модулями тільки шляхом передачі їм аргументів, використання в підпрограмах змінних, що належать іншим процедурам або головній програмі, стало вважатися небажаним.

Процедурне і структурне програмування торкнулися насамперед процес опису алгоритму як послідовності кроків, що ведуть від варійованих вихідних даних до шуканого результату. Для вирішення спеціальних завдань стали розроблятися мови програмування, орієнтовані на конкретний клас задач: на системи управління базами даних, імітаційне моделювання, системне програмування, комп'ютерну графіку і таке інше.

При розробці трансляторів все більше уваги стало приділятися виявленню помилок у вихідних текстах програм, забезпечуючи цим скорочення витрат часу на налагодження програм.

Застосування програм у різних областях людської діяльності призвело до необхідності підвищення надійності всього програмного забезпечення. Одним з напрямків вдосконалення мов програмування стало підвищення рівня типізації даних. Теорія типів даних виходить з того, що кожне використовуване в програмі дане належить одному і тільки одному *типу даних*. Тип даного визначає безліч можливих значень цього та набір операцій, допустимих над цими даними. Дане конкретного типу в ряді випадків може бути перетворено в даний іншого типу, але таке перетворення повинно бути явно представлено в програмі. Залежно від ступеня виконання перерахованих вимог можна говорити про рівень типізації тієї чи іншої мови програмування. Прагнення підвищити рівень типізації мови програмування призвело до появи мови Паскаль (Pascal), яка вважається строго універсальною, хоча й у ній дозволені деякі неявні перетворення типів, наприклад, цілого в дійсне тощо. Застосування строго типізованого мови при написанні програми дозволяє ще на етапі трансляції вихідного тексту виявити багато помилок використання даних і цим підвищити надійність програми. Разом з тим суворі типізація сковувала свободу програміста, ускладнювала застосування деяких прийомів перетворення даних, часто використовуваних в системному програмуванні. Практично одночасно з мовою Паскаль була розроблена мова Сі (C), більшою мірою орієнтована на системне програмування (із її допомогою було написано більшу частину відомої ОС UNIX).

Всі універсальні мови програмування, незважаючи на відмінності в синтаксисі і використовуваних ключових словах, реалізують одні й ті ж канонічні структури: оператори присвоювання, цикли і розгалуження (умовні оператори). У всіх сучасних мовах присутні базові типи даних (цілі й дійсні арифметичні типи, символьні тощо), є можливість використання агрегатів даних, у тому числі масивів і структур (записів). Для арифметичних даних

дозволені звичайні арифметичні операції, для агрегатів даних зазвичай передбачена лише операція присвоювання і можливість звернення до елементів агрегату. Разом з тим при розробці програми для вирішення конкретної прикладної задачі бажана велика концептуальна близькість тексту програми до опису завдання. Наприклад, якщо розв'язок завдання вимагає виконання операцій над комплексними числами або квадратними матрицями, бажано, щоб у програмі явно були присутні оператори додавання, віднімання, множення і ділення даних типу комплексного числа, додавання, віднімання, множення й обернення даних типу квадратної матриці тощо.

Вирішення цієї проблеми можливе декількома шляхами.

1. Побудовою мови програмування, що містить якомога більше типів даних, і вибором для кожного класу задач деякого підмножини цієї мови (створення універсальної алгоритмічної мови). Такі мови програмування інколи називають *мовами-оболонками*. На цю роль у 70-х роках минулого сторіччя претендувала мова програмування ПЛ/1 (PL/1), що виявилася настільки складною, що так і не вдалося побудувати її формалізований опис та створити відповідний транслятор. Хоча певний час ця мова програмування й набула широкого застосування як у країнах Західній Європі, так і в колишньому СРСР.

2. Побудовою розширюваної мови, що містить невелике ядро і припускає його розширення новими типами даних і операторами, які відображають концептуальну сутність конкретного класу задач. Таку мову називають *мовою-ядром*. Одними з перших прикладів таких мов програмування були Сімула (Simula) й Алгол-68, які не отримали широкого розповсюдження, але спричинили великий вплив на розробку інших мов програмування.

Подальшим розвитком другого шляху став *об'єктно-орієнтований підхід до програмування*, основні ідеї якого спираються на наступні положення:

- програма є моделлю деякого реального процесу або частини

реального світу;

- модель реального світу або його частини може бути описана як сукупність взаємодіючих між собою об'єктів;
- *об'єкт* описується набором параметрів, значення яких визначають стан об'єкта, і набором операцій (дій), які може виконувати об'єкт;
- взаємодія між об'єктами здійснюється посилкою спеціальних повідомлень від одного об'єкта до іншого;
- повідомлення, отримане об'єктом, може вимагати виконання певних дій, наприклад, зміни стану об'єкта;
- об'єкти, описані одним і тим же набором параметрів і здатні виконувати один і той же набір дій, являють собою клас однотипних об'єктів.

З точки зору мови програмування клас об'єктів можна розглядати як тип даного, а окремий об'єкт – як дане цього типу. Визначення програмістом власних класів об'єктів для конкретного набору завдань повинно дозволити описувати окремі завдання в термінах самого класу завдань (при відповідному виборі імен типів і імен об'єктів, їх параметрів і виконуваних дій).

Таким чином, об'єктно-орієнтований підхід передбачає, що при розробці програми повинні бути визначені класи використовуваних в програмі об'єктів і побудовані їх описи, потім створені екземпляри необхідних об'єктів і визначено взаємодію між ними.

Класи об'єктів часто зручно будувати так, щоб вони утворювали ієрархічну структуру. Наприклад, клас «Студент», що описує абстрактного студента, може служити основою для побудови класів «Студент 1 курсу», «Студент 2 курсу», «Деканат» і т.д., які володіють усіма властивостями студента взагалі і деякими додатковими властивостями, що характеризують, наприклад, студента конкретного курсу. При розробці інтерфейсу з користувачем програми можуть використовувати об'єкти загального класу «Вікно» і об'єкти класів спеціальних вікон, наприклад, вікон інформаційних повідомлень, вікон введення даних і т.п. У таких ієрархічних структурах

один клас може розглядатися як базовий для інших, похідних від нього класів. Об'єкт похідного класу має всі властивості базового класу і деякими власними властивостями, він може реагувати на ті ж типи повідомлень від інших об'єктів, що і об'єкт базового класу і на повідомлення, що мають сенс тільки для похідного класу. Зазвичай кажуть, що об'єкт похідного класу успадковує всі властивості свого базового класу.

Деякі параметри об'єкта можуть бути локалізовані всередині об'єкта і недоступні для прямого впливу ззовні об'єкта. Наприклад, під час руху об'єкта-автомобіля об'єкт-водій може впливати тільки на обмежений набір органів управління (рульове колесо, педалі газу, зчеплення і гальма, важіль перемикання передач) і йому недоступний цілий ряд параметрів, що характеризують стан двигуна і автомобіля в цілому.

Очевидно, для того, щоб продуктивно застосовувати об'єктний підхід для розробки програм, необхідні мови програмування, що підтримують цей підхід, тобто дозволяють будувати опис класів об'єктів, утворювати дані об'єктних типів, виконувати операції над об'єктами. Однією з перших таких мов стала SmallTalk в якій всі дані є об'єктами деяких класів, а загальна система класів будується як ієрархічна структура на основі визначених базових класів.

Досвід програмування показує, що будь-який методичний підхід в технології програмування не повинен застосовуватися сліпо з ігноруванням інших підходів. Це відноситься і до об'єктно-орієнтованого підходу. Існує ряд типових проблем, для яких його корисність найбільш очевидна, до таких проблем відносяться, зокрема, завдання імітаційного моделювання, програмування діалогів з користувачем. Існують і завдання, в яких застосування об'єктного підходу ні до чого, крім зайвих витрат праці, не приведе. У зв'язку з цим найбільшого поширення набули об'єктно-орієнтовані мови програмування, що дозволяють поєднувати об'єктний підхід з іншими методологіями. У деяких мовах і системах програмування застосування об'єктного підходу обмежується засобами інтерфейсу з

користувачем (наприклад, Visual FoxPro ранніх версій).

Найбільш використовуваними в даний час об'єктно-орієнтованими мовами є Об'єктний Паскаль (Object Pascal) і C++, причому найбільш розвинені засоби для роботи з об'єктами містяться в C++.

Практично всі об'єктно-орієнтовані мови програмування є мовами, що розвиваються, їх стандарти регулярно уточнюються і розширюються. Наслідком цього розвитку є неминучі відмінності у вхідних мовах компіляторів різних систем програмування. Одними з найбільш поширених в наш час є системи програмування Microsoft Visual Studio (що підтримує програмування із застосуванням таких мов програмування, як C++, Visual Basic, Java, C# та ін.) і Embarcadero Rad Studio (підтримує роботу з мовами C++, Object Pascal і т.д.).

