

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ДЕННОЇ ОСВІТИ

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Допускається до захисту

Завідувач кафедри _____ О.ОЛЬХОВСЬКА
(підпис)

«_____» _____ 2022 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО ДИПЛОМНОЇ РОБОТИ

на тему

«Створення програмного забезпечення тренажеру з теми «Методології програмування» англomовного дистанційного навчального курсу «Теорія програмування»

**зі спеціальності 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
ступеня магістра**

Виконавець роботи Купченко Олексій Володимирович _____
«_____» _____ 20__ р.

(підпис)

Науковий керівник к. ф.-м. н., доц. Черненко О.О.

_____ «_____» _____ 20__ р.
(підпис)

ПОЛТАВА 2022 р.

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

ЗАТВЕРДЖУЮ

Завідувач
кафедри _____ О.ОЛЬХОВСЬКА
(підпис)

«_03_» вересня 2022 р.

**ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ ДИПЛОМНОЇ РОБОТИ**

Здобувач вищої освіти зі спеціальності 122 «Комп'ютерні науки»

Освітня програма «Комп'ютерні науки»

Прізвище, ім'я, по батькові Купченко О.В.

- 1. Тема:** «Створення програмного забезпечення тренажеру з теми «Методології програмування» англomовного дистанційного навчального курсу «Теорія програмування»

затверджена наказом ректора № _____-Н від «___» _____ 2022 р.

Термін подання студентом дипломної роботи _____ «_16_»__11__ 2022 р.

2. Вихідні дані до дипломної роботи _____

3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

4. Перелік графічного матеріалу (з точним визначенням кількості блок-схем, іншого графічного матеріалу) _____

5. Консультанти розділів дипломної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ			
Постановка задачі			
Інформаційний огляд			
Теоретична частина			
Практична частина			
Висновки			

Список використаних джерел			
----------------------------	--	--	--

6. Календарний графік виконання дипломної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	09.09.2022	
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику	15.09.2022	
3. Постановка задачі	25.09.2022	
4. Інформаційний огляд джерел бібліотек та інтернету	01.10.2022	
5. Теоретична частина	10.10.2022	
6. Практична частина	10.11.2022	
7. Закінчення оформлення	16.11.2022	
8. Доповідь студента на кафедрі	20.11.2022	
9. Доробка (за необхідністю), рецензування	25.11.2022	

Дата видачі завдання «03» вересня 2022 р.

Здобувач вищої освіти _____
(підпис)

Науковий керівник _____
(підпис) _____
(науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту дипломної роботи

Дипломна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «_____» _____ 2022 р.

Секретар ЕК _____
(підпис) _____
(ініціали та прізвище)

Затверджую
Зав. кафедрою _____
к.ф.-м.н. О. ОЛЬХОВСЬКА

Погоджено
Науковий керівник _____
доцент, Черненко О.О.

«_____» _____ 2022 р.

«_____» _____ 2022 р.

План

**Дипломної роботи здобувача вищої освіти ступеня магістра
спеціальності 122 Комп'ютерні науки
освітня програма 122 Комп'ютерні науки
Прізвище, ім'я, по батькові Купченко О.В.**

**На тему: «Створення програмного забезпечення тренажеру з теми
«Методології програмування» англomовного дистанційного навчального
курсу «Теорія програмування»**

Вступ

- 1. ПОСТАНОВКА ЗАДАЧІ**
 - 1.1 Змістовна постановка
 - 1.2 Обґрунтування методу та його алгоритм
- 2. ІНФОРМАЦІЙНИЙ ОГЛЯД**
 - 2.1 Теоретичні основи
 - 2.2 Інформаційна модель
 - 2.3 Огляд існуючих аналогів програмного тренажеру
- 3. ТЕОРЕТИЧНА ЧАСТИНА**
 - 3.1 Алгоритмізація задачі за темою роботи
 - 3.2 Розробка алгоритму роботи тренажеру
 - 3.3 Ознайомлення з середовищами для розробки програмного забезпечення
 - 3.4 Вимоги до технічного забезпечення
- 4. ПРАКТИЧНА ЧАСТИНА**
 - 4.1 Опис процесу розробки програми
 - 4.2 Проектування та розробка БД
 - 4.3 Опис роботи програми
 - 4.4 Дослідження можливостей програмної реалізації

Висновки

Додатки

Здобувач вищої освіти _____ О. Купченко
«_____» _____ 2022 р.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	7
1.1. Змістовна постановка	7
1.2. Обґрунтування методу та його алгоритм	7
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД	10
2.1. Теоретичні основи	10
2.2. Інформаційна модель та її характеристики	14
2.3. Огляд існуючих аналогів програмного тренажеру	17
3. ТЕОРЕТИЧНА ЧАСТИНА	25
3.1. Алгоритмізація задачі за темою роботи	25
3.2. Розробка алгоритму роботи тренажеру	27
3.3. Ознайомлення з середовищами для розробки програмного забезпечення	31
3.4. Вимоги до технічного забезпечення.....	38
4. ПРАКТИЧНА ЧАСТИНА.....	40
4.1. Опис процесу розробки програми	40
4.2. Проектування та розробка БД.....	42
4.3. Опис роботи програми	43
4.4. Дослідження можливостей програмної реалізації.....	47
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТКИ	51
Додаток А(DB.cs)	51
Додаток Б(Form1.cs)	51

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

ПК – Персональний комп'ютер

ПЗ – програмне забезпечення

SQL – structured query language

ОС – операційна система

ОЗУ – оперативний запам'ятовуючий пристрій

DOS – disk operation system

BIOS – basic input/output system

БД – база даних

СУБД – система управління базами даних

ІС – інформаційна система

РЕФЕРАТ

Тема дипломної роботи: «розробка програми для конвертації даних на основі XML формату»

Дипломна робота виконана на 76 сторінках, містить 26 малюнків, 14 використаних джерел.

Актуальність теми, обумовлена потребою створенні автоматизованої інформаційної системи для навчання та тестування знань учнів та відпрацювання навчального матеріалу з інструментарієм для аналізу результатів, щоб за допомогою цієї системи підвищити якість освітньої програми та автоматизувати процес перевірки пройдених студентами тестів.

Мета роботи: розробити необхідне математичне та програмне забезпечення для відпрацювання навчального матеріалу та аналізу результатів навчання.

Для виконання поставленої задачі необхідне вирішення наступних завдань:

- проаналізувати інформаційні системи аналогічного призначення;
- розглянути інструменти розробки та вибрати найбільш підходящі для розробки даної інформаційної системи;
- сформулювати технічне завдання;
- Розробити ІС;
- Запровадити ІС;
- здійснити оцінку ефективності взаємодії користувачів із ІС.

Об'єктом дослідження дипломної роботи є навчання і засвоєння матеріалу школярами.

Предметом дослідження є онлайн інструмент для тренувань та тестування студентів.

Наукова новизна полягає в тому, що створено інформаційну систему, що має в своєму розпорядженні інструменти збору та аналізу даних для досліджень, розроблено формулу для розрахунку показника рівня засвоєння матеріалу учнів для побудови кривої навчання.

Практична значущість полягає в тому, що дана інформаційна система буде застосовуватись в університеті для відпрацювання навчального матеріалу та аналізу результатів учнів.

Економічна ефективність проекту полягає в тому, що система, що розробляється, дозволяє багаторазово розширювати користувальницьку базу, звільняючи викладача від рутини, що забирає велику кількість часу. Розрахунки з наближеною до сьогоденної кількості користувачів показують, що навіть у такому разі автоматизована система тестування знижує витрати на бізнес-процес перевірки завдань більш ніж удвічі.

Ключові слова: тренажер, база даних, аналіз результатів, тестування

ВСТУП

В сучасному світі комп'ютеризація грає значну роль у розвитку суспільства. В часи пандемії досить поширеним стало дистанційне навчання, а отже, створення різних програмних засобів, які активізують навчальну активність студентів, залишається досить актуальним. Серед різних навчальних програмних засобів таких, як, програми-тести, електронні підручники та довідники, особливу увагу займають програми-тренажери.

Створення такої програми в рамках даної роботи дає можливість відпрацювання студентами спеціальності «Комп'ютерні науки» навичок розв'язування задач з теми «Методологія програмування» англomовного дистанційного навчального курсу «Теорія програмування».

Серед тренажерів даного дистанційного курсу подібних задач не було реалізовано, тому створення такого програмного продукту, який може бути використаний для всіх форм навчання, є важливим.

Мета роботи – побудувати алгоритм та створити програмну реалізацію навчального тренажеру з теми «Методологія програмування» англomовного дистанційного навчального курсу «Теорія програмування».

Завдання, які необхідно виконати для досягнення мети:

- опрацювати методичні рекомендації до виконання курсової роботи;
- сформулювати постановку задачі;
- виконати огляд літературних джерел, що використовується для вивчення даної теми, а також огляд існуючих програмних продуктів, які використовуються при вивченні різних дисциплін;
- вивчення теоретичного матеріалу з теми;
- розробка алгоритму навчального тренажеру;
- побудова блок–схеми алгоритму тренажеру;
- вибір оптимальної платформи для програмної реалізації тренажеру;
- створення програмної реалізації тренажеру.

Об'єктом розробки є програмне забезпечення тренажеру з теми «Методологія програмування» англomовного дистанційного навчального курсу «Теорія програмування».

Предметом розробки є тренажер для англomовного дистанційного навчального курсу «Теорія програмування».

Методи розробки. Для створення програми було обрано мову програмування Microsoft Visual Studio. Для створення алгоритму тренажеру використано теоретичні матеріали дистанційного курсу «Теорія програмування».

Дипломна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел та додатків.

Вступ – містить сутність роботи та актуальність досліджень та розробок.

Перший розділ – змістовна постановка задачі.

Другий розділ – опис результатів аналізу сучасного стану досліджень використання тренажерів для навчання.

