

**Міністерство освіти і науки України**  
**Головне управління освіти і науки Полтавської облдержадміністрації**  
**Полтавське територіальне відділення МАН України**  
**Кременчуцьке районне наукове товариство учнів «Мала академія наук»**

**Секція:** інформатика та  
обчислювальна техніка

**Підсекція:** мультимедійні системи,  
навчальні та ігрові програми

## **РОЗРОБКА КОМПЛЕКСУ ПРОГРАМ ПЕДАГОГІЧНОГО КОНТРОЛЮ ЗНАНЬ**

**Роботу виконав:**

Зайкін Андрій Вікторович,  
учень 11 класу  
Кременчуцької ЗОШ І-ІІІ ступенів  
№ 17 «Вибір» імені М.Г. Неленя

**Науковий керівник:**

Лагута Світлана Миколаївна,  
вчитель інформатики  
Кременчуцької ЗОШ І-ІІІ ступенів  
№ 17 «Вибір» імені М.Г. Неленя,  
спеціаліст вищої категорії

**Науковий консультант:**

Лисенко Тетяна Іванівна,  
викладач інформатики  
Кременчуцького педагогічного  
училища ім. А.С. Макаренка,  
Заслужений вчитель України

**Полтавське територіальне відділення МАН України**  
**Кременчуцьке районне наукове товариство учнів «Мала академія наук»**

**ТЕЗИ**

до науково-дослідницької роботи

**«Розробка комплексу програм педагогічного контролю знань»**

учня 11 класу ЗОШ №17 “Вибір” імені М.Г. Неленя

Зайкіна Андрія Вікторовича

Науковий керівник: вчитель інформатики, Лагута Світлана Миколаївна

Особливості сучасного шкільного навчання – використання інформаційно-комунікаційних технологій на уроках з різних предметів. Тому стає проблема забезпечення вчителів сучасними засобами ведення уроку і контролю знань з використанням комп’ютера. Використання сучасних програм дає можливість творчої діяльності як вчителя, так і учня.

Мета дослідження - визначення вимоги до сучасних систем тестування, що можуть оптимізувати процес контролю знань та розробити власну програму з урахуванням вироблених критеріїв.

Основні складові системи:

1. Програма для вчителя, яка аналізує базу даних з запитаннями та варіантами відповідей на них і виводить результат на монітор для демонстрації;
2. Програма для учня, яка аналізує базу даних і виводить на монітор питання з варіантами відповідей у довільному порядку, на які учень відповідає, обравши один або кілька варіантів відповідей за обмежений час або вводить свій варіант відповіді.

Розглянувши можливості, що надають користувачам системи тестування, такі як УТК (універсальний тестовий комплекс), Асистент 2, OpenTEST та інші, були визначені такі вимоги до ефективного педагогічного засобу контролю:

- валідність;

- однозначність;
- надійність;

В результаті виконання практичної частини дослідження була створена програма, яка реалізує процес комп'ютерного тестування. До програмного комплексу входить база даних тестів з таблицями:

- marksystem – таблиця рівнів оцінювання;
- questions – таблиця завдань;
- results – таблиця зберігання оцінок та відсоткового рівня знань;
- tests – набір тестів за темами.

Створений комплекс може стати ефективним засобом контролю якості знань, активізує пізнавальну і дослідницьку діяльність учнів, включає тестування, що допомагає вчителю провести сучасний, нестандартний урок з використанням інформаційно-комунікаційних технологій.

Використання методичного комплексу: для проведення тестування або у визначенні реального рівня знань учнів. Метод апробується у старших класах інформаційно-технологічного профілю у ЗОШ № 17 м. Кременчука.

## ЗМІСТ

Вступ .....	5
Розділ 1. Тестування як засіб контролю знань .....	6
1.1 Особливості програм педагогічного контролю.....	6
1.2 Аналіз умов використання тестових завдань.....	8
1.3 Загальні вимоги до створення тестів.....	11
Розділ 2. Практична розробка програмного засобу .....	13
2.1 Особливості програми .....	13
2.2 Структура програми та реалізація основних завдань.....	16
2.3 Результати практичного використання.....	20
Висновки.....	21
Список використаних джерел.....	23
Додатки .....	24

## ВСТУП

Найбільш проблемно-актуальним на етапі перебудови системи освіти в Україні виступають дослідження в галузі педагогічного контролю, зокрема, вимірювання рівня навченості учнів і студентів та оцінювання якості навчального процесу.

Вступ України до міжнародних інституцій та приєднання до Болонського процесу потребує якісних змін у методах підготовки майбутніх фахівців та оцінці якості такої підготовки. Одним із таких методів виступає тестування, яке дає змогу оцінити рівень засвоєння навчального матеріалу учнями (студентами) й виявити прогалини у їхніх знаннях. [1]

Тестування дають змогу одержувати однозначні відповіді або ж відповіді в межах припустимої ймовірності. Природно, що механізми тестувань, пов'язані із суспільством, істотно відрізняються від інших напрямів своєю складністю, а тому потребують глибшого вивчення.