1.5. Сучасні системи програмування

Microsoft Visual Studio – це серія продуктів фірми Microsoft, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як прості програми без графічного інтерфейсу (так звані *консольні програми*), так і програми з графічним інтерфейсом, а також веб-сайти, веб-додатки, веб-служби тощо.

Різні версії Visual Studio включають одну або декілька наступних мов програмування:

Visual Basic.NET;

Visual C++;

Visual C#;

Visual J#;

Visual F.

Загальний вигляд системи програмування Microsoft Visual Studio

наведено на рис. 1.1.

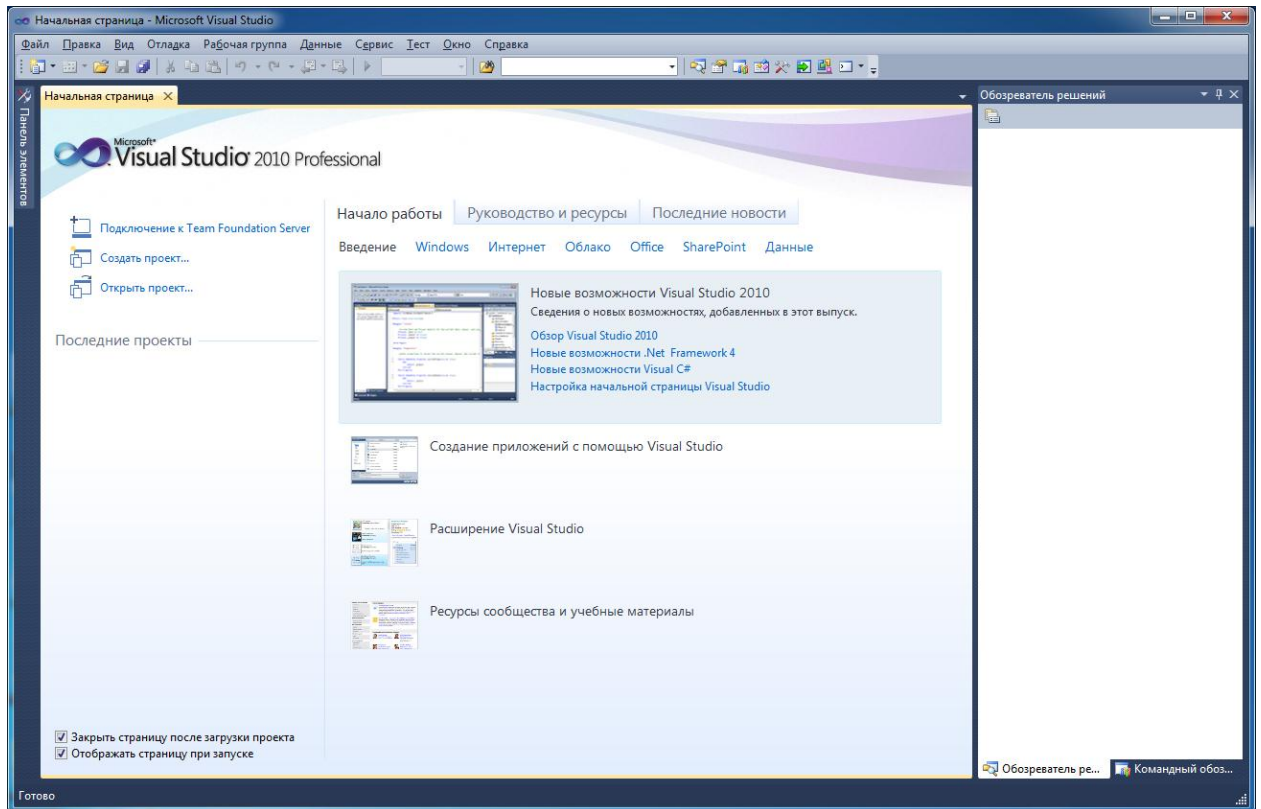


Рис. 1.1. Система програмування Microsoft Visual Studio

Embarcadero Rad Studio – це сучасне інтегроване середовище швидкої розробки програмного забезпечення для роботи в середовищі ОС Microsoft Windows. Воно підтримує розробку Windows-застосунків на мовах програмування Delphi, яка є наступницею мови Object Pascal, і C++.

Delphi в основному використовується для розробки настільних застосунків та корпоративних СУБД, проте цей інструмент можна використовувати для розробки будь-якого загального програмного забезпечення. Не залишена осторонь і можливість побудови веб-застосунків.

Embarcadero Rad Studio поширюється у кількох редакціях з різними можливостями і цінами: Personal, Professional, Enterprise.

Загальний вигляд системи програмування Embarcadero Rad Studio наведено на рис. 1.2.

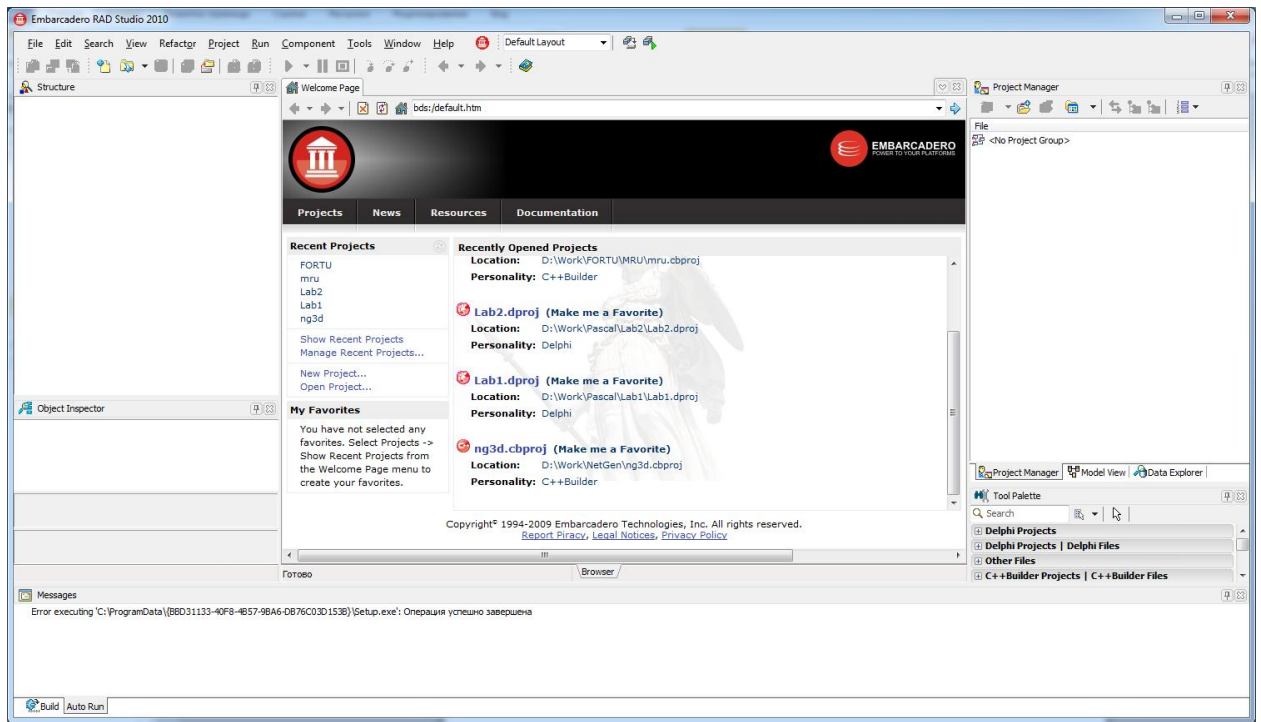


Рис. 1.2. Система програмування Embarcadero Rad Studio

Qt Creator – інтегроване середовище розробки програмного забезпечення, призначене для створення так званих *крос-платформових застосунків* (тобто таких програм, що здатні виконуватися на різних типах комп'ютерів під управлінням різних операційних систем) з використанням бібліотеки Qt. Підтримується розробка як класичних програм мовою C++, так і використання мови QML, для визначення сценаріїв в якій використовується JavaScript, а структура і параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt Creator може використовувати компілятори мови програмування C++ – GCC або Microsoft VC++ і GDB як засіб верифікації програм. Для Windows версій бібліотека комплектується компілятором, заголовними і об'єктними файлами MinGW.

На відміну від двох попередніх систем програмування Qt Creator є повністю безкоштовним програмним продуктом. Програми, створені в цій системі, без істотних змін можуть одночасно запускатися в середовищі таких операційних систем як Windows, Linux та MacOS.

Загальний вигляд системи програмування Qt Creator наведено на

рис. 1.3.