Третій розділ – дослідження властивостей методу та їх результати. Представлено сутність та відмінність досліджень алгоритмів використання тренажерів для навчання, а також структуру процедур використання тренажерів для навчання.

Четвертий розділ –. представляє структуру, процес проектування і розробки інструментального забезпечення, призначеного для проведення дослідження.

Висновки - складаються з 1 сторінки;

Список літератури – включає в себе список використаної літератури;

Додатки – код робочого проекту. Таблиць – 1, рисунків – 25 , бібліографія – 13.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

1.1. Змістовна постановка

В дипломному проєкті головною задачею є створення елементів програмного забезпечення тренажеру з теми «Методологія програмування» англomовного дистанційного навчального курсу «Теорія програмування».

У зв'язку з цим необхідно:

- дослідити функції і методи оцінювання рівня знань за допомогою інструктивних контрольних робіт.
- створити базу даних, що містить тестові завдання, теми, предмети та учнів.
- розробити та реалізувати алгоритми формування порядку та змісту запитань у тематичних блоках.
- створити блок-схему роботи складеного алгоритму.
- ознайомитися з тренажерами подібної тематики, проаналізувати їх роботу, щоб уникнути негативних аспектів оглянутих програм і взяти (якщо можливо) позитивні.

основні вимоги до тренажеру:

- простий і зрозумілий інтерфейс програми;
- зручна взаємодія з тренажером;
- повідомити користувача у разі вибору неправильної відповіді;
- перевірити та налагодити систему.

1.2. Обґрунтування методу та його алгоритм

Навчальні тренажери можуть використовуватися як для визначення знань учня в певній галузі, так і з навчальною метою.

Систематична перевірка знань великої кількості досліджуваних студентів призводить до необхідності автоматизованої перевірки знань, використання комп'ютерної техніки та відповідних процедур перевірки знань.

Комп'ютерні тренажери все частіше використовуються в освіті як ефективний спосіб перевірки та навчання знань. Одна з його переваг полягає в тому, що отримання достовірних результатів контролю займає мінімум часу, а результати доступні практично відразу після завершення контрольного тестування знань після навчання з тренажером. Порівняно з традиційним оцінюванням і контролем знань тренування з тренажером відрізняються об'єктивністю вимірювання результатів навчання, оскільки вони керуються не суб'єктивною думкою викладачів, а об'єктивними критеріями.

Результати автоматизованих перевірок легше аналізувати, ніж суб'єктивні оцінки.

Застосування програм тренування та комп'ютерних тестів у навчанні є економічно вигідним і підвищує ефективність навчального процесу, об'єктивність оцінювання рівня знань, є розумним доповненням до інших методів перевірки знань.

При використанні комп'ютерного тренування з тренажером важливу роль відіграє вибір програми перевірки знань, розробка та створення навчальних тестів.

Програма тренування містить усі необхідні інструменти для проведення всього циклу тренування - від розробки контрольних тестів, проведення тестів до запису та аналізу результатів тестування.

Призначений для індивідуальної та групової перевірки знань студентів, співробітників у різних галузях. Контрольні роботи можуть включати предметні розділи. Проблема в навчанні – відформатований текст. Візуальний редактор тестів дозволяє швидко створювати тести з високим ступенем видимості та обізнаності. У створеному тесті ви можете використовувати різні види вибору, введення відповідей, що дозволяє ефективно контролювати знання.

Налаштування системи дозволяє змінювати тестовий режим незалежно від навчальної туди.

Програма призначена для запису результатів тестування та моніторингу процесу контролю навчання на комп'ютерному тренажері у режимі реального часу.

За допомогою цієї програми ви можете проходити навчання з дисципліни «програмування» та проводити індивідуальний аналіз результатів тестування.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Теоретичні основи

Протягом 20 століття спостерігалася стійка глобальна тенденція до розширення використання широкомасштабних методів навчання та тестування у всіх сферах людської діяльності. Для українських викладачів питання стає актуальним після появи в країні альтернативних закладів різних форм власності, і, як наслідок, державного механізму акредитації, акредитації та акредитації вищих навчальних закладів у його нинішньому вигляді, який включає елементи технології навчання та тестування учнів.

Найважливішою складовою системи навчання і частиною навчального процесу є контроль навчальної діяльності студентів, спрямований на визначення успішності навчання кожного студента, аналіз отриманих результатів і корекцію процесу безперервної освіти. Удосконалення системи оцінювання та модернізація діагностики навчання учнів є інструментами підвищення якості освіти при реформуванні змісту освіти.

Останнім часом, окрім уже згадуваного поняття «діагностика», замість традиційного поняття «контроль» все частіше вживається поняття «моніторинг». При проектуванні навчального процесу перед учителем постає завдання вибору методів і форматів контролю навчальних досягнень учнів, нормативів якості засвоєння навчального матеріалу, розробки процедур контролю, обґрунтування індивідуальних корекцій навчальної діяльності учнів.

Забезпечення не лише контролю та оцінювання результатів, а й процесу навчання сприяє пошуку дієвих і об'єктивних методів контролю знань. Системи оцінювання та контролю мають відповідати вимогам до управління пізнавальною діяльністю учнів і слугувати відповідним інструментом для їх реалізації [10-11].

Суб'єктивність оцінювання знань певною мірою пов'язана з недостатньою розробленістю методів контролю системи знань. Об'єктивний підхід полягає в

тому, щоб завжди використовувати відповідні інструменти для виявлення наявності знань. Сучасні технології використовують тести як інструмент навчання та вимірювання рівня знань, за допомогою якого можна не тільки визначити якість освіти, а й оптимізувати управління навчальним процесом.

Ефективність функціонування системи професійної освіти значною мірою залежить від діяльності, яка привносить у ВНЗ інноваційну діяльність. Вони змушують усіх учасників освітнього процесу виявляти й аналізувати свої знання, уміння та рівень навичок, спрямовуючи їхню діяльність на перспективний шлях трансформації. При цьому головна роль належить вчителю. Його бажання запроваджувати інновації та його здатність організовувати та керувати цим процесом є невід'ємною частиною успішних інноваційних зусиль.

Однією з інноваційних форм комп'ютеризованих систем навчання є тестування.

Test — слово англійського походження, що означає іспит, спроба, перевірка. Тест — стандартизоване завдання, яке за результатами тестування робить висновок про знання, уміння, навички (здібності, професійну придатність, обдарованість тощо) особи, що перевіряється. У сучасній теорії та практиці тестового контролю розрізняють понад 20 видів тестів: залежно від мети, характеру та функції контролю, характеру та форми відповіді.

Іспити як система завдань певної форми та відповідного змісту є науковими інструментами оцінювання знань, умінь і навичок студентів, що сприяють індивідуальному контролю результатів навчання кожного, мобільним процесом управління освітою. Контрольні роботи, як правило, ефективніші за традиційні форми контролю знань (контрольні роботи, іспити, заліки, диференційовані оцінки). Адже недостатньо оцінити рівень знань, треба ще й передбачити, як учні ці знання використовуватимуть. При цьому зростає роль психологічного тестування.

Необхідно поєднати його з перевіркою рівня знань. Тоді результати будуть більш об'єктивними, враховуючи не лише рівень і кількість знань, а й риси характеру учня, його нахили та навчальні можливості, уміння аналізувати,

узагальнювати, встановлювати причинно-наслідкові зв'язки. Досвід показує, що в більшості випадків оцінювання та контроль успішності учнів здійснюється за кінцевими результатами. На жаль, його активність, рівень динамічного розвитку та вплив зовнішніх факторів на момент оцінки оцінити практично неможливо.

Практика показала, що навчання за допомогою тестування має такі переваги перед іншими формами контролю знань:

- Може перевірити якість знань і навичок визначеної кількості студентів протягом досить обмеженого періоду часу;
- Можливий контроль знань, умінь, навичок на необхідних, заздалегідь запланованих рівнях;
- Справжній самоконтроль;
- більш-менш об'єктивно оцінювати знання;
- Увага учнів фіксується не на формуванні відповіді, а на характері її розуміння;
- Створити умови для постійного зворотного зв'язку між учнями та вчителями.

Проте тестовий контроль знань має і суттєві недоліки:

- Імовірність випадкового вибору правильної відповіді;
- Використовуйте закрите тестування, щоб оцінити лише ймовірність кінцевого результату (вірно-хибно), а процес, який призвів до цього, не розкривається;
- Психологічний недолік — стандартизоване мислення, незалежно від рівня розвитку особистості;
- Багато часу, витраченого на складання необхідних тестових «бібліотек», їх варіантів, складність процесу;
- Тести не сприяють розвитку мови [5].

Систематичне навчання за допомогою тестування запроваджено з метою підвищення ефективності системи перевірки знань учнів. Поступово вводяться іспити на спеціальності «Офіс-менеджмент», що дозволяє психологічно

підготувати студентів. Спочатку надавалися прості тести з необов'язковими відповідями, і лише через деякий час з'явилися більш складні конструкції. Навчальні тести, складені з використанням критично-ситуаційного підходу, виявилися високопродуктивними.

Інформація в тестових питаннях може бути подана в довільній формі – текст, графічні зображення, звукова інформація, відео сюжети, формули тощо. Крім того, ми практикуємо відкритість і доступність бази екзаменаційних питань, тобто зміст екзаменаційних питань добре відомий студентам. Але при цьому тести постійно оновлюються з урахуванням змін курсів, програм і сьогоdnішніх реалій. Контрольні роботи з діловодства гарантують об'єктивність в оцінюванні знань, умінь і навичок учнів, сприяють усуненню проявів суб'єктивізму, а отже формують позитивне ставлення до предмета та вчителя.

При проведенні екзаменаційного контролю дуже важливо дотримуватися організаційних моментів (пояснити мету іспиту, послідовність і схему його проведення, визначити терміни і проведення, а також надати кожному студенту бланк відповідей встановленого зразка, що значно економить час іспиту. Студенти та викладачі).

Тестові завдання значно скорочують час очікування студентами оцінок, що є важливим фактором – як психологічним, так і педагогічним.