**Об'єкт дослідження** – використання системи тестів як засобу для контролю знань.

**Предмет дослідження** – вимоги до організації тестування як інструменту оцінки якості знань.

**Мета дослідження** – визначити різні підходи до побудови тестових програм та вимоги до сучасних систем тестування, що можуть оптимізувати процес контролю знань та розробити власну програму з урахуванням вироблених критеріїв.

**Завдання дослідження:**

1. Визначити основні структурні складові систем тестування та їх функції;
2. Проаналізувати наявні системи контролю знань та визначити критерії, яким має задовольняти досконала тестова оболонка;
3. На основі визначених критеріїв розробити власну програму для педагогічного засобу контролю знань.

Практичне значення роботи – застосування на уроках «Основи інформаційних технологій» в Кременчуцькій загальноосвітній школі № 17 імені М.Г. Неленя.

## РОЗДІЛ 1

### Тестування як засіб контролю знань

#### 1.1 Особливості програм педагогічного контролю

Педагогічний контроль являє систему науково обґрунтованої перевірки результатів освіти, навчання і виховання. Стосовно процесу навчання, контроль означає виявлення, вимірювання і оцінювання знань, умінь та навичок і є взаємопов'язаною і взаємозумовленою діяльністю викладача і студента, учителя і учня. [1]

Зазвичай тест складається з двох компонентів: завдання і еталона відповіді. Завдання створюється відповідно до вимог валідності, однозначності і простоти з урахуванням цілей вивчення предмета. Еталон створюється на основі послідовного опису правильного і повного виконання завдання з виділенням його сутнісних операцій. Операція тесту є одиничною дією, яка відображає засвоєння учнем знань, умінь і навичок кожного навчального елемента, що є метою навчання для даного навчального предмета. [2]

Класифікація тестів може бути аналогічною класифікації рівнів засвоєння і характеризуватися способом виконання потрібних дій.

**Тести I рівня** передбачають перевірку вмінь тих, хто проходить випробовування за допомогою підказки. Вони вимагають розпізнавання, виявлення відмінностей встановлення класифікаційної належності.

**Тести II рівня** передбачають виявлення умінь самостійно відтворювати за допомогою пам'яті раніше засвоєну діяльність в алгоритмічній формі. До них належать:

- тести, що вимагають підстановки;
- конструктивні тести, тобто ті, що вимагають самостійного відтворення відповіді за допомогою пам'яті ;
- тести – типові задачі, виконання яких вимагає використання засвоєних алгоритмів діяльності.

**Тести III рівня** передбачають виявлення готовності того, хто проходить випробовування до продуктивних дій евристичного типу, тобто виконання нетипових завдань і ситуацій за допомогою самостійної трансформації засвоєних раніше правил і дій.

**Тести IV рівня** передбачають виявлення творчих умінь, тобто дослідницьких можливостей учня щодо одержання ним нової інформації. Це тести-проблеми, алгоритми розв'язання яких невідомі. [1]

До організації тестування ставляться певні вимоги [2]:

- можливість швидкої розробки матеріалу для тестування, що охоплює не тільки ключові моменти змісту освіти, а й певні його нюанси;
- невелика тривалість сеансу тестування;
- забезпечення конфіденційності;
- наявність засобів швидкої перевірки й підведення підсумків (виставлення певної оцінки).

На практиці всі вказані пункти пов'язані й залежать один від одного. Зокрема, перевірка результатів тестування залежить від змістовного наповнення тесту, а тривалість тестування повинна бути збалансована з рівнем складності завдань.

У кожному конкретному випадку у викладача через специфіку дисциплін, навчального закладу, програми або інших чинників з'являються свої вимоги до системи тестування. Тому важливо створити гнучке й певною мірою універсальне середовище. Логічно, що найбільш придатною платформою й основою виявляються комп'ютерні системи, які, зазвичай, включають три складові:

- апаратну, що має відповідати поточним потребам або мінімальним вимогам;
- програмну, за допомогою якої реалізується тестова система;
- людську, що займається координацією усього процесу тестування, створенням і керуванням системою. [3, 6]

Отже, можна визначити основні складові та функції системи контролю знань:

- модуль призначений для створення і редагування навчальних курсів, реєстрації викладачів, учбових груп. Користувачами модуля є адміністратори і викладачі;
  - модуль проходження навчальних курсів і реєстрації учнів. Користувачами модуля є учні;
  - модуль контролю в режимі реального часу поточних результатів тестування. Користувачами є тільки викладачі;
  - модуль одержання статистичних звітів за результатами тестування. Користувачами модуля є адміністратори і викладачі;
- Усі модулі повинні бути зв'язані між собою у деякій базі даних.

## **1.2 Аналіз умов використання тестових завдань**

Навчальний план на певному етапі вимагає точно визначати реальний рівень знань учнів. Деякі недоліки теперішньої системи тестування не дозволяють зробити це. Наприклад:

- більшість програм для тестування мають визначену кількість питань, яку не можна змінити без втручання в тіло програми, тобто без роботи програміста;
- кількість балів, яка надається за правильну відповідь також є незмінною;
- однакова послідовність виведення тестових завдань на екран монітора збільшує ймовірність списування;
- незмінність завдань призводить до того, що певна програма може використовуватися для контролю знань лише з одної базової дисципліни;
- усі тестування за допомогою комп'ютера проводяться за однією схемою, а це не дає можливості вчителям самостійно модернізувати навчальний процес.