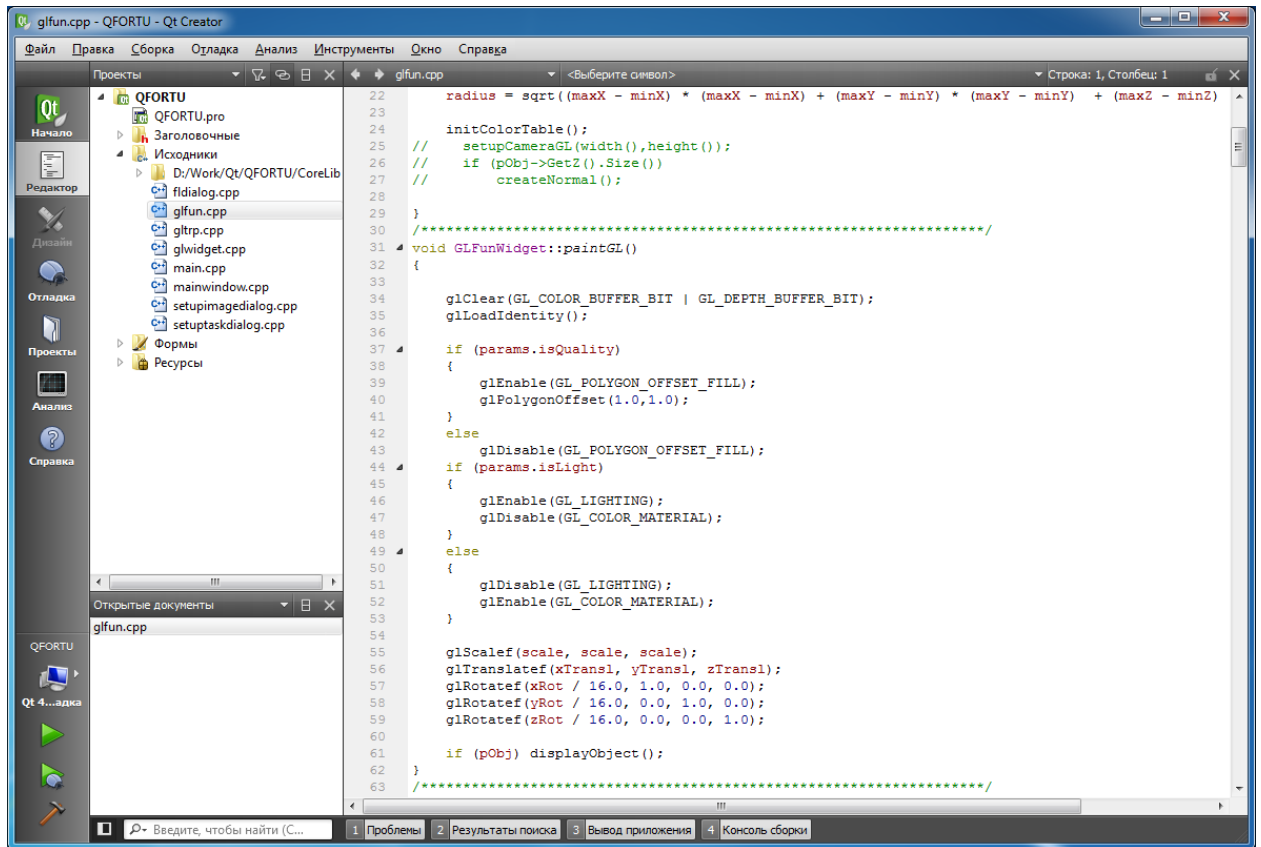


Рис. 1.3. Система програмування Qt Creator

2. ОСОБЛИВОСТІ РОЗРОБКИ КОМП'ЮТЕРНИХ СПЕЦІАЛІЗОВАНИХ ПРОГРАМ В ГАЛУЗІ ФІЗИЧНОГО ВИХОВАННЯ ТА СПОРТУ

Умовою здійснення майбутніми фахівцями ефективної професійної діяльності в сучасному динамічному розвитку суспільства є система цілеспрямованої та взаємопов'язаної сукупності комп'ютерно-орієнтованих методів та навчально-педагогічних засобів, спрямованих на формування у студентів відповідного рівня знань, умінь і практичних навичок застосування інформаційних технологій. Також у сучасному інформаційному суспільстві важливим фактором професійної успішності є сформованість навичок застосування інформаційних технологій в процесі здобуття вищої освіти та освіти впродовж всього життя, які вони отримують із застосуванням інформаційних технологій, що дозволить значно підвищити рівень якості професійно значущих знань, вмінь та навичок.

На нашу думку, одним з найважливіших компонентів електронних навчально-методичних комплексів дисципліни є програмне забезпечення, яке дозволить викладачу проводити моніторинг сформованості професійно значущих якостей майбутнього фахівця з фізичного виховання та спорту. З цією метою на основі розроблених нами методик створене комп'ютерне програмне забезпечення: для педагогічного моніторингу оцінювання рівня готовності майбутніх фахівців з фізичного виховання та спорту до застосування інформаційних технологій – «RTEST: моніторинг формування готовності майбутніх фахівців фізичного виховання та спорту до застосування інформаційних технологій»; «моніторинг формування готовності майбутніх фахівців фізичного виховання та спорту до впровадження технології спортивного менеджменту програмно-тестувальний комплекс «БСК»» і «R_local»: інформаційна система «База знань» для накопичення та систематизації електронних навчальних ресурсів вищого

навчального закладу. На прикладі розробленого комп'ютерного програмного забезпечення детально розглянемо етапи його розробки.

На першому етапі розробляються анкети для оцінки обраної професійно значущої якості майбутнього фахівця з фізичного виховання й спорту. Обов'язковою умовою є застосування методики оцінки мотивації до майбутньої професійної діяльності. В процесі розробки анкет для тестування на основі аналізу науково-методичної літератури й особистого педагогічного досвіду виділяються компоненти готовності майбутнього фахівця з фізичного виховання й спорту до застосування професійно значущих якостей. На основі цього й створюються анкети, які дозволяють виявити рівні сформованості виділеної професійно значущої якості майбутнього фахівця.

На другому етапі розробляються оцінні шкали для обробки результатів анкетування. Теоретичний аналіз наукових педагогічних робіт показав, що як правило це три рівні готовності (високий, середній, низький). У деяких випадках, як наприклад у нашому дослідженні, з'являється необхідність в одержанні інтегрального (узагальнюючого) показника готовності майбутнього фахівця до застосування професійно значущої якості у майбутній професійній діяльності. Для цього нами використана цільова функція з ваговими коефіцієнтами, які відображають рівень значимості компонентів готовності. Вагові коефіцієнти нами були визначені за допомогою методу експертних оцінок.

На третьому етапі розробляється алгоритм програмного забезпечення. Програмісту в словесній або письмовій формі описується технічне завдання на розробку програмного продукту.

Технічне завдання повинне містити:

- зовнішній вигляд (інтерфейс) програмного забезпечення (див. рис. 2.1).
- введення персональних даних і результатів тестування (див. рис. 2.2).;
- анкетування (див. рис. 2.3);
- можливості обробки і редагування результатів тестування (див. рис. 2.4);

– можливості експорту даних і статистичної обробки результатів тестування (рис. 2.5);

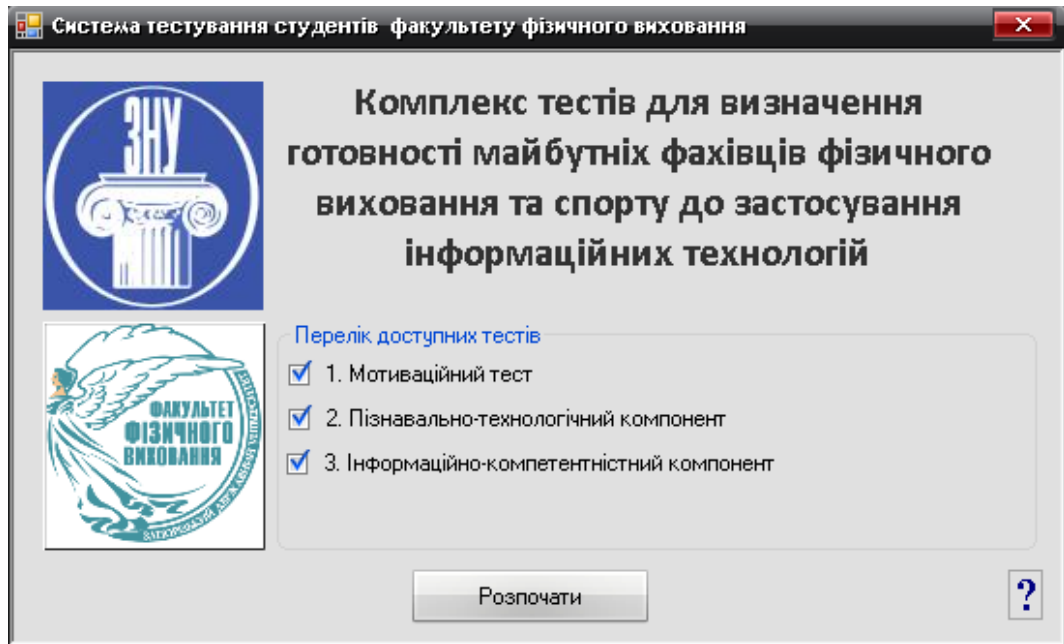


Рис. 2.1 Зовнішній вигляд (інтерфейс) програмного продукту

– можливості корекції вагових показників цільової функції (якщо це необхідно) (рис. 2.6);

– інформація про розробників методики педагогічного моніторингу професійно значущих якостей і програмного продукту (рис. 2.7);.