Навчання на тренажері може охоплювати окрему тему, частину або весь матеріал курсу, а його результати повинні бути проаналізовані. Розбір необхідний для того, щоб учень міг перевірити, наскільки добре він оцінює свої знання, вірить у свої сили і коригує свою підготовку. Учитель не тільки фіксує неправильні факти і називає правильну відповідь, але й детально пояснює учням причини неправильної поведінки.

Учням, які допустили помилку, було запропоновано письмово або усно відповісти на запитання: Чому обрана відповідь неправильна? Надайте докази на підтримку правильного вибору. Вивчення неправильної відповіді, усвідомлення помилки та її причини, знаходження неправильної ланки в ланцюзі розумової діяльності значно зменшує можливість запам'ятовування неправильних знань і

«заміни» правильної відповіді. Таким чином виключається практика «вгадування» студентами правильного розв'язання поставлених на іспиті питань.

Наступним етапом впровадження та використання тестів при вивченні різних предметів є їх електронна форма. Їх робота забезпечується на локальному комп'ютері та локальній мережі класу. Такий формат часто викликає додатковий інтерес у студентів, крім того, він дозволяє їм самостійно перевірити та оцінити свій рівень знань з певної теми або набору тем курсу без участі викладача. Програми комп'ютерного тестування використовують анімацію, звуки, елементи гри та систему для запису прогресу та результатів тестування, особливо таких показників, як час вирішення кожного запитання, кількість помилок і правильних відповідей. Підсумковий рахунок [2-3].

Важливою умовою контрольної роботи як загально навчального засобу є її періодичність, яка залежить від навчального предмета, його ролі та місця в навчальній програмі, специфіки засвоєння знань. Тестування повинно стати звичною і зручною формою регулярного контролю знань студентів. Необхідно пам'ятати, що контрольна робота - це не самоціль, а ефективна форма повторення - узагальнення й організації вивченого. Контрольно-оцінювальна функція навчання є лише одним із елементів добре організованого, технічно продуманого навчального процесу. Якщо учні володіють міцними знаннями, їх оцінки, у будь-якій формі, не повинні викликати особливих труднощів.

2.2. Інформаційна модель та її характеристики

Застосування програм навчання на тренажерах та комп'ютерних тестів у навчанні є економічно вигідним і підвищує ефективність навчального процесу, об'єктивність оцінювання рівня знань, є розумним доповненням до інших методів перевірки знань. При використанні комп'ютерного навчання важливу роль відіграє вибір програми перевірки знань, розробка та створення тестів.

Програма навчання містить усі необхідні інструменти для проведення всього циклу тестування - від розробки контрольних тестів, проведення тестів до запису та аналізу результатів тестування. Призначений для індивідуальної та групової перевірки знань студентів, працівники всіх сфер.

Контрольні роботи можуть включати предметні розділи. Проблема в тесті – відформатований текст. Візуальний редактор тестів дозволяє швидко створювати тести з високим ступенем видимості та обізнаності. У створеному тесті можна використовувати різні типи варіантів, введення відповідей (8 типів), що дозволяє створювати ефективні контрольні завдання. Розроблені в програмі тести використовуються в зашифрованому вигляді.

Налаштування системи дозволяє змінювати тестовий режим незалежно від тесту.

Програма призначена для навчання та запису результатів тестування, моніторингу процесу контролю тестування на інших комп'ютерах у режимі реального часу. Дозволяє створювати різні протоколи перевірки знань.

За допомогою цієї програми можна проводити індивідуальний аналіз результатів навчання.

Функціональні характеристики та склад структури програми повинні відповідати таким вимогам:

- забезпечення безпеки;
- дотримання кількох дисциплін;
- простота використання (адаптивний інтерфейс);
- широкий спектр питань;
- безперешкодний доступ викладачів;
- наявність зберігання інформації;
- гласність результатів тестування.

Підтримуйте безпеку програми, виконуючи такі дії:

Одноразово згенеровані динамічні паролі трапляються з викладачами, і студенти не зможуть повторно скласти той самий іспит із тими самими

обліковими даними. Ці вимоги виконуються завдяки багаторівневій структурі програми – усі дані про облікові записи користувачів (учнів) зберігаються на сервері програми – на комп'ютері викладача. Створення списків користувачів та матеріалів для їх ідентифікації/аутентифікації може здійснювати лише викладач, оскільки діалогова форма запису/вилучення даних про користувачів із сервера встановлена лише на комп'ютері викладача.

Для перевірки матеріалу ключ перевірки знаходиться на комп'ютері викладача.

Доступність інтерфейсу визначається як набір інструментів і структура, яка дозволяє будь-якому користувачеві взаємодіяти з програмою з мінімальними зусиллями та часом. Система реалізує наступні 8 пунктів, які необхідні для справді хорошого інтерфейсу:

1. Зрозумілість;
2. Стислий;
3. Схвалення;
4. Чутливість;
5. Замовлення;
6. Привабливість;
7. Ефективність;
8. Побалуйте себе.

Тест оцінює три типи запитань: запитання, на які необхідно відповісти безпосередньо шляхом введення відповіді, запитання з варіантами відповідей, лише одна з яких правильна, і запитання з варіантами відповідей, які містять кілька правильних варіантів. Різні формати запитань допомагають сконцентрувати увагу учнів. Варто зазначити, що процес пошуку правильної відповіді набагато простіший, ніж формулювання самодостатньої відповіді на запитання. Однак тестувальники часто плутаються в одному з варіантів відповіді, який, на його думку, здається більш правдоподібним, ніж правильний.

Розробка матеріальних тестів має творчу складову – демонстрацію вміння писати неправильні, але дуже обґрунтовані варіанти відповідей. Також

вважається, що популярність питань зі списками варіантів відповідей є перевіркою знань нижчого рівня.

Призначте варіанти завдань, оберіть один, а із запропонованих завдань виберіть найбільш правильну відповідь. Тому в інструкції до таких завдань також написано: «Обведи число з найбільш правильною відповіддю». Природно, всі інші відповіді до завдання вважаються правильними, але в різному ступені.

Усі дані, включаючи інформацію про студентів і створені облікові записи, зберігаються в інтегрованій базі даних.

Результати тестового навчання завантажуються в каталог, доступний лише викладачам для подальшої обробки. Крім того, після проходження тесту студенти можуть самостійно оцінити свої результати.

2.3. Огляд існуючих аналогів програмного тренажеру

Для створення тестів з предметних областей розроблені та розробляються спеціальні інструментальні оболонки, які дозволяють створювати комп'ютерні тести шляхом формування бази даних із набору тестових завдань.

Існуючі інструментальні програми, що дозволяють розробляти комп'ютерні навчальні тести, можна розділити на дві категорії: загального призначення та спеціального призначення. Загальна програма містить навчальну оболонку як невід'ємну частину. Серед них Adonis, Linkway (Microsoft), Fairy, Raduga тощо. Спеціальна навчальна оболонка використовується лише для створення тестів.

Слід детально розглянути деякі існуючі початкові тестові тренажери.

1. Мультитестер <http://multitester.org.ua/>

Тестовий комплекс з гнучкими системами та роботою по мережі. Готуйте та перевіряйте результати навчання через локальну мережу за допомогою MultiTester. Дивно, що система має можливість відстежувати хід навчання в реальному часі. Крім того, в кінці тесту проводиться автоматичне оцінювання за встановленими критеріями.

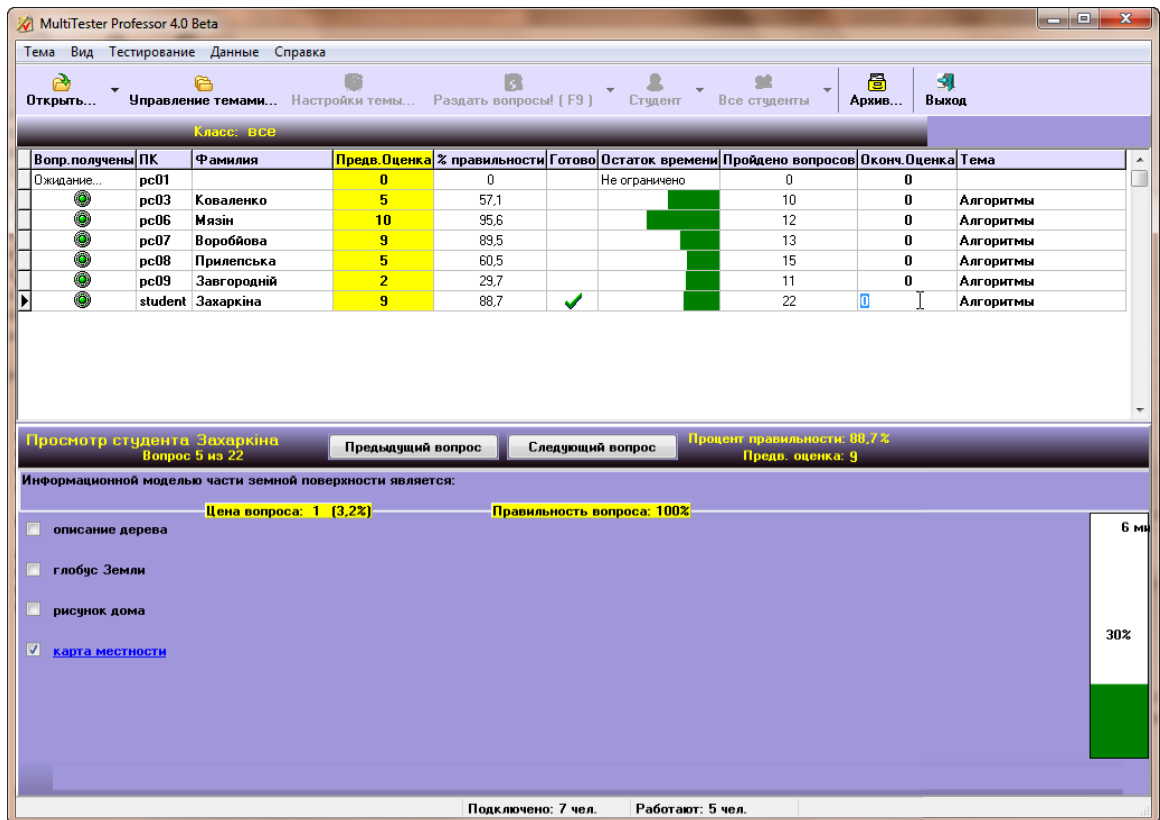


Рис. 2.1 – зображення роботи MultiTester

До складу системи MultiTester входять:

- Програма для вчителів (професор)
- Редактор питань (QuEditor)
- Студентська програма (Студенти)

Зауважте, що «Політест» працює за моделлю клієнт-сервер, де MultiTester Student є клієнтом, а сервером є MultiTester Professor. На персональному комп'ютері вчителя є база даних, де зберігаються всі дані. При необхідності викладач надсилає матеріал клієнту (MultiTester Student) через мережу. Зв'язок між клієнтом і сервером налаштовується автоматично, але його також можна налаштувати вручну.