Визначення критеріїв, за допомогою яких можна позбутися основних недоліків комп'ютерного тестування, допоможе більш точно наблизитися до досягнення мети роботи.



Сучасна програма для проведення тестування за допомогою комп'ютера повинна задовольняти такі критерії [4, 5, 10]:

- кількість питань у тесті має визначати сам учитель, чи та людина, яка проводить тестування;
- кількість балів, що надається за правильну відповідь також має вказувати вчитель;
- процес зміни запитань і варіантів відповідей має бути простим, легким та доступним для виконання навіть для тих, хто має лише основні невеликі навички в роботі з комп'ютером;
- сам процес створення тесту та його проведення повинен бути простим і зручним як для вчителя, так і для учнів;
- аналізування виконаної роботи також повинно проводитися досить швидко та легко.

Все програмне забезпечення можна розділити на дві великі групи: безкоштовне та те, за яке потрібно заплатити певну суму. Отримати кошти для купівлі програми у школі не завжди можна. Тому я розгляну кілька програм для тестування з серії безкоштовних.

Для порівняння візьмемо три програми що мають «гриф» безкоштовні.

**1.** УТК (Універсальний тестовий комплекс) – програма для проведення тестування як на локальній машині так і по локальній мережі. Адреса в Інтернеті <http://utk.mastak.ru/>.

**2.** Асистент 2 – програма для проведення контролю знань студентів. Адреса в Інтернеті <http://theosoft.virtuale.net>

**3.** OpenTEST – оболонка для проведення інтерактивного тестування в вікні браузера. Адреса в Інтернет [www.opentest.com.ua](http://www.opentest.com.ua)

Розглянемо детальніше кожну з вищезгаданих програм.

УТК (Універсальний тестовий комплекс) - створена для проведення тестування студентів в комп'ютерному класі. Сама програма складається з двох частин – генератора та тестера. Генератор використовується для створення тестів та налагодження тестування в класі, а тестер – модуль для локального тестування. Для

використання генератора в повній мірі потрібно зареєструвати програму в Інтернеті (реєстрація безкоштовна). Нажаль всі версії програми після 1.51 вже платні. Програма тестер проводить тестування та видає відповіді учню що проходить тестування. Недоліком є те, що варіанти відповіді не перемішуються. Результати тестування мають красивий та зрозумілий вигляд, зберігаються в файлі, їх досить легко можна переглянути та роздрукувати [7].

Ще один недолік – генератор працює тільки на одному комп'ютері і тестування по локальній мережі можна провести тільки при включеному та налагодженому генераторі тестів.

Асистент2 - програма має легку форму створення тестів, використовуючи простий текстовий редактор – блокнот. Створені тести легко можна змінити, але суттєвим недоліком програми є те, що не підтримуються графічні об'єкти та під час припинення тестування можна отримати 12 балів відповівши тільки на одне перше питання. Сервер статистики показує досить непогану статистичну інформацію загального характеру, яка також зберігається у вигляді текстового файлу [8].

Інсталяційний пакет OpenTest включає в себе сервер Apache, бази даних MySQL та набір модулів написаних на PHP. Після інсталяції на комп'ютері створюється налагоджений WEB-сервер з підтримкою MySQL та PHP. Дана технологія з використанням відкритого коду написання програми та орієнтація на Інтернет-технології дають можливість встановити тільки на сервері програму для тестування, а на локальних машинах ніяких програм встановлювати непотрібно [9].

Інтерфейс програми складається з 5 великих модулів: тестування - для проведення тестування, статистика – для виведення різноманітної статистичної інформації, студія тестів – для створення та редагування тестів, керування користувачами – для контролю та керування користувачами, зона адміністрування – для керування тестовими процесами в режимі on-line. Результати дослідження розглянуті у таблиці 1.1.

### Порівняльна характеристика тестових програм

Характеристика	УТК	Асистент 2	OpenTest
Зовнішній вигляд.	★★★★☆	★★★★☆	★★★★☆
Необхідне програмне забезпечення.	Генератор тестів, тестер на кожну машину.	Програма для тестування та тести на кожну машину.	Інсталяція на сервер.
Програма для створення тестів.	Є	Блокнот чи аналог.	TestEditor, або віртуальний модуль.
Статистика.	★★★★☆	★★★★☆	★★★★☆
Тестування по локальній мережі.	Можливе при включеному генераторі.	Можливе.	Завжди при включеному сервері.
Зміна коду програми	Неможлива.	Неможлива.	Можлива PHP.
Документація.	★★★★☆	★★★★☆	★★★★☆

### 1.3 Загальні вимоги до створення тестів

До організації тестування ставляться певні вимоги [1,6]:

#### 1. Валідність:

- а. Змістова – відповідність завдання змісту і обсягу сформованої в учнів орієнтовної основи діяльності, яка визначається правилами і методами виконання діяльності;
- б. Функціональна – відповідність вимог до дій того, хто проходить випробування, рівню його засвоєння;

#### 2. Однозначність:

- а. Змістова – всі учні повинні однаково розуміти завдання, щоб у процесі його виконання діяти відповідно до еталону;

- б. Функціональна – завдання повинно містити тільки одну задачу, яка відповідає певному рівню засвоєння;

**3. Надійність:**

- а. Тестового випробовування – забезпечується сукупністю тестів даного рівня з визначеною кількістю сутнісних операцій в них;
- б. Контролю якості освіти і навчання – забезпечується одержанням статистично вірогідного числа відповідей тих, хто проходить випробовування, залежно від обсягу генеральної сукупності.

З урахуванням перелічених вимог формуємо програмні модулі.

## РОЗДІЛ 2

### Практична розробка програмного засобу

#### 2.1 Особливості програми

Учню пропонується пройти тестування з певних тем. Оцінювання виконується на рівні окремої теми шляхом виставлення певного підсумкового бала.

Кожний тематичний тест складається з тестових завдань. Тестове завдання – це неподільна тестова одиниця, що включає в себе:

- питання;
- відповіді на нього з інформацією про правильність;
- тип інтерфейсу;
- використовуване джерело даних (база даних, генератор);
- час, який відводиться для відповіді;
- рівень складності тестового завдання;
- вагу питання;
- внутрішню групу, до якої належить тестове завдання;
- додаткові параметри.

Щоб учневі не задавалися питання, на які він уже відповідав у попередніх сеансах тестування при наявності нових незнайомих запитань, у програмі реалізовано спеціальний статистичний механізм відбору. За цим механізмом послідовність запитань, які потрібно задати учневі, формується у довільному порядку. Оцінювання результатів сеансу тестування й виставлення підсумкового бала реалізується за такими правилами: приймається лінійний розподіл шкали оцінок пропорційно кількості правильних відповідей (0-12 балів). Підсумкова оцінка обчислюється за формулою [5]:

$$Z = \frac{\sum_{k=0}^n Z_k}{n},$$

де  $Z$  – підсумкова оцінка;  $Z_k$  – оцінка за  $k$ -те тестове завдання;  $n$  – кількість заданих тестових завдань. У разі правильної відповіді на запитання  $Z_k=12$ , інакше -  $Z_k=0$ . Оскільки в сеансі тестування різні учні потребують різного часу для відповіді, цей час також враховується для виставлення балу. Максимальна можлива тривалість надання відповіді задається окремо для кожного тестового завдання.

Графік залежності оцінки за тестове завдання від тривалості його опрацювання наведено на рис. 2.1 [2]

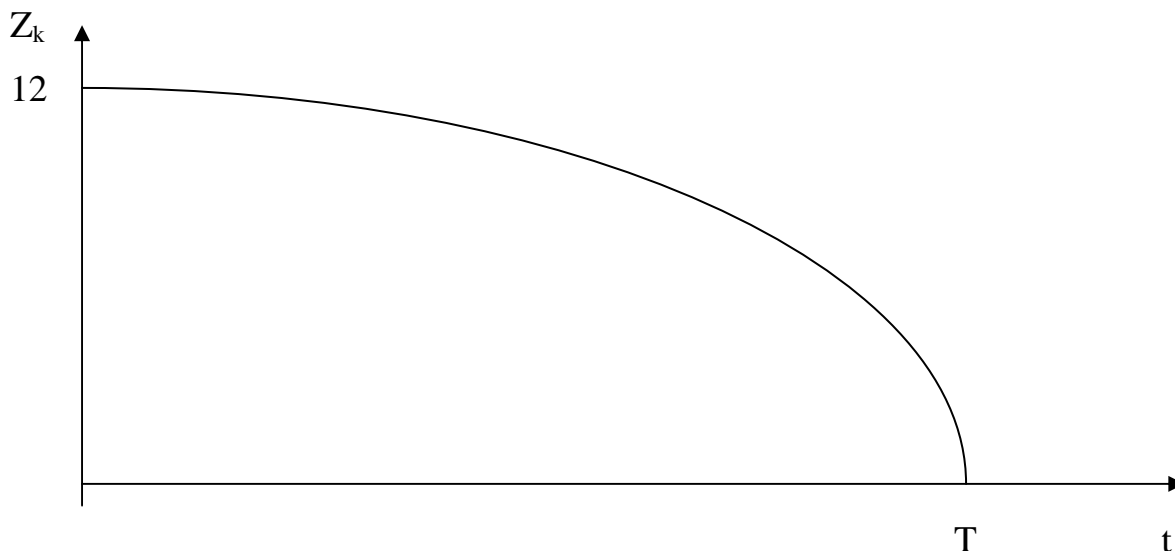


Рис. 2.1 Графік залежності оцінки від тривалості відповіді

Існує певна ймовірність угадування учнем правильних відповідей. Таку ймовірність угадування можна врахувати тільки під час виставлення підсумкової оцінки. Для кожного тестового завдання підраховується ймовірність угадування, що залежить від типу тесту [5]:

- при виборі однієї правильної відповіді з кількох імовірність угадування конкретного тестового завдання:

$$P_k = \frac{1}{m},$$

де  $P_k$  – імовірність угадування;  $m$  – кількість відповідей для даного завдання;

- при виборі кількох правильних відповідей з кількох наданих:

$$P_k = \frac{1}{2^m},$$

- у відкритому тесті  $P_k = 0$ .