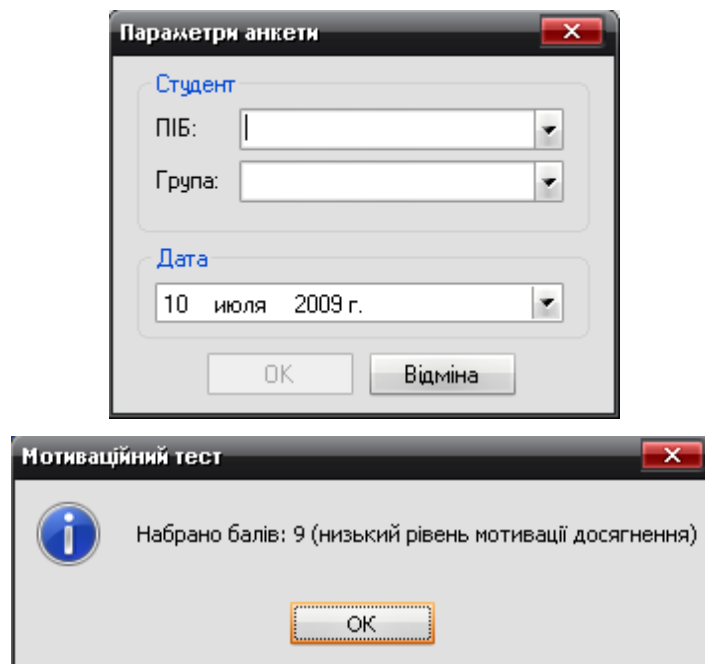


Рис. 2.2 Введення персональних даних і результатів тестування

Мотиваційний тест

1. На мою думку, успіх у житті залежить від випадку, а не від розрахунку.
 - ☐ так;
 - ☐ ні.
2. Якщо я залишусь без улюбленого заняття, моє життя стане марним.
 - ☐ так;
 - ☐ ні.
3. У будь-якій справі важливим для мене є її виконання, а не кінцевий результат.
 - ☐ так;
 - ☐ ні.
4. Вважаю, що люди страждають більше від невдач на роботі, чим від поганих стосунків з близькими людьми.
 - ☐ так;
 - ☐ ні.
5. На мою думку, більшість людей живе не близькими, а віддаленими цілями.
 - ☐ так;
 - ☐ ні.

Рис. 2.3 Анкетування

Система тестування студентів факультету фізичного виховання

Комплекс тестів для визначення готовності майбутніх фахівців фізичного виховання та спорту до застосування інформаційних технологій

Перелік доступних тестів

- ☒ Список студентів...
- ☒ Тести...
- ☒ 3. Інформаційно-компетентнісний

Розпочати

Результати тестування

- Редагувати
- Пошук...
- Експорт
- Статистична обробка результатів...
- Перерахувати результати...
- Параметри...
- Про програму...