Така структура зберігання інформації дозволяє правильно налаштувати систему захисту даних, тобто виключити несанкціоновані дії з боку просунутих

студентів, такі як дублювання відповідей, виправлення результатів тестування, виправлення питань.

Крім того, деякі учні можуть зловживати правилами самооцінювання. Щоб виключити протиправну діяльність, система MultiTester має такі засоби захисту, як одноразове відкриття та неможливість закриття програми. Тільки вчителі можуть закривати та перезапускати тест.

Навчання може проводитися процесом (тобто наступний студент сідає на місце студента, який виконав тест, отримує запитання та починає працювати незалежно від інших).

Мова інтерфейсу: російська.

Підтримувані операційні системи: Windows XP/7.

2. Система UniTest (<http://sight2k.com/rus/unitest/>)

Додаток являє собою єдину систему для розробки комп'ютеризованих навчальних тестів, проведення контрольних тестів через мережу або локально, написання звітів та аналізу результатів через остаточну сертифікацію.

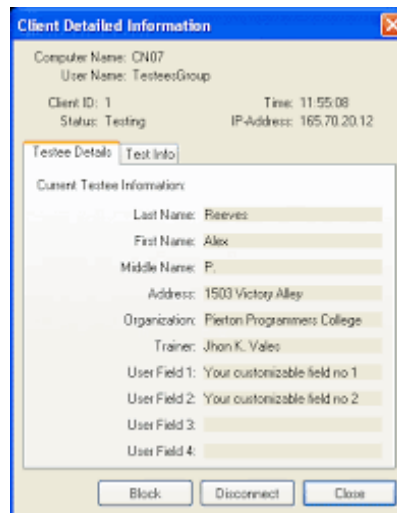


Рис. 2.2 – зображення роботи UniTest

Система підтримує основні типи навчальних питань, а також додаткові типи питань, такі як питання з різним рівнем оцінювання, порядком сортування, побудови листування, питання, що потребують розгорнутої відповіді, тощо.

У додатку при розробці тестів можна використовувати графічну інформацію, таблиці, аудіо-, відеоматеріали, редагування тексту тощо.

UniTest System має широкі можливості для налаштування розділів навчання, створення самогенерованих запитань для тесту, вибору режимів навчання, керування загальним доступом та іншими налаштуваннями системи конфіденційності, запобігання дублюванню та передачі результатів тесту, а також гнучкого призначення балів між завданнями.

Простий і зручний інтерфейс дозволяє швидко і точно скласти підсумковий іспит з усіма запропонованими варіантами. Крім того, UniTest System може інтегруватися з пакетом програм Microsoft Office.

Програмний комплекс розроблений і постійно розвивається в тісній співпраці з викладачами вузів, співробітниками відділів кадрів і відділів підготовки кадрів підприємств різних країн Росії і Європи.

Деякі особливості цього пакета:

Тестові запитання можуть містити будь-який об'єкт OLE (документи, графіку...), включаючи інтерактивні об'єкти (анімацію, відео- та аудіокліпи...).

– У кожне окреме питання можна вводити різноманітні ефекти (наприклад, відео- та аудіоматеріали, анімацію). Система також дозволяє включати в складання презентацій та інші процеси для створення середовища інформації для максимальної інтерактивності.

– Вбудований текстовий редактор, спрощена версія програми Microsoft Word, дозволяє створювати візуальний текст, який відповідає вашим вимогам до інтерфейсу.

– Вважайте, що відповіді на певні питання можна вважати частково правильними, наприклад, використовуючи відсотки.

– Підтримуються всі основні та багато додаткових типів запитань: множинний вибір, запитання з різною вагою відповіді, запитання з вибором набору елементів, порядок сортування (наприклад, «виберіть процесори в порядку зменшення потужності» або «виберіть правильні твердження»), Установіть відповідний зв'язок («відповідність між королем і часом його правління»), чи є в тексті відповіді ключове слово, введіть його безпосередньо;

– Можливість «нарощувати» базу запитань і призначати розділи в навчанні. (Наприклад, у вас є 3 логічних розділи в навчанні: легкі, середні та складні завдання. База становить 200 запитань для легких, -150 для середніх і -170 для важких, і 15 питань вибираються випадковим чином з бази для кожного тесту. легкі питання, середні 10 запитань і важкі запитання - 5 питань);

– Вбудована система стиснення даних використовує 448-бітну технологію BlowFish, світовий стандарт захисту інформації. При використанні цієї технології всі дані багаторазово стискаються, додатково захищаючи їх від несанкціонованого читання.

– Контролювати кількість і складати список активних користувачів навчальної оболонки в реальному часі. Це дозволяє записувати поведінку учасника навчання, наприклад, щоб побачити поточний бал.

– Режим обмеженого доступу та режим блокування.

3. Навчальна оболонка RichTest.

<http://soft.oszone.net/program/11544/RichTest/>

Тестові корпуси для навчальних закладів і корпоративне тестування. Складається з 3 модулів: Адміністратор, Клієнт, Звіт.

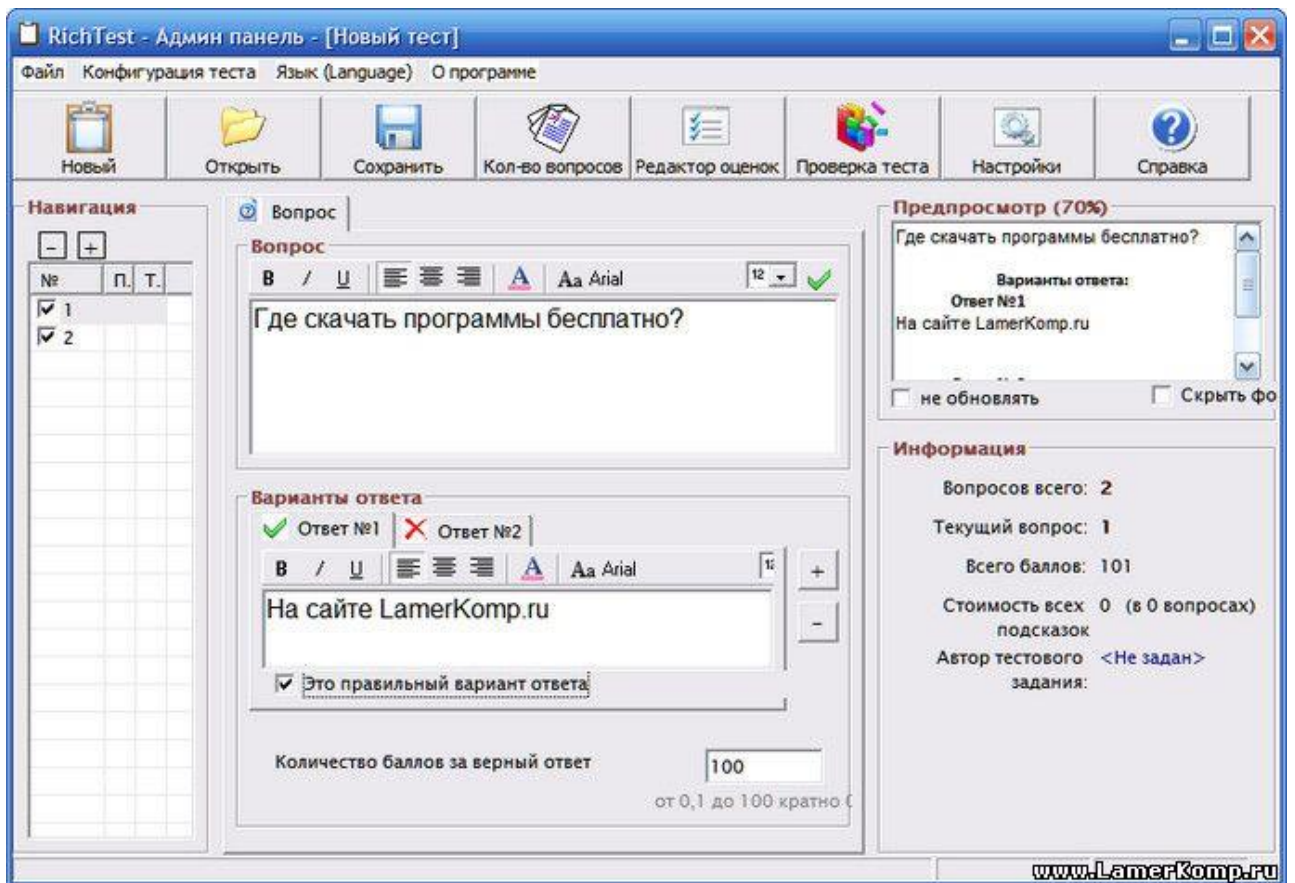


Рис. 2.3 – зображення роботи RichTest

У RichTest є різноманітні вбудовані шаблони, які можна використовувати для зміни зовнішнього вигляду ваших тестів. Також можливе детальне налаштування системи оцінювання.