Потім обчислюється сумарна ймовірність угадування. Тоді бали від 0 до 12 розподіляються у межах шкали оцінок рівномірно.


Маючи готовий проект, можна точно визначити його переваги та відмінності від інших програм:

- програма складається з кількох модулів, призначених спеціально для вчителя, адміністратора та учня;
- запитання та відповіді на них створює сам вчитель, отже, за допомогою цієї програми може проводитися тестування з будь-якої дисципліни;
- учитель самостійно встановлює кількість балів, яка надається за правильну відповідь, для кожного з запитань тесту, а це означає, що тестування буде більш гнучким і відповідатиме сучасному розділу учнів за ступенем знань на чотири основних рівні: незадовільний, задовільний, достатній та високий;
- для запису питань та варіантів відповідей тесту вчитель може використовувати зручний інтерфейс роботи з базою даних;
- інтерфейс програми досить простий і вікно для роботи містить у собі незначну кількість кнопок . Усі кнопки підписані, щоб учитель і учень могли швидко зорієнтуватися в програмі. А це створює найбільш зручні умови праці для обох сторін;
- після закінчення тестування програма виводить результати на монітор і показує, яку кількість балів із можливої набрав учень. Завдяки цьому процес виставлення оцінки за роботу є відкритим. Це, в свою чергу, виключає можливість впливу на підсумковий результат суб'єктивного ставлення вчителя до учня;
- гнучкість створення та проведення тестів за допомогою нової програми дозволяє використовувати її в будь-яких сферах життя, де потрібен контроль над рівнем знань людей.

## 2.2 Структура програми та реалізація основних завдань

Для розробки програми використовувалося середовище програмування Borland Delphi 7.0 під керуванням Windows XP. Дане середовище програмування дозволяє створювати програми, що складаються з необмеженої кількості модулів, підтримує всі сучасні технології в області програмування, відповідає принципам об'єктно-орієнтованого програмування. Створення програм у середовищі операційної системи Windows XP забезпечує зручний інтерфейс користувача, сумісність практично з усіма видами периферійних пристроїв, обмін даними між різними додатками операційної системи.

Створений програмний комплекс складається з модулів:

- 
  - Question.pas - опис класу запитання;
- 
  - ResultsForm.pas - модуль форми обробки результатів;
- 
  - SettingsForm.pas - модуль форми налаштувань;
- 
  - StatisticForm.pas - форма статистики;
- 
  - TesterMain.pas - головна форма проекту - модуль тестової оболонки;
- TZConfig.ini - модуль налаштування конфігурації;
- 
  - TestSuite.exe - виконуваний модуль.

Фрагмент програмного коду для створення і збереження форми результатів тестування можна переглянути у додатку А.



Вікно програми на початку тестування має вигляд (рис. 2.2)