Рис. 2.4 Можливості обробки і редагування результатів тестування

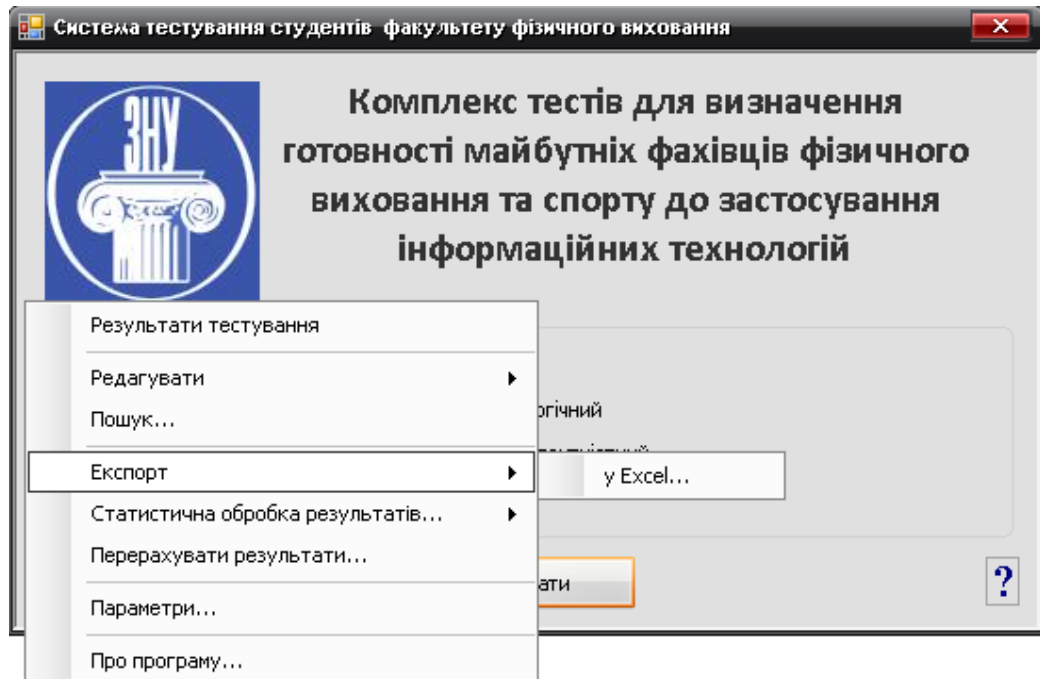


Рис. 2.5 Можливості експорту даних і статистичної обробки результатів тестування

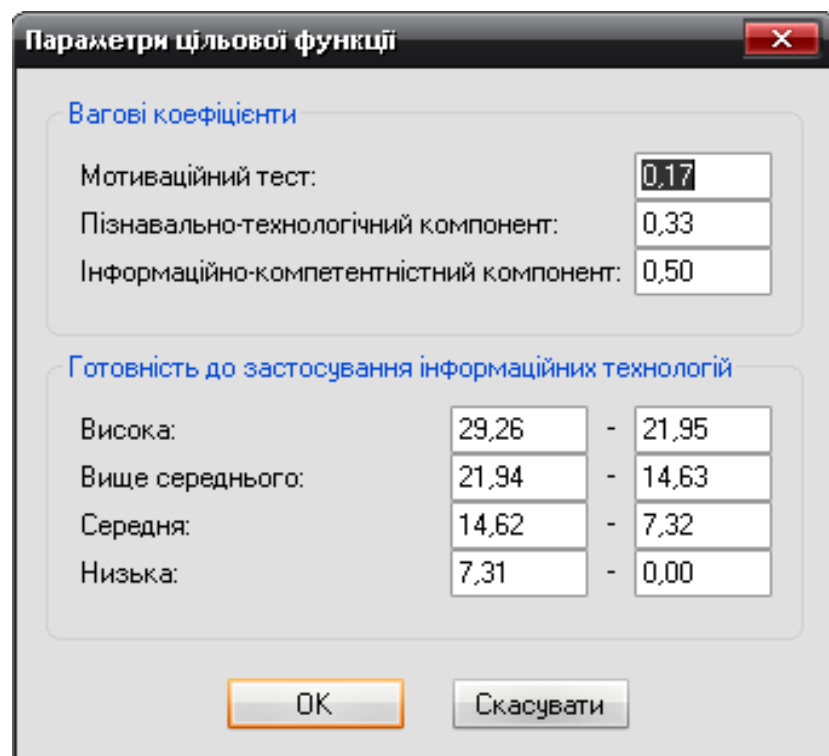


Рис. 2.6 Можливості корекції вагових показників цільової функції

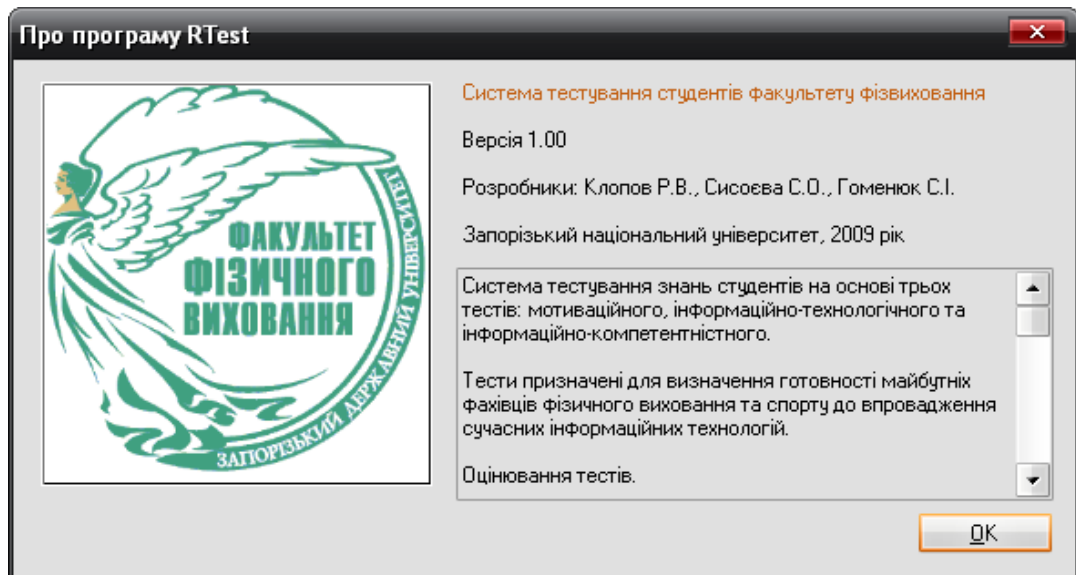


Рис. 2.7 Інформація про розробників методики педагогічного моніторингу професійно значущих якостей і програмного продукту.

Також з метою підвищення ефективності професійної підготовки майбутніх фахівців із застосуванням інформаційних технологій у вищих навчальних закладах та з урахуванням сучасного розвитку педагогічних програмних засобів розроблено і запропоновано комп'ютерну програму «R_local»: інформаційна система «База знань» для накопичення та систематизації електронних навчальних ресурсів вищого навчального закладу».

Перше вікно програми R_local. Ви маєте можливість вибирати наявну літературу, глосарій понять, довідники (рис. 2.8-2.16).

Призначення функціональних кнопок програми з ліва на право (див рис. 2.17):

- перехід до першого запису таблиці;
- перехід до попереднього запису таблиці;
- перехід до наступного запису таблиці;
- перехід до останнього запису таблиці;

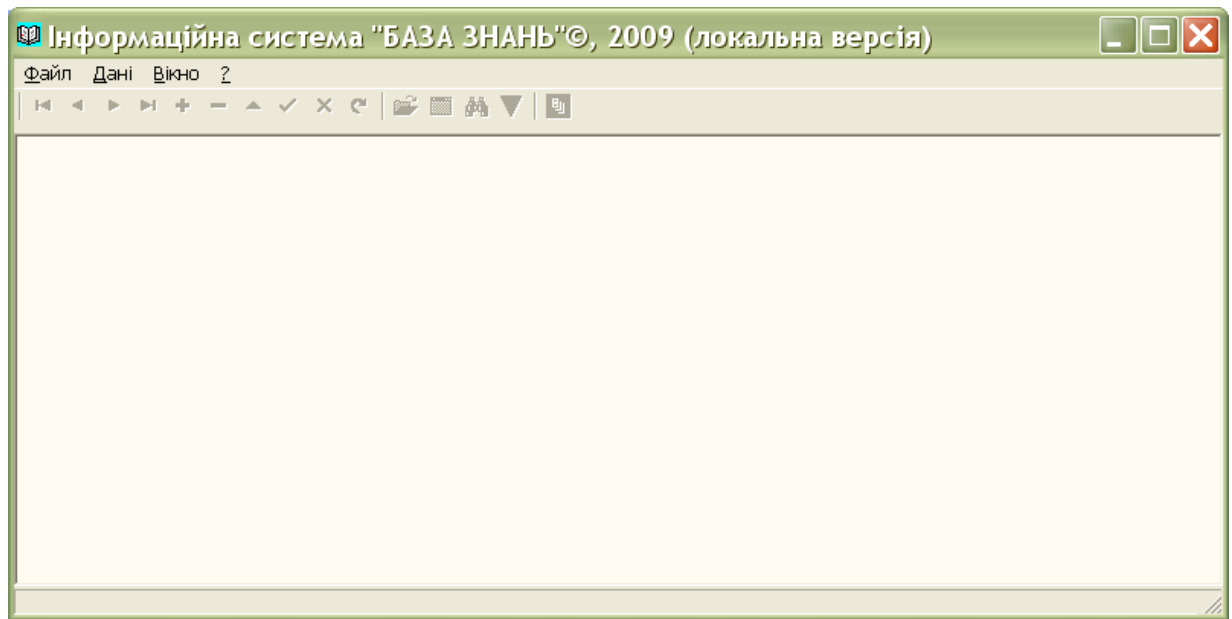


Рис. 2.8 Перше діалогове вікно програми

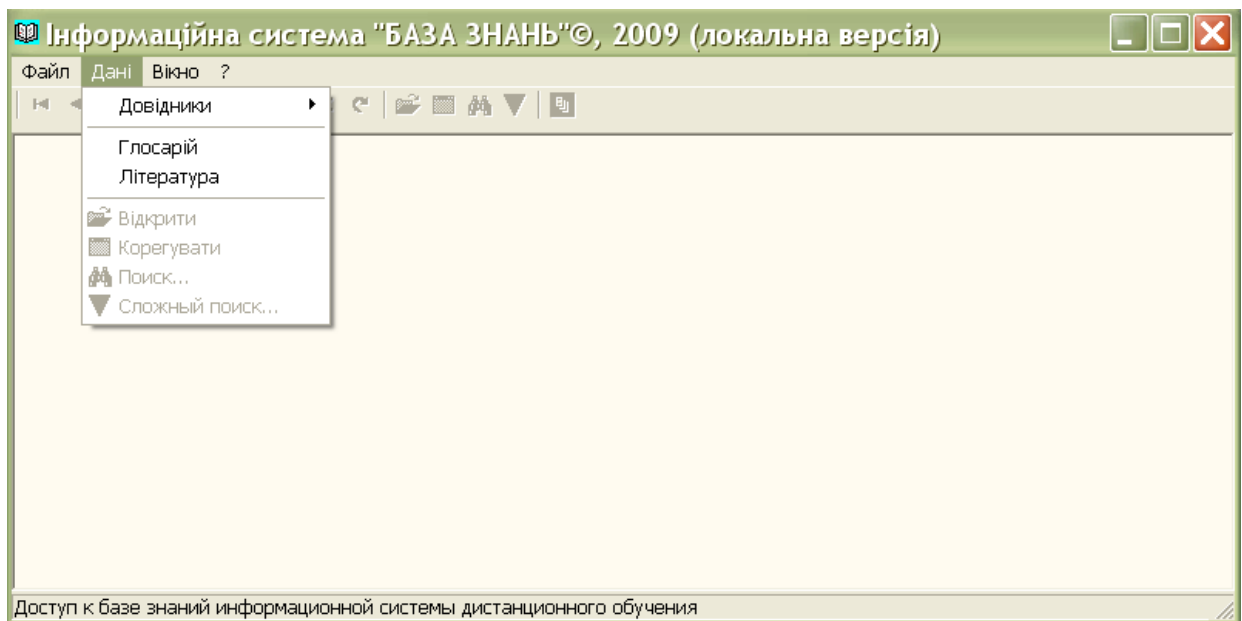


Рисунок 2.9 Зміст меню даних.