Система дозволяє вбудовувати в розробку тестів графічні зображення, таблиці, форматовані тести. Він має кілька різних варіантів запитань:

- Радіо
- Множинний вибір
- Введення текстової відповіді
- Картографування
- Упорядкувати відповіді

Операційна система: Windows 2000/XP/Vista/Server 2003/Server 2008/7
Мова інтерфейсу: російська.

Тип ліцензії: Безкоштовне програмне забезпечення

4. «ІНДІГО» - програма для створення тестів, розташована за адресою <http://indigotech.ru/>

Навчальний тренажер INDIGO - це гнучкий інструмент для розробки комп'ютерних тестів і обробки результатів для наступних завдань:

- Підсумковий контроль для студентів коледжу.
- Претендентів на роботу відбирають потенційні роботодавці.
- Визначення кваліфікаційного рівня співробітників (переміщення, атестація, атестація, кадрові зміни).
- Пройдіть психологічні тести (наприклад, IQ тести).
- Використання в маркетингових опитуваннях, соціологічних дослідженнях
- Шкільні олімпіади, студентські першості тощо. Обґрунтованість програм для створення комп'ютерних тестів INDIGO

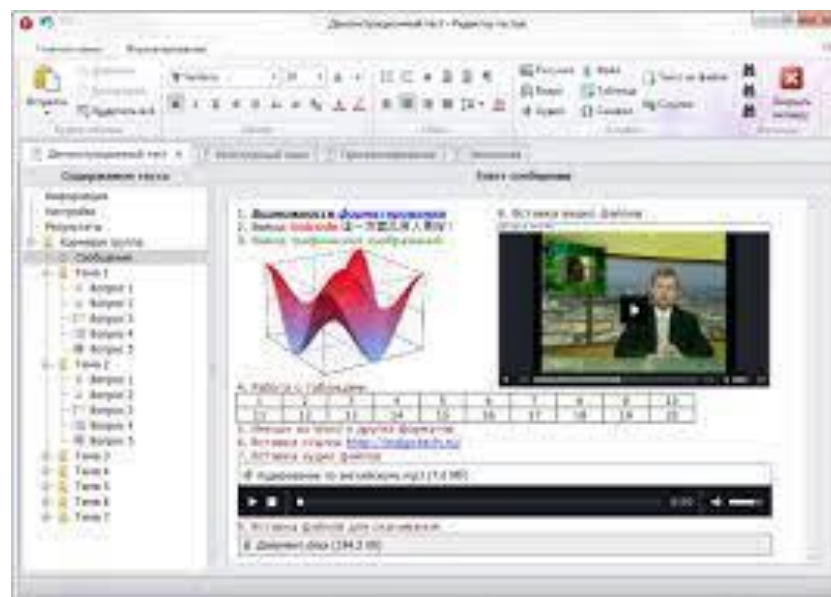


Рис. 2.4 – зображення роботи INDIGO

Забезпечення впровадження та розвитку підвищеного рівня контролю в навчальних закладах за рахунок таких характеристик:

- Ефективно інтегрувати систему в робоче середовище навчальних закладів;
- Доступний та інтуїтивно зрозумілий інтерфейс, що дозволяє швидко та досконало освоїти програму;
- Наявність широкого спектру налаштувань обробки результатів, що значно скорочує час остаточної перевірки підсумкових атестаційних тестів.

INDIGO має такі переваги:

1. Простота і послідовність установки програми;
2. Доступний інтерфейс управління;
3. Веб-інтерфейс користувача;
4. Диференціація отримання інформації;
5. Автоматична генерація варіантів тестів
6. Кілька варіантів конструкторів тестів;
7. Варіативність оцінки результатів тестування
8. Доступність зберігання інформації;
9. Правила структури тесту
10. Роздруківка результатів тесту
11. Експорт протоколу тесту.
12. Ієрархічне групування питань.

Системні вимоги програмного забезпечення

Модулі керування сервером і клієнтом (настільні програми) потребують Windows XP/2003/Vista/2008/7/8.

Користувачам потрібне мережеве підключення до сервера та один із таких браузерів:

Google Chrome 6 і вище, Mozilla Firefox 3.6 і вище, Opera 10.6 і вище, Internet Explorer 9 і вище. Рекомендований браузер Google Chrome (завантажте офлайн-версію інсталятора).

Встановлений і запущений сервер INDIGO 2.0 займає до 100 МБ оперативної пам'яті (з Admin Client до 180 МБ оперативної пам'яті). Споживання

пам'яті збільшується в міру роботи користувача. Для 200 одночасних користувачів рекомендується комп'ютер із принаймні 2 ГБ оперативної пам'яті.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритмізація задачі за темою роботи

Алгоритм - це кінцевий набір правил, що визначає послідовність дій для вирішення певного типу задач. З іншого боку, він має п'ять важливих особливостей [5].

1) Визначеність. Алгоритм завжди повинен завершуватись після завершення кінцевого числа кроків. Кількість кроків може бути як завгодно більшою, вибір занадто великих значень m і n в алгоритмі Евкліда може призвести до того, що деякі кроки будуть виконуватися більше мільйона разів (буде використано будь-яку модифікацію алгоритму, зациклювання або рекурсію).

Процес, що має всі характеристики алгоритму, за винятком можливих закінчень, називається обчислювальним методом. Евклід запропонував алгоритм знаходження найбільшого загального дільника та аналогічну геометричну конструкцію для «найбільшого загального дільника» довжин двох прямих, це вже обчислювальний метод, виконання якого не закінчується, якщо задана довжина непорівнянна.

2) Визначеність. Кожен крок алгоритму має бути точно визначено. Дії, які необхідно зробити, повинні бути чітко визначені для кожної можливої ситуації. На практиці алгоритми можуть бути описані звичайними мовами, формалізованими псевдомовами та мовами програмування. Обчислювальний метод, виражений мовою програмування, називається програмою.

3) Введення. Алгоритм має деякі (можливо рівні нулю) вхідні дані, тобто. значення, задані перед запуском завдання або динамічно визначені під час виконання завдання. Ці вхідні дані беруться із певного набору об'єктів. Наприклад, алгоритм має два вхідних значення, m і n , які належать безлічі позитивних цілих чисел.

4. На закінчення. Алгоритм має одне чи кілька вихідних даних, тобто. значень, які мають дуже специфічне відношення до вхідних даних. Алгоритм Евкліда має лише одне вихідне значення, найбільший спільний дільник двох вхідних значень.

5) Ефективність. Алгоритм зазвичай вважається ефективним, якщо всі його оператори досить прості, щоб їх можна було виконати за допомогою олівця і паперу за обмежений час. Алгоритм Евкліда використовує тільки такі операції: розподіл одного позитивного цілого числа на інше, порівняння з нулем та присвоєння значення однієї змінної іншій.

3.2. Розробка алгоритму роботи тренажеру

Крок 1.

Тренажер починає свою роботу з першого тестового питання, в якому є 4 варіанти відповіді.

Скільки існує парадигм програмування?

1. 3
2. 4
3. 5
4. 6

Після того, як користувач вибрав варіант відповіді необхідно натиснути кнопку «Перевірити», якщо вибраний варіант не правильний, то програма продемонструє відповідне повідомлення на екрані. У випадку коли відповідь правильна, програма продемонструє напис «Вірно».

Після цього користувач повинен натиснути кнопку «Наступне», і тренажер продемонструє вікно з наступним питанням.

Крок 2.

Наступне питання тестового плану, але в ньому користувач повинен обрати 2 варіанти відповіді, в іншому випадку програма продемонструє надпис з помилкою

До гібридних мов програмування відносяться?

1. C++
2. C#
3. Java
4. Object Pascal

Крок 3.

Третє питання тестового плану. В цьому питанні 5 варіантів відповіді, з яких лише 1 правильна. При виборі неправильної відповіді, програма продемонструє надпис з помилкою.

Що вірно до цього фрагмента?


```
For(;;) {
    Console.WriteLine("test");
}
```

1. Код буде скомпільовано, але тіло циклу ніколи не буде виконано.
2. Код буде скомпільовано, оскільки цикл оголошено неправильно.
3. Код буде скомпільовано, але під час його виконання буде видано виняток.
4. Код буде скомпільовано і нескінченно друкувати «тестове» слово.
5. Немає правильної відповіді

Крок 4.

Четверте питання містить поле для вводу відповіді користувачем. У випадку вводу неправильної відповіді програма демонструє надпис з помилкою. Введення відповіді чутливе до регістру. В іншому випадку «Вірно».

```
class Program {
    private static string result;
    static void Main() {
        SaySomething();
        Console.WriteLine(result);
    }
    static async Task<string> SaySomething() {
        await Task.Delay(5);
        result = "Hello world!";
        return "Something";
    }
}
```

Крок 5.

Питання з полем для вводу відповіді користувачем.