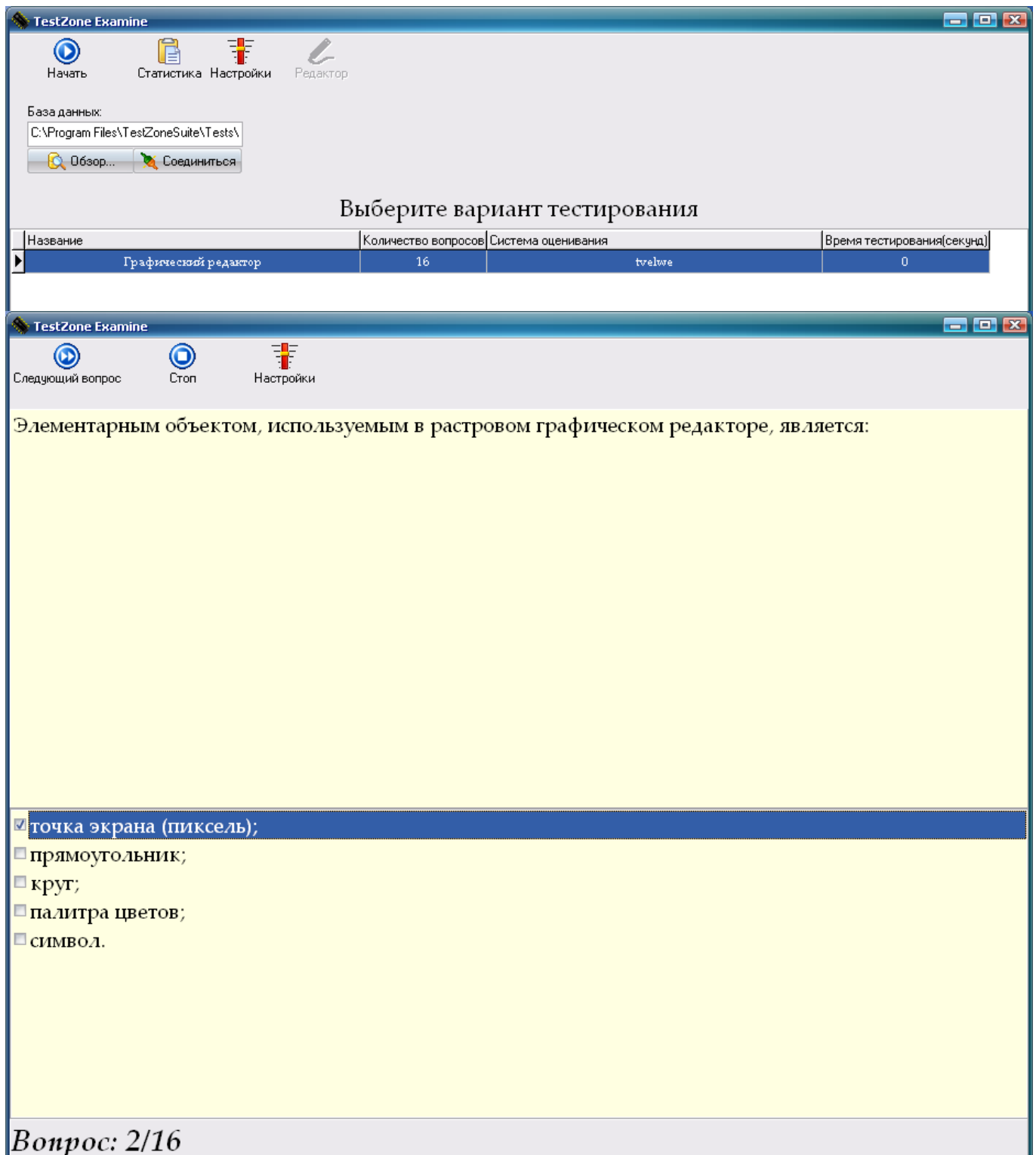


Рис. 2.2 Вікна програми на початку тестування

До програмного комплексу входить база даних тестів з таблицями:

- marksystem – таблиця рівнів оцінювання;

- questions – таблиця завдань;
- results – таблиця зберігання оцінок та відсоткового рівня знань;
- tests – набір тестів за темами.

На початку тестування необхідно приєднатися до бази даних Tests. Завдання пропонуються з таблиці questions (рис. 2.3)

qid	qtype	qtext	qpicture	qanswers
6	radio	Одной из основных функций графического редактора является		ввод изображений;
7	radio	Элементарным объектом, используемым в растровом графическом редакторе является		точка экрана (пиксель);
8	radio	Деформация изображения при изменении размера рисунка называется		векторной графики;
9	radio	Примитивами в графическом редакторе называют:		простейшие фигуры, рисуемые с помощью сплайнов;
10	radio	Кнопки панели инструментов, палитра, рабочее поле, меню		полный набор графических примитивов графического редактора;
11	radio	Наименьшим элементом поверхности экрана, для которого выделяется отдельный адрес в памяти, является		точка;
12	radio	Сетка, которую на экране образуют пиксели, называют:		видеопамять;
13	radio	Графика с представлением изображения в виде совокупности точек называется		фрактальной;
14	radio	Пиксель на экране монитора представляет собой:		минимальный участок изображения, которому соответствует один адрес в памяти;
15	radio	Видеоадаптер - это:		устройство, управляющее работой монитора;
16	radio	Для хранения 256-цветного изображения на кодирование каждого пикселя требуется		2 байта;
17	radio	Цвет точки на экране цветного монитора формируется из смеси		красного, зеленого, синего и яркости;
18	radio	Растровый графический файл содержит чёрно-белое изображение		10000 бит;
19	radio	Растровый графический файл содержит чёрно-белое изображение		100 бит;
20	radio	Для двоичного кодирования цветного рисунка (256 цветов) требуется		100 бит;
21	radio	Видеопамять - это:		электронное устройство для хранения двоичной информации;
*	(№)			

Рис. 2.3 Таблица завдань бази даних тестів

Результати тестування зберігаються у таблиці Results, задіяне автоматичне зберігання з процентним співвідношенням правильних і помилкових відповідей.