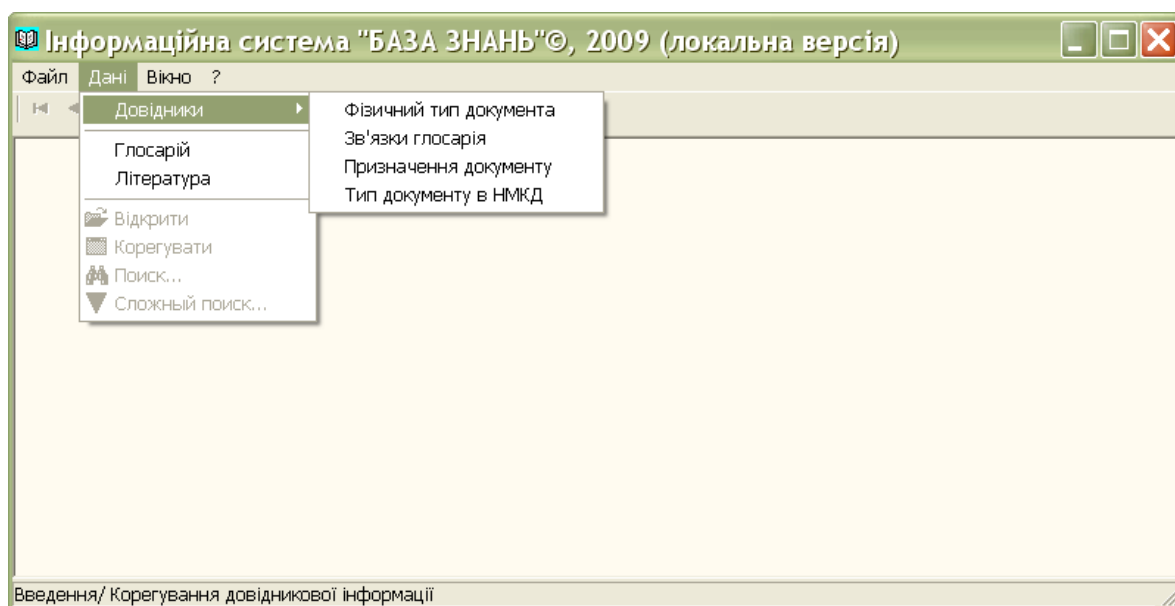


Рис. 2.10 Типи довідників

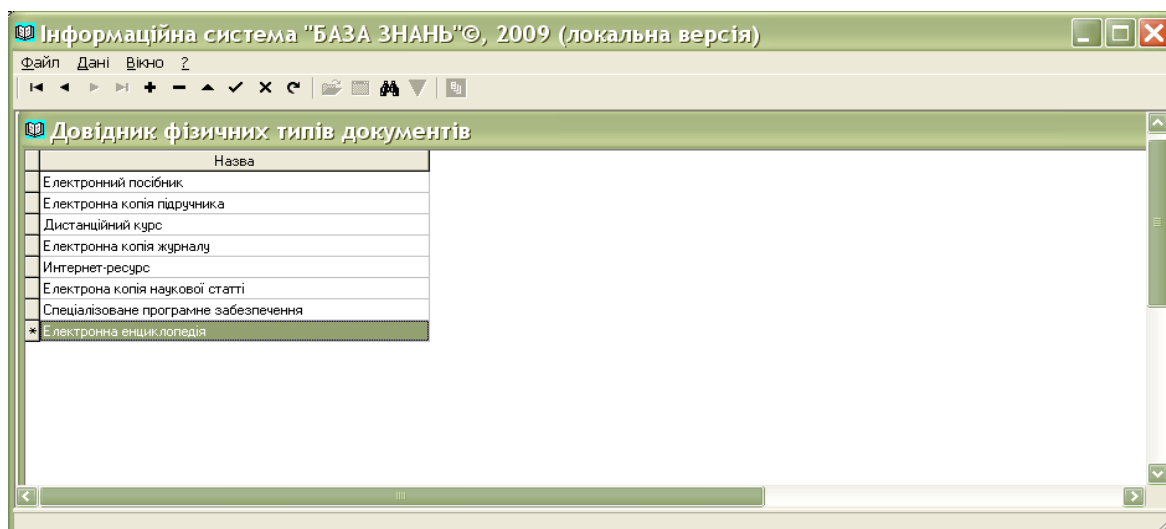


Рис. 2.11 Довідник фізичних типів електронних документів

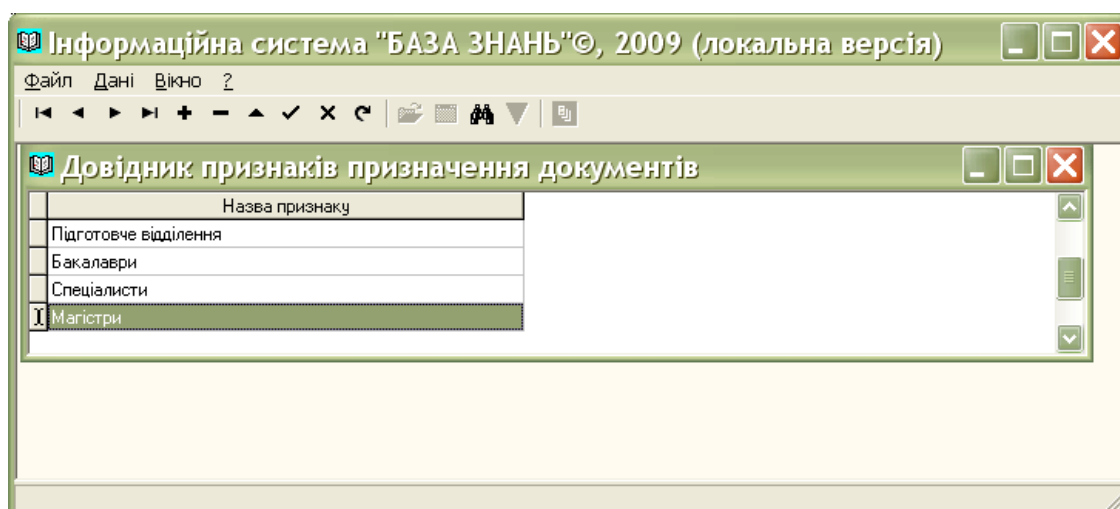


Рис. 2.12 Довідник ознак призначення електронних документів

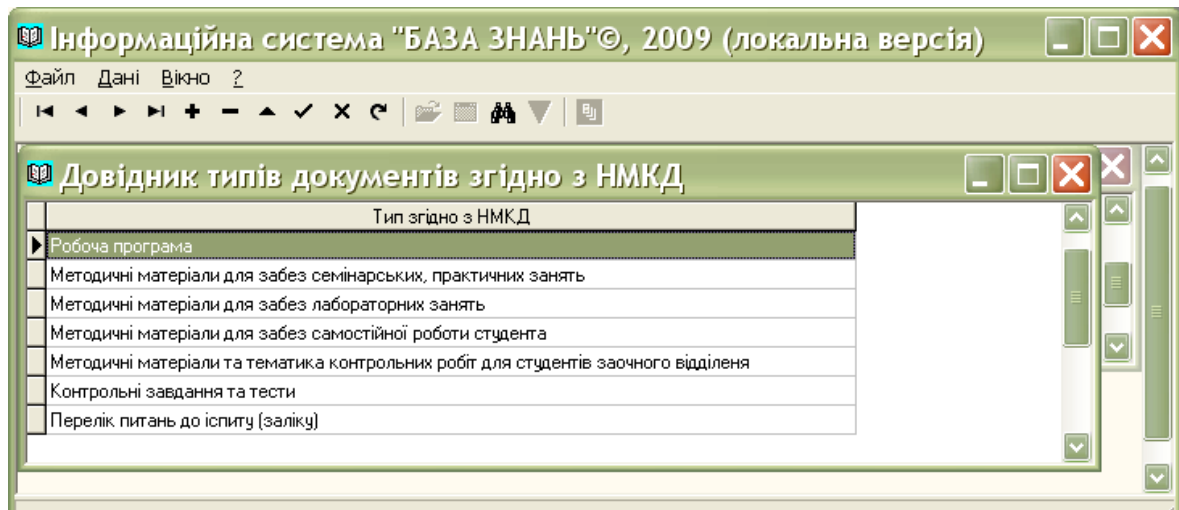


Рис. 2.13 Довідник типів документів згідно НМКД (НМКД – навально-методичний комплекс дисципліни)