Що виведеться на екран при компіляції наступного коду? У випадку вводу неправильної відповіді програма демонструє надпис з помилкою. Введення відповіді чутливе до регістру. В кінці не потрібна крапка. В іншому випадку «Вірно»

```
delegate void Printer();
static void Main()
```

```

{
    List<Printer> printers = new List<Printer>();
    for (int i = 0; i < 10; i++)
    {
        printers.Add(delegate { Console.WriteLine(i); });
    }
    foreach (var printer in printers)
    {
        printer();
    }
}

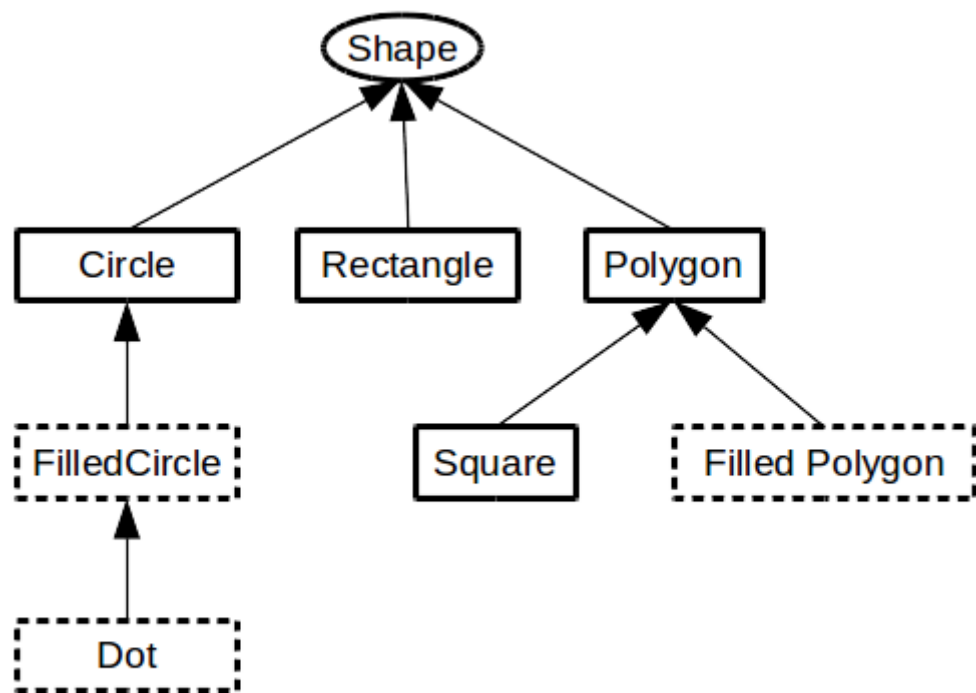
```

Крок 6.

В наступному кроці програма демонструє вікно в якому є схема і 2 поля для вводу. У відповідні поля необхідно ввести Circle та Poligon. Ці поля чутливі до регістру. В кінці не потрібно розділових знаків. Користувач повинен доповнити схему ввівши інформацію у відповідні поля.

У випадку вводу неправильної відповіді програма демонструє надпис з помилкою. В іншому випадку «Вірно».

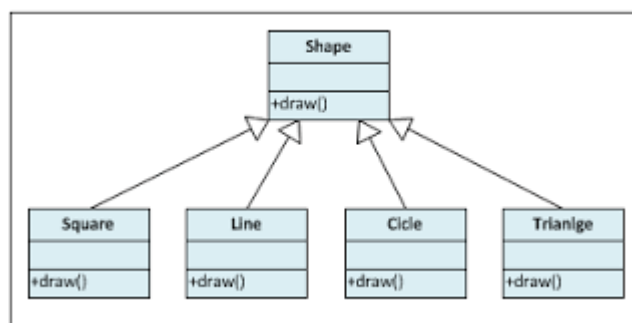
Доповніть схему



Крок 7.

Питання з полем для введення користувачем.

Ієрархія якої структури зображено на схемі. Відповідь чутлива до регістру та розділових знаків. У випадку вводу неправильної відповіді програма демонструє надпис з помилкою. В іншому випадку «Вірно».



Крок 8.

В наступному кроці програма демонструє таблицю, в якій користувач повинен проставити «+» у відповідності з мовами програмування.

Виберіть відповідність (проставити +). Натискаючи на комірку, користувач має змогу змінити її значення. У випадку вводу неправильної відповідності програма демонструє надпис з помилкою. В іншому випадку «Вірно».

Мова	Спец. процесор	Швидкість обробки	Крос-платформен	Простота
C	-	-	-	-
Java	-	-	-	-
Python	-	-	-	-
Assembler	-	-	-	-

Крок 9.

Задача на відповідність (проставити +). Алгоритм працює у відповідності до попереднього кроку.

Блок	Блок введення-виведення	Блок розрахунку	Блок початку(кінець)	Логічний блок
	-	-	-	-
	-	-	-	-
	-	-	-	-
	-	-	-	-
				

Крок 10.

Питання з полем для вводу відповіді користувачем.

Програма демонструє вікно для вводу відповіді користувачем. Користувачу необхідно ввести код, який виводить у консолі «Hello world!». Відповідь чутлива до регістру та розділових знаків. У випадку вводу неправильної відповідності програма демонструє надпис з помилкою. В іншому випадку «Вірно».

Напишіть код який виводить у консоль Hello world!

Крок 11.

Питання з полем для вводу відповіді користувачем. Відповідь чутлива до регістру та розділових знаків. У випадку вводу неправильної відповідності програма демонструє надпис з помилкою. В іншому випадку «Вірно».

Напишіть код для зчитування числа у змінну num.

А потім виведіть останню цифру цього числа.

3.3 Ознайомлення з середовищами для розробки програмного забезпечення

Середовища розробки ПО (програмного забезпечення) є об'єднанням програмних засобів, які призначені для написання (створення) програмних продуктів. Середовище розробки включає в свій зміст: компілятор, інтерпретатор, відладчик, засоби автоматизації збирання, а також редактор тексту.

Компілятор - це така програма, яка зчитує вихідні коди, написані програмістом і перетворює ці коди в програму.

Інтерпретатор - це програма яка зчитує команди, що знаходяться в вихідних кодах, відразу виконуючи їх.

Коли в середовищі розробки ПО присутні всі вищеназвані компоненти, тоді таке середовище називають інтегрованою. Такі середовища розробки збільшують темп, а також зручність розробки за рахунок: автоматизації, можливості виробляти весь цикл створення і розробки ПЗ.

Зазвичай середовище розробки ПО призначена для розробки тільки на одній мові програмування. А таке середовище розробки як інтегрована, надає право вибрати творцеві програми мову програмування для розробки, зручний розробнику (з мов підтримуваних цим середовищем). Прикладом цього є: Visual Studio, Komodo, Geany, Kylix, NetBeans, Eclipse.

Microsoft Visual Studio - одна з інтегрованих середовищ розробки, розроблена на C ++ і C #, підтримується Windows OS. Дане середовище розробки

переведена на десять мов (також і на російську мову). У Visual Studio творець може вести розробку веб-сайтів, веб-служб, писати консольні додатки, а також додатки з графічним інтерфейсом. Також VS підтримує різного роду доповнень. Найзнаменитіші доповнення - це ReSharper (виконує пошук помилок в коді під час написання коду програми розробником, до компіляції); Visual Assist (на відміну від ReSharper підтримує також і C++); AnkhSVN (використовує в Visual Studio систему контролю версій, яка носить назву Subversion).

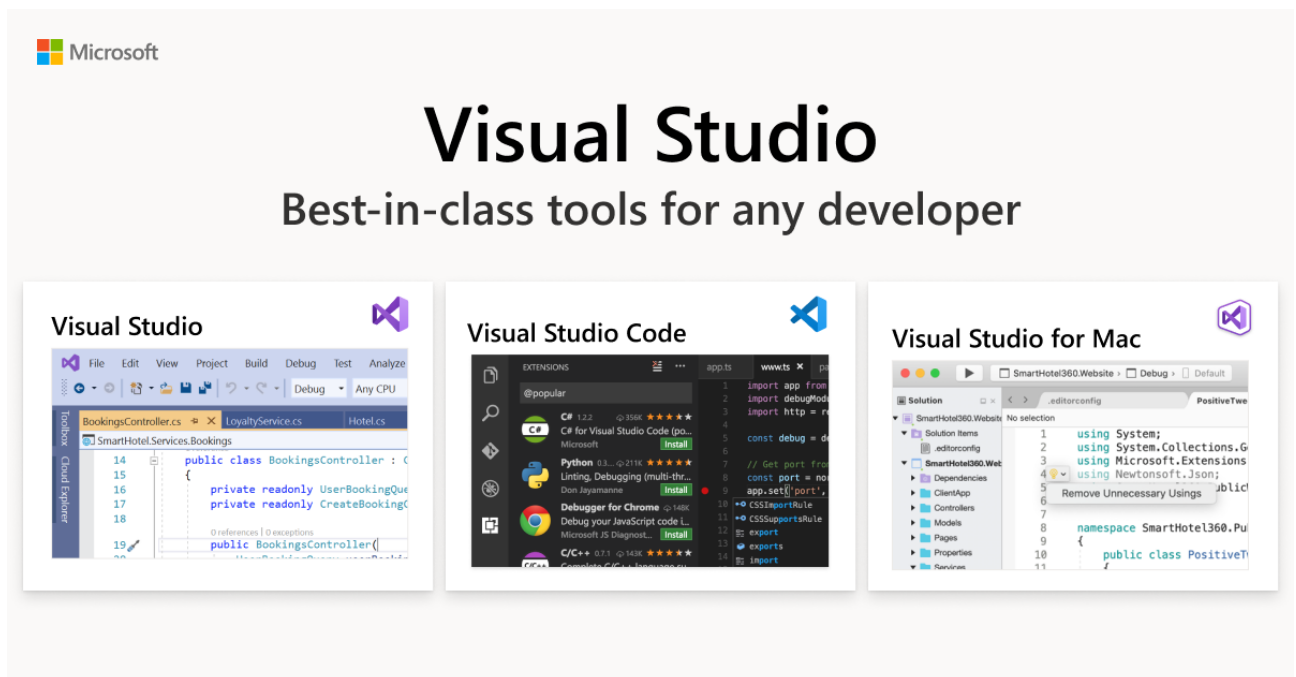


Рис. 3.1 – Visual Studio

Переваги: Зрозумілий інтерфейс середовища розробки, зручність, автоматичне виявлення помилок в коді.

Недоліки: Складно для початківців програмістів.

Середовище особливо поширена в англomовних країнах, Росії, Китаї, Німеччині, Франції, Португалії, Італії, Японії, Іспанії та Кореї.

Geany також інтегроване середовище розробки програмного забезпечення. Підтримується на ОС Linux, а також на Mac OS і на Windows. Працює з тридцятьма двома мовами (також і з російською мовою). У складі Geany відсутня

компілятор. Компілятор можна встановити як доповнення. Підтримує досить багато мов програмування, серед яких присутні класичний C, C++ і C#.