Вікно закінчення тестування має вигляд (рис. 2.4)

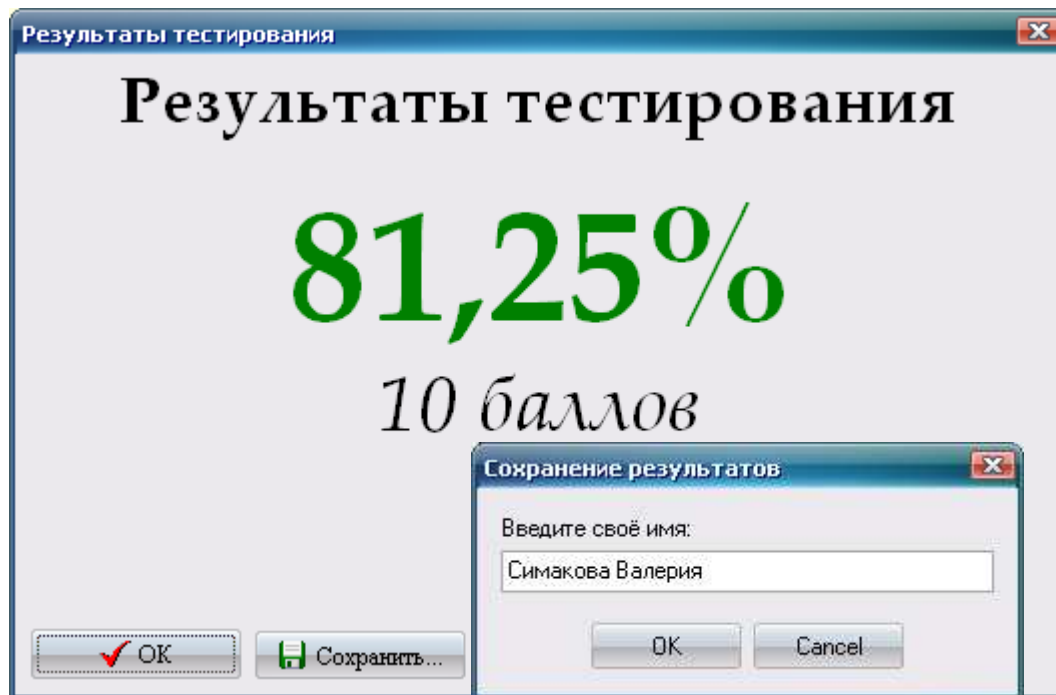


Рис. 2.4 Вікно завершення тестування і збереження результатів

The window, titled "Статистика", has tabs for "Результаты тестов" and "Журнал". It contains a table with the following data:

Имя	Процент правильных ответов	Оценка	Затраченное
autosave40472	75	9	
autosave40472	68,75	8	
autosave40472	12,5	3	
autosave40473	81,25	10	
Симакова Валерия	81,25	10	

At the bottom right, there is a button labeled "Очистить" with a red 'X' icon.

Рис. 2.5 Вікно перегляду статистичних даних

На рис. 2.5 представлені результати тестувань різних учнів з використанням відсоткової шкали, шкали балів та шкали затраченого часу. Для ведення статистики та аналізу результатів тестування використовується автоматичне збереження даних.

### **2.3 Результати практичного використання**

Застосування в навчальному процесі комп'ютерного тестування показало, що в учнів з'являється впевненість в об'єктивності оцінки одержаних знань, а можливість самоконтролю є додатковою мотивацією до вивчення дисципліни. Використання програмного комплексу дає змогу перейти на вищий рівень засвоєння знань, що у свою чергу підвищує якість підготовки учня.

У результаті роботи маємо досить зручну та функціонуючу програму для проведення тестування, можливості якої не обмежуються лише школами або навчальними закладами. Її можна використовувати у будь-якій сфері життєдіяльності людини, де періодично виникає необхідність у проведенні тестування або в визначенні реального рівня знань людей. До того ж програма не обмежує тих, хто буде створювати питання для тестів, адже обсяг завдань для тестування не обмежується нічим, окрім обсягу пам'яті вашого комп'ютера.

## ВИСНОВКИ

1. З точки зору літератури з розробки тестів і визначення їх якості, визначені вимоги до програми контролю знань:
  - валідність;
  - однозначність;
  - надійність;
2. Класифіковані і проаналізовані програмні засоби, які можуть бути використані для комп'ютерного тестування. Визначені критерії, яким має задовольняти досконала програма перевірки якості знань:
  - кількість питань у тесті має визначати сам учитель, чи та людина, яка проводить тестування;
  - кількість балів, що надається за правильну відповідь також має вказувати вчитель;
  - процес зміни запитань і варіантів відповідей має бути простим, легким та доступним для виконання навіть для тих, хто має лише основні невеликі навички в роботі з комп'ютером;
  - сам процес створення тесту та його проведення повинен бути простим і зручним як для вчителя, так і для учнів;
  - аналізування виконаної роботи також повинно проводитися досить швидко та легко.
3. На основі визначених критеріїв розроблена програма для тестування, яка може розширити коло використання (наприклад у профорієнтації), бо за останній період тести на профорієнтацію набувають усе більшої популярності не тільки за кордоном, а й у нашій країні.
4. Проведена експериментальна перевірка ефективності використання програми, яка показала підвищення інтересу учнів до предмету, який вивчається, підвищення рівня знань з контрольної теми та економії часу на проведення перевірки знань.

Практичне значення дослідження полягає в тому, що розроблена й впроваджена комп'ютерна програма, яка забезпечує можливість створення комп'ютерних тестів для різних предметів з оптимізацією перевірки якості знань.