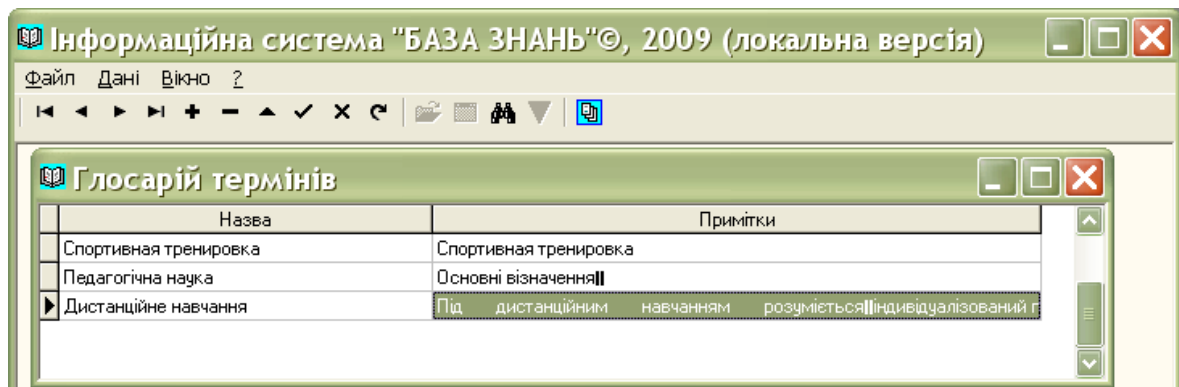


Рис. 2.14 Глосарій термінів та література зв'язана з цими поняттями

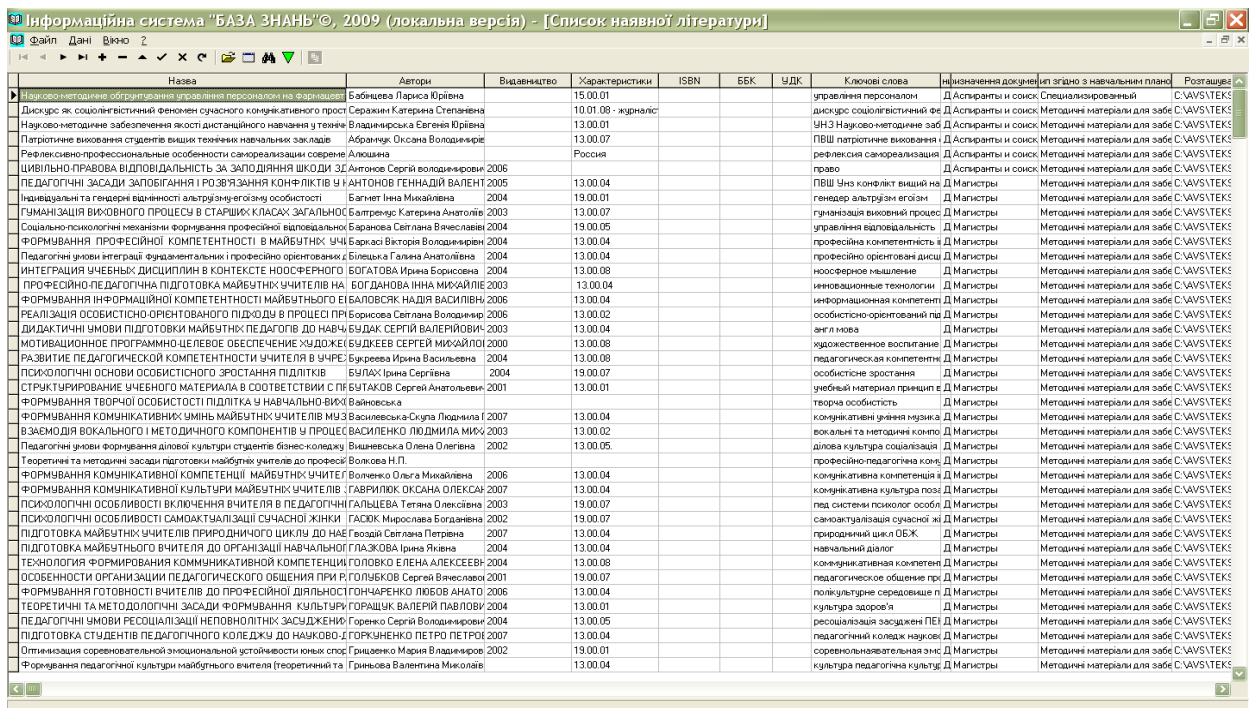


Рис. 2.15 Перелік наявної літератури у вигляді таблиці.

Введення/Коригування даних про літературне джерело

Назва: енности самореализации современного управленца как личностное измерение его профессиональной успешности

Автори: Алюшина

Вид-во:

Характеристики: Россия **ISBN:** **ББК:** **УДК:**

Ключові слова: рефлексия самореализация управленцы

Фізичний тип: Електронна копія наукової статті **Призначення:** Аспиранты и соискатели

Тип згідно з НМКД: Методичні матеріали для забез самостій **Розташування:** C:\AVS\TEKST\DISSERT\Алюши

Примітки: ☐ Навчання рухомим діям

Navigation buttons: Previous, Previous, Next, Next, +, -, Up, Check, Cancel, Refresh, and a folder icon.

Рис. 2.16 Вікно введення та коригування даних.

- додати новий запис у таблицю;
- видалити поточний запис;
- перейти в режим редагування поточного запису;
- зберегти поточні зміни;
- відмінити поточні зміни;
- оновити інформацію у таблиці;
- відкрити поточний електронний документ;
- введення та корегування даних про літературу у вигляді картки;
- пошук у поточному стовпці;
- параметричний пошук у таблиці літератури;
- перелік джерел, що зв'язані з поняттям у глосарії.

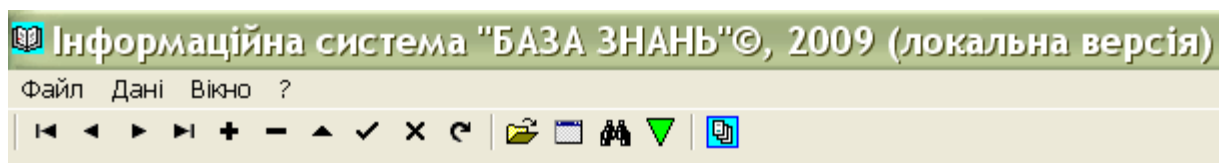


Рис. 2.17 Призначення функціональних кнопок.

Таким чином на основі теоретичного аналізу наукової педагогічної літератури, нашого дослідження і власного практичного досвіду одним з найважливіших компонентів електронних навчально-методичних комплексів дисципліни є програмне забезпечення, яке дозволить викладачу проводити моніторинг сформованості професійно значущих якостей майбутнього фахівця з фізичного виховання та спорту. З цією метою на основі розроблених нами методик створене комп'ютерне програмне забезпечення: моніторинг готовності майбутніх фахівців з фізичного виховання та спорту до застосування інформаційних технологій у професійній діяльності – програмно-тестувальний комплекс «RTEST», моніторинг формування готовності майбутніх фахівців фізичного виховання та спорту до впровадження технології спортивного менеджменту – програмно-тестувальний комплекс «БСК», і комп'ютерна програма «R_local»: інформаційна система «База знань» для накопичення та систематизації електронних навчальних ресурсів вищого навчального закладу. На прикладі розробленого комп'ютерного програмного забезпечення детально розглянуті етапи створення.

ГЛОСАРІЙ

База даних – сукупність інформаційних матеріалів, що зберігаються і систематизовані таким чином, щоб їх можна було б знайти й обробити за допомогою комп'ютера.

Байт – стандартна одиниця виміру інформації, яка утворена з 8 біт (може приймати 256 числових значень: від 0 до 255).

Біт – базова (мінімальна) одиниця виміру інформації, яка може приймати два значення – 0 або 1.

Гіпертекст – це електронний документ, який сформовано із застосуванням спеціалізованої мови розмітки і має посилання (гіперпосилання) на інші документи.

Глобальна комп'ютерна мережа – обчислювальна мережа, що не має чіткої територіальної прив'язки. Зазвичай це деяка сукупність взаємопов'язаних між собою локальних мереж.

Демодуляція – процес перетворення аналогового електричного сигналу у цифровий (зворотній до модуляції).

Джерело інформації – це деякий матеріальний об'єкт, що володіє певними відомостями (інформацією), які представляють конкретний інтерес для потенційних користувачів.

Документ (документована інформація) – це інформація, що зафіксована на матеріальному носії з реквізитами, які дозволяють її ідентифікувати.

Життєвий цикл програмного продукту – процес його побудови і розвитку. Складається з таких основних стадій: проектування, реалізація та експлуатація (супровід).

Інтерпретатор – програма, що виконує покроковий переклад тексту програми на деякій алгоритмічній мові в машинний код і тут же його виконує.

Інтернет – глобальна телекомунікаційна мережа інформаційних і обчислювальних ресурсів.

Інтранет – різновид Інтернет, що організований і працює в рамках окремої організації.

Інформаційна система – сукупність технічного, програмного й організаційного забезпечення, що призначена для своєчасного забезпечувати належних людей потрібною інформацією.

Інформація – певна сукупність відомостей про навколишній світ, події або діяльності.

Компілятор – програма, що здійснює переклад тексту програми на деякій мові програмування в машинний код.

Компонент інформаційної системи – це її частина, що виділена за зазначеною ознакою або сукупністю ознак і розглядається як одне ціле. Такі компоненти за своїм призначенням передусім поділяються на забезпечувальні і функціональні.

Комп'ютерна мережа – система зв'язку між комп'ютерами.

Комутатор – пристрій, що дозволяє з'єднувати декілька вузлів комп'ютерної мережі в межах одного сегмента.

Концентратор – мережевий пристрій, що дозволяє з'єднувати декілька сегментів комп'ютерної мережі в єдиний логічний сегмент.

Локальна комп'ютерна мережа – поєднання групи комп'ютерів, що обмежене певними територіальними рамками (кімната, поверх, будівля, група розташованих поруч будівель).

Маршрутизатор – мережевий пристрій, що відповідає за оптимальну з урахуванням топології мережі розсилку пакетів даних між різними сегментами комп'ютерної мережі.