Переваги: Простота і зручність, підсвічування вихідного коду, можливість підключати доповнення.

Недоліки: не включає до свого складу компілятор.

Серед розповсюджувалися в багатьох країнах (Більш ніж у тридцяти).

Komodo або Active State Komodo - була написана на JavaScript, XUL, Python. Інтерфейс даного середовища тільки на англійською мовою. Працює на тих же операційних системах як Geany: на ОС Linux, Windows і Mac OS.



Рис. 3.2 – Active State Komodo

Підтримує десять мов програмування, серед які присутні: PHP, Ruby, HTML5.

Переваги: Доповнення Code Explorer дозволяє переглядати об'єктне дерево скрипта або бібліотеки, середовище є кроссплатформеним, зручний відладчик з

можливістю вилученого налагодження, можливість налаштувати інтерфейс середовища «під себе».

Недоліки: Висока вартість, підтримує мало мов програмування, сильно завантажує комп'ютер (а саме оперативну пам'ять), є складним для розуміння.

Поширена в основному в англомовних країнах.

Kylix - інтегроване середовище. Функціонує на OS Linux. Працює з C, C++ і ObjectPascal.

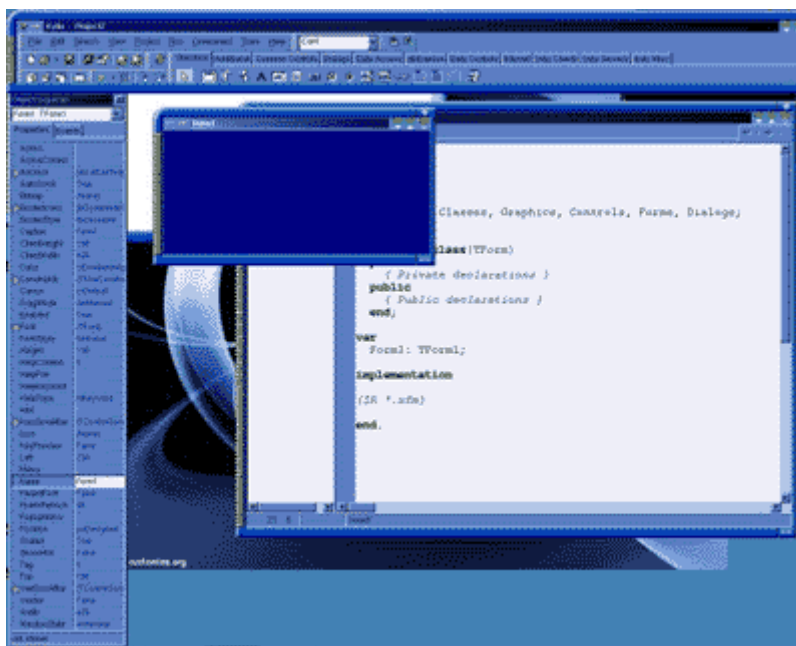


Рис. 3.3 - Kylix

В даному середовищі є можливість писати програми веб-служб.

Kylix випускався в трьох пакетах. Ці пакети: Enterprise Edition - включав в себе сто дев'яносто компонентів (був найбільшим і самим дорогим пакетом програми); Professional Edition (дешевший варіант, який включав в себе близько 165 компонентів); Open Edition - безкоштовний пакет програми, що містить в собі 75 компонентів, в ньому відсутній кошти для роботи з базами даних.

Оновлена версія Kylix 2, на відміну від Kylix працювала набагато швидше. Наприклад, Kylix 1 здійснював сортування бульбашкою масиву з 115 елементів півтори хвилини, Kylix 2 - одну секунду.

У 2002 році дане середовище розробки припинив підтримувати розробник.

Переваги: Зручний в перенесенні написаного з однієї операційної системи на іншу.

Недоліки: Дане середовище більше не підтримується розробником.

Поширена в основному в Європейських країнах і США, через те що розробник (Borland) перестав підтримувати Kylix - стає все менш популярною і не затребуваною.

Netbeans - інтегроване середовище розробки програмного забезпечення. Була реалізована на програмному мовою Java. Це середовище розробки високої якості. Вміє працювати на декількох операційних системах, тобто є кроссплатформним. Працює більш ніж з п'ятьма програмними мовами.

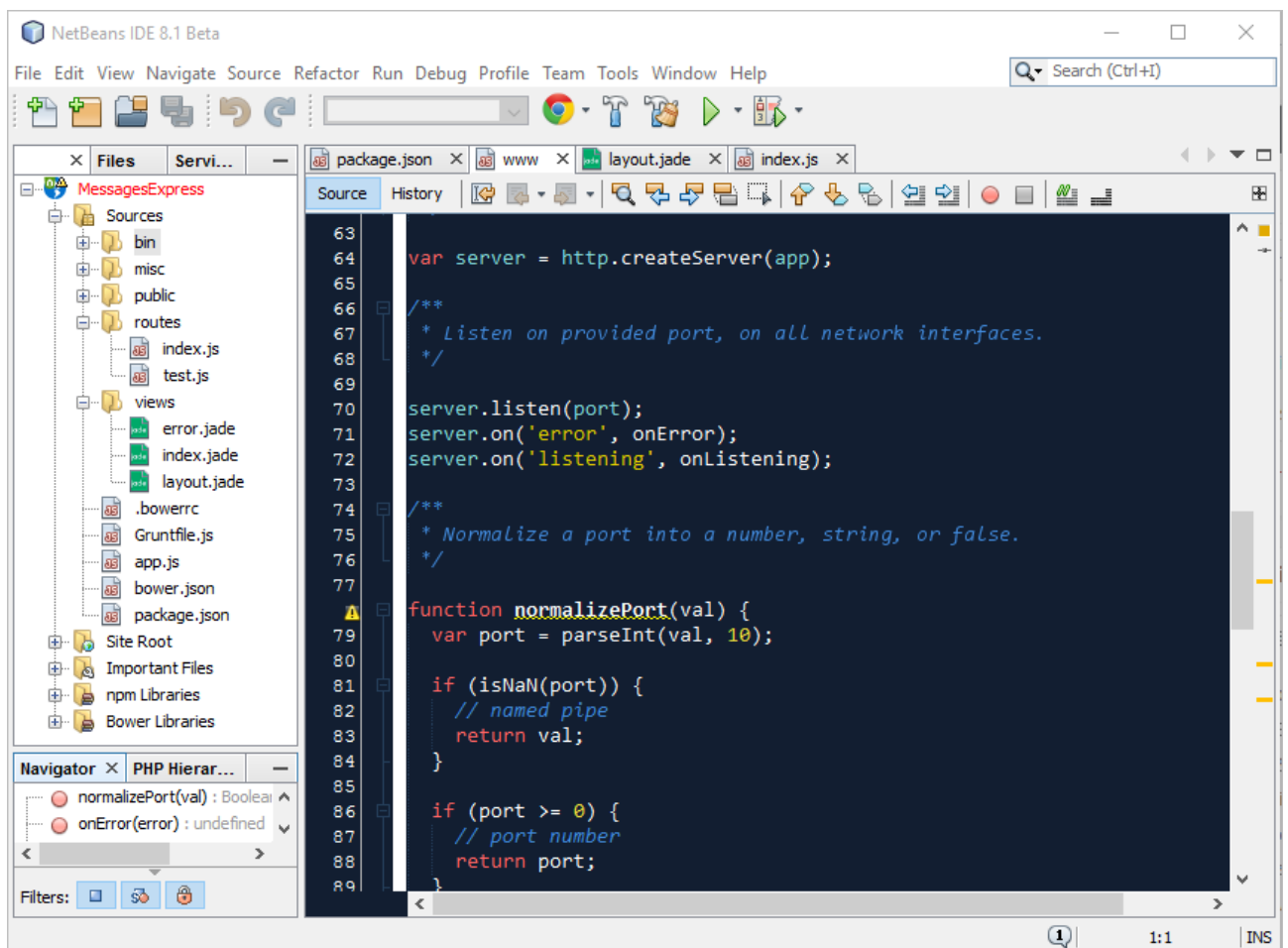


Рис. 3.4. - Netbeans

Переваги: Є безкоштовною, присутній система контролю версій, підсвічування синтаксису, можливо перейменовувати змінну / клас одним кліком, в тому випадку якщо вручну перейменовувати занадто довго (автоматизоване перейменування), є можливість форматування коду по CodeStyle, розробником середовище постійно вдосконалюється, поліпшується.

Недоліки: Часом в середовищі розробки виникають проблеми з кодуванням, довгий запуск програми.

Поширена в багатьох країнах, в силу того що є зручною і безкоштовною.

Eclipse - ще одна інтегрована середовище розробки ПО. Написана на мові Java в дві тисячі третьому році. Також є кроссплатформенной. За рахунок приєднуються до цього середовища доповнень - є можливість створювати програмні продукти більш ніж на п'яти мовах програмного коду.

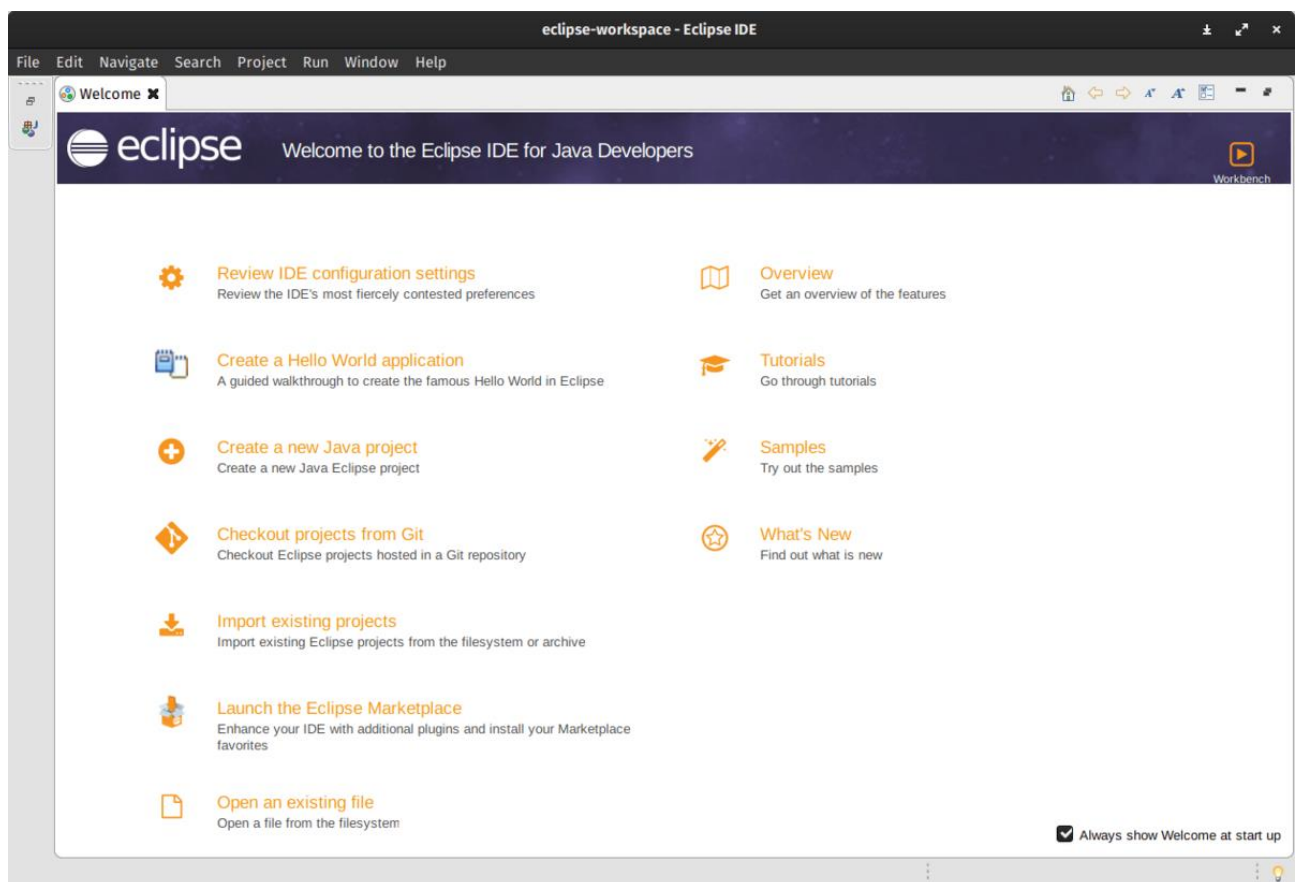


Рис. 3.5 - Eclipse

Переваги: Постійне оновлення версій середовища розробки, підтримка багатьох мов (включно з російською), безкоштовно, підтримка багатьох мов програмування, середовище має промисловий рівень, гнучка - що легко налаштовується як під будь-яку платформу, так і під будь-який користувач.

Недоліки: Сильно навантажує ОЗУ комп'ютера, довго запускається, проте якщо комп'ютер досить потужний - ця проблема легко вирішується.

Поширений у багатьох країнах, популярний.

З перелічених середовищ для створення програмних продуктів ми обрали середовище програмування Microsoft Visual Studio.

3.4 Вимоги до технічного забезпечення

ОС і середовище швидко змінювалися з останніх 30 років, що робить речі застарілими, перш ніж вони зможуть стати продуктивними протягом свого життя. Наприклад, Microsoft випустила понад 20 основних випусків ОС за останні 25 років. Кожна мова та середовище випускають новий випуск кожні шість місяців. Коли ОС або її середовище змінюються, це має ефект пульсації, що призводить до застарілих програм. Як ми можемо зупинити таке поширення продуктів і технологій і зробити програми стабільними? Не забезпечуючи жодної перевірки, одним із можливих підходів є уникнення операційної системи та її середовища. У деяких областях обчислень це робиться повільно, непомітно.

Після завантаження ПК він перебуває в реальному режимі, де адресація обмежена 1 МБ і немає захисту вашого коду. Це просто в режимі дискової операційної системи (DOS). Щоб завантажувати та запускати більші програми та мати доступ до більшої пам'яті, вам потрібно навчитися переходити в захищений режим за допомогою інструкцій на мові асемблера. Однак усі виклики базової системи введення/виводу (BIOS) доступні лише в реальному режимі. Якщо ви хочете використовувати BIOS для вводу-виводу, вам потрібно мати механізми

перемикання із захищеного режиму в реальний для виконання викликів BIOS. Таким чином, ваша програма або якась керуюча програма повинні виконувати це перемикання, прозоре для користувача. Якщо ви плануєте використовувати програмні переривання (їх 255) замість переривань BIOS, то вам потрібно написати власний код на зборці для вирішення всіх операцій введення-виводу. Ми використовували як BIOS, так і програмні переривання і написали перемикач захищеного режиму в режим реального для роботи в обох режимах. Це нетривіальна річ, яку потрібно зробити під час складання, яка вимагає від вас ознайомлення з документом специфікації архітектури Intel, який доступний на їх веб-сайті. Цей документ є корисним ресурсом для розуміння та реалізації переривань, схем адресації, засобів завдань, пасток і винятків.

Тестування проводиться зазвичай у кількох етапах життєвого циклу, коли розробляється програмне забезпечення. Технологія тестування програмного забезпечення містить такі етапи:

- визначення функціоналу, що підлягає і не підлягає тестуванню;
- Формулювання підходів, що використовуються для даного продукту;
- Написання тест кейсів;
- розробка критерію проходження тестів;
- Визначення вимог середовища проведення тестування;
- проведення тестування та оцінка результатів;
- Звітність результатів.

Системні вимоги:

- операційна система Windows 7 чи наступних версій;

Апаратні вимоги:

- Досить вільної пам'яті комп'ютера для установки програми;
- 32 або 64-розрядний процесор із тактовою частотою 1GHz;
- Оперативна пам'ять не менше 1Gb.

Тестування проводилося на ноутбучі компанії DELL із процесором Intel Core i3-6060U, тактовою частотою 2GHz, операційною системою Windows 10.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис процесу розробки програми

Розроблена програма в рамках даної дипломної роботи навчальний тренажер для тестування студентів є багаторівневою прикладною програмою.

Обробка даних у таких системах здійснюється за допомогою елементів, розбитих на кілька логічних секцій. Розділення компонентів програми на різні рівні може підвищити робочу надійність і масштабованість програми. Це досягається завдяки можливості застосування нових технологій до будь-якого рівня без зміни всього рішення. Архітектура типової багаторівневої програми включає рівень презентації, середній рівень і рівень даних. Середній рівень зазвичай включає рівень доступу до даних, рівень бізнес-логіки та загальні компоненти, такі як автентифікація та інші засоби перевірки. Рівень даних включає реляційні бази даних. Багаторівневі програми зазвичай зберігають конфіденційну інформацію на рівні доступу до даних середнього рівня, щоб підтримувати ізоляцію від кінцевих користувачів, які отримують доступ до рівня презентації.

Вид дизайну.

Навчальний тренажер реалізує такі функції:

- аутентифікація та керування користувачами;
- тестування користувача;
- збір статистичних даних;
- отримати сертифікат;
- опублікувати результати здачі тестів.

Тестовою системою є клієнт (тонкий клієнт)/сервер з трирівневою архітектурою.

Таким чином, розроблена система має рівень сервера, рівень клієнта та рівень бази даних.

Клієнтська частина (рівень 1) відповідає за представлення даних користувачеві. Користувачеві достатньо мати програму-браузер для взаємодії з тестовою системою. За допомогою програми-браузера користувач взаємодіє з сервером бази даних і прикладною програмою через мережу.

Бізнес-логіка (другий рівень) і рівень обробки даних відповідають за основну логіку додатків і логіку обробки даних. На цьому рівні знаходиться додаток тестової системи.

Сервер (рівень 3) відповідає за надання даних клієнтам.

Представляє інтерфейс користувача та основну бізнес-логіку програми керування послугами та забезпечує доступ до даних незалежно від їх розташування. Додаток встановлюється на сервер разом з базою даних і взаємодіє з користувачем через браузер.

Незалежність від платформи.

Оскільки операційна система Windows встановлена практично на всіх існуючих обчислювальних платформах, навчальна система гарантує взаємодію з користувачем, незалежно від платформи.

Подальший розвиток навчального тренажеру не вимагатиме зміни іншої операційної системи користувача або іншого інтерфейсу користувача.

Розширюваність.

Ця техніка економить час, витрачений на розгортання програми, розміщуючи всю функціональність програми на окремому сервері та видаляючи її з клієнтських програм. У той же час це спрощує оновлення системи та управління при роботі з різними обчислювальними платформами.

Додаток використовує внутрішній контейнер і зовнішнє сховище для зберігання результатів. Внутрішній контейнер призначений для зберігання таких речей:

- запитання та альтернативні відповіді
- графічні об'єкти та малюнки;

Цей сценарій демонструє роботу навчального тренажеру, її взаємодію з програмним продуктом, встановленим на сервері.

В програмі також було реалізовано перемикач мови, на вибір було: українська чи англійська.

4.2 Проектування та розробка БД

У системі керування контентом застосовується СУБД MySQL. База даних складається з таблиці quiz.

Усі дані зберігаються на локальному сервері в окремих базах даних

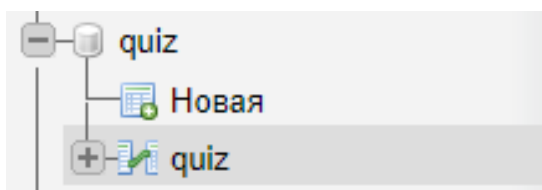


Рис. 4.1 - Структура БД(Загальна)

Нижче описано структуру БД, яка була створена для додатку

+ Options			id	fotourl	