Розроблена система пройшла тестування на уроках з інформатики у 10-х класах та при вивченні спецкурсів «Основи інформаційних технологій», «Основи програмування», «Основи комп'ютерної графіки» у 11-Б класі ЗОШ № 17 міста Кременчука.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ніна Тверезовська. Визначення якості тестів засобами кваліметрії// Інформатика. - 2006. – № 2. - С.45-51.
2. Сергій Лисенко. Особливості функціонування програмного комплексу тестування знань учнів// Інформатика. - 2006. – № 2. - С.56-58.
3. Бочкин А. И. О надежности оценки доли знаний методом теста с выбором варианта ответа// Информатика и образование. – 2002. - №12. – С. 55-60.
4. Свиридов А.П. Основы статистической теории обучения и контроля знаний. – М.: Высшая школа, 1981. – 263 с.
5. Ротаєнко П.А. Про вірогідність результатів тестування із закритою формою завдань// Комп'ютер у школі та сім'ї. - 2004. - № 6. – С. 123-124.
6. Зоц В.М. Контроль і оцінка// Шлях освіти. – 1997. - № 1 – С.53-55.
7. Офіційний сайт програми УТК 1.51 [onLine] Доступ: <http://utk.mastak.ru/>
8. Сайт програми Асистент 2 [onLine] Доступ: <http://theosoft.virtuale.net>
9. Офіційний сайт програми OpenTest [onLine] Доступ: [www.opentest.com.ua](http://www.opentest.com.ua)
10. <http://psifactor.dp.ua/index.php?act=show&idraz=2&idpod=2>

# ДОДАТКИ



## Додаток А-1

Фрагмент програмного коду для створення і збереження форми результатів  
тестування

```
procedure ShowResults;
var Mts,Scs:Longint;
    Time:Longint;
begin
  with FResults do
  begin
    LPercent.Caption:=FormatFloat('##0.##"%"',ResultPercent);
    if ResultPercent>=60 then LPercent.Font.Color:=clGreen
    else if ResultPercent>=50 then LPercent.Font.Color:=clYellow
      else LPercent.Font.Color:=clRed;
    ResultMark:=PutMark(ResultPercent,TestMarkSys);
    LMark.Caption:=IntToStr(ResultMark)+' баллов';
    LTimeCaption.Hide;
    LTime.Hide;
    if TestTime>0 then
    begin
      Time:=TestTime-TimeElapsed;
      Mts:=Time div 60;
      Scs:=Time mod 60;
      if Mts<10 then LTime.Caption:='0'+IntToStr(Mts)+': '
      else LTime.Caption:=IntToStr(Mts)+': ';
      if Scs<10 then LTime.Caption:=LTime.Caption+'0'+IntToStr(Scs)
      else LTime.Caption:=LTime.Caption+IntToStr(Scs);
      LTimeCaption.Show;
      LTime.Show;  end;
    end;
  end;
end;
```

## Додаток А-2

Фрагмент програмного коду для створення і збереження форми результатів  
тестування

```

procedure TFResults.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    ResultsDisplaying:=False;
    Action:=caFree;
end;
procedure TFResults.BOKClick(Sender: TObject);
begin
    MarkQuery.Close;
    FResults.Close;
end;
procedure TFResults.BSaveClick(Sender: TObject);
    var UserName:AnsiString;
        DT:AnsiString;
        FSets:TFormatSettings;
begin
    if AutoSaving then
UserName:='autosave'+FormatFloat('###0',Date()+Time())
        else if not(InputQuery('Сохранение результатов','Введите своё
имя:',UserName)) then EXIT;
        if Length(UserName)>255 then Delete(UserName,255,Length(UserName)-
254);
        DT:=FormatDateTime('dddddd',Date)+FormatDateTime('tt',Time);
        FSets.DecimalSeparator:= '.';
        if not(ExecuteSQLQuery(MarkQuery,'INSERT INTO
results(rname,rpercent,rmark,rtime,rdatetime) VALUES
('+QuotedStr(UserName)+','+FormatFloat('##0.##',ResultPercent,FSets)+
','+IntToStr(ResultMark)+','+IntToStr(TestTime-

```

## Додаток А-3

Фрагмент програмного коду для створення і збереження форми результатів  
тестування

```
TimeElapsed)+' , '+QuotedStr(DT)+' );', False))
  then
  begin
    Application.MessageBox('Ошибка при сохранении результатов!
Операция не выполнена', 'Сохранение результатов', MB_OK+MB_ICONERROR);
    EXIT;
  end
  else Application.MessageBox('Сохранение успешно
выполнено!', 'Сохранение результатов', MB_OK+MB_ICONASTERISK)
end;
procedure TFResults.FormCreate(Sender: TObject);
begin
  ResultsDisplaying:=True;
  ShowResults;
  LMark.Font.Name:=InterfaceFont;
  LPercent.Font.Name:=InterfaceFont;
  LTime.Font.Name:=InterfaceFont;
  LTimeCaption.Font.Name:=InterfaceFont;
  ResultCaption.Font.Name:=InterfaceFont;
  if AutoSave then
  begin
    AutoSaving:=True;
    BSave.Click;
    AutoSaving:=False;
  end;
end;
end.
```