Мережева комунікаційна система – фізичне середовище передачі інформації між комп'ютерами у мережі: електричні кабелі, оптичне волокно і таке інше.

Модем (модулятор-демодулятор) – пристрій, що здійснює перетворення даних з цифрової форми, в якій вони зберігаються в комп'ютері, в аналогову (електричні коливання), в якій вони можуть бути передані по телекомунікаційній мережі (і навпаки).

Модуляція – процес перетворення цифрового сигналу у аналогові електричні коливання.

Об'єктно-орієнтований аналіз – декомпозиція предметної області на деяку множину об'єктів, що взаємодіють між собою. Кожен об'єкт описується певною сукупністю атрибутів і методів (дій, які може виконувати в процесі свого функціонування). Об'єкти з однаковими наборами атрибутів і методів належать до одного класу.

Операційна система – сукупність програм, що управляє апаратними і програмними ресурсами комп'ютера, а також забезпечує виконання команд користувача.

Персональна інформаційна система – різновид інформаційної системи, призначеної для вирішення задач, пов'язаних з інформаційними потребами однієї людини.

Протокол (мережевий протокол обміну даними) – це набір угод, який визначає порядок обміну даними між різними програмами. Протоколи визначають засоби передачі повідомлень і обробки помилок в мережі, а також дозволяють розробляти стандарти, не прив'язані до конкретної апаратної платформи.

Робоча станція – елемент комп'ютерної мережі. Це робоче місце користувача, на якому він вирішує свої прикладні задачі при роботі в мережі.

Сервісне програмне забезпечення – сукупність програмних продуктів, що надають користувачам додаткові послуги в роботі з комп'ютером і розширюють можливості ОС.

Система автоматизації документообігу – організаційно-технічна система електронного документообігу, що забезпечує процес створення, управління доступом і розповсюдження електронних документів у комп'ютерних мережах, а також забезпечує контроль над потоками документів в організації.

Система кодування – сукупність методів і правил кодування.

Система управління базами даних – спеціалізована програма, що призначена для організації і ведення баз даних.

Структура інформаційної системи – склад утворюючих її підсистем (компонентів).

Структура коду – умовне позначення складу й послідовності розташування знаків у коді.

Структурний аналіз – методологія дослідження і аналізу предметної області, що полягає в декомпозиції (розбитті) складної системи на більш прості підсистеми й зв'язки між ними.

Сховище даних – централізована база даних, що об'єднує інформацію з різнорідних джерел, систем і баз даних та надає зібрану інформацію програмам кінцевих користувачів.

Технічні носії інформації – матеріальні об'єкти в яких інформація знаходять своє відображення у вигляді символів, образів, сигналів, технічних рішень і процесів.

Топологія мережі – схема з'єднання каналами зв'язку комп'ютерів між собою в мережі передачі даних.

Транзакція – група послідовних операцій, яка являє собою логічну одиницю роботи з даними. Транзакція може бути виконана або цілком і успішно (при цьому зберігається цілісність даних), або не виконана взагалі і тоді вона не повинна призвести ніякого ефекту.

Транслятор – програма, що здійснює переклад тексту програми з однієї мови програмування в іншу.

Трафік – обсяг інформації, що проходить комп'ютерною або телекомунікаційною мережею за одиницю часу.

Шлюз (мережевий шлюз) – апаратне або програмне забезпечення для з'єднання між собою комп'ютерних мереж із різними протоколами обміну інформації.

ASCII – американський стандартний код для обміну інформацією. Це восьмибітова таблиця кодування для представлення десяткових цифр, латинського та національного алфавітів, знаків пунктуації і управляючих символів при передачі інформації в мережах телекомунікації.

HTML – гіпертекстова мова розмітки, що застосовується для форматування (оформлення) Web-сторінок.

URL – універсальний локатор ресурсу, стандартний засіб написання адрес ресурсів в мережі Інтернет.

Web-документ – гіпертекстовий документ, мінімальна логічна одиниця інформації в мережі Інтернет.

Web-канал – це Web-сайт, що здатний самостійно ініціювати постачання інформації клієнту, який підписаний на відповідну інформаційну послугу.

Web-сайт – група логічно (тематично) пов'язаних між собою Web-документів, що зберігаються на одному Web-сервері.

Web-сервер – файловий сервер, що призначений для зберігання окремих Web-документів або Web-сайтів.

Web-сторінка – теж саме, що й Web-документ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ковалюк Т.В. Основи програмування / Т.В. Ковалюк. – К.: ВНУ Києв, 2005. – 400 с.
2. Міщенко О.А. Сутність мультимедійних технологій навчання / О.А. Міщенко // Педагогіка, психологія та медико-біологічні проблеми фізичного виховання і спорту: наукова монографія за ред. проф. С.С. Єрмакова – Харків: ХДАДМ (ХХІІ), 2006. – №3. – С. 69-71.
3. Гуревич Р.С. Інформаційно-телекомунікаційні технології в навчальному процесі та наукових дослідженнях: навчальний посібник для студентів педагогічних ВНЗ і слухачів інститутів післядипломної освіти / Р.С. Гуревич, М.Ю. Кадемія. – Вінниця: Планер, 2005. – 336 с.
4. Комп'ютерна програма «БСК: моніторинг формування готовності майбутніх фахівців фізичного виховання та спорту до впровадження технологій спортивного менеджменту» / Л.В.Безкоровайна, Л.П.Сущенко, Р.В.Клопов. – свідоцтво про реєстрацію авторського права на твір №27055 від 25.12.2008 р., видане Державним департаментом інтелектуальної власності Міністерства освіти і науки України.
5. Комп'ютерна програма «Програмно-тестувальний комплекс «R_local: інформаційна система «База знань» для накопичення та систематизації електронних навчальних ресурсів вищого навчального закладу» / Р.В.Клопов, С.О.Сисоєва, С.І.Гоменюк. – свідоцтво про реєстрацію авторського права на твір №32494 від 22.03.2010 р., видане Державним департаментом інтелектуальної власності Міністерства освіти і науки України.
6. Комп'ютерна програма «Програмно-тестувальний комплекс «RTEST: моніторинг формування готовності майбутніх фахівців фізичного виховання та спорту до застосування інформаційних технологій» / Р.В.Клопов, С.О.Сисоєва, С.І.Гоменюк. – свідоцтво про реєстрацію авторського права на твір №30385 від 21.09.2009 р., видане Державним

департаментом інтелектуальної власності Міністерства освіти і науки України.

7. Козина Ж.Л. Эффективность начального обучения технике гандбола на основе применения современных информационных технологий / Ж.Л. Козина // Физическое воспитание студентов творческих специальностей: Сб. научн. тр. под ред. Ермакова С.С. – Харьков: ХГАДИ (ХХПИ), 2004. – №4. – С.22-28.

8. Ротерс Т.Т. Мультимедійні технології в процесі підготовки спеціалістів з фізичного виховання / Т.Т. Ротерс // Молода спортивна наука України: Зб. наук. праць з галузі фізичної культури та спорту. – Львів: НФВ «Українські технології», 2001. – С. 269-273.

9. Денисова Л.В. Організація процесу підготовки спеціалістів з оздоровчої фізичної культури із застосуванням гіпермедійного навчального середовища / Л.В. Денисова // Теорія і методика фізичного виховання і спорту. – 2005. – № 2-3. – С. 114-116.

10. Harnessing Technology Review 2007: Progress and impact of technology in education: Summary report – 20 p. [Електронний ресурс] – Режим доступу: <http://publications.becta.org.uk/download.cfm?resID=33978>

11. Карчевская Н.В., Хомякова А.Р. Дидактические возможности мультимедийных презентаций в подготовке студентов инженерно-педагогических специальностей / Н.В. Карчевская, А.Р. Хомякова. [Електронний ресурс] – Режим доступу: <http://intkonf.org/category/arhiv/ukrayinska-nauka-v-merezhi-internet/pedagogika/> (06.10.09 р.)

12. <http://www.microsoft.com/express>

13. <http://www.embarcadero.com/products/delphi>

14. <http://qt.nokia.com>

Методичне видання
(українською мовою)

Клопов Роман Вікторович

**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У ФІЗИЧНОМУ
ВИХОВАННІ ТА СПОРТІ:
ОСОБЛИВОСТІ РОЗРОБКИ СПЕЦІАЛІЗОВАНОГО
КОМП'ЮТЕРНОГО ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**

Рецензент: к.п.н., доц. Сватъєв А.В.
Відповідальний за випуск: Клопов Р.В.
Коректор: Андрєєва О.С.

Підписано до друку 2.09.2011. Формат 60 × 90/16.
Наклад 40 прим.

Запорізький національний університет
69600, м. Запоріжжя, МСП-41
вул. Жуковського, 66

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
ДК № 2952 від 30.08.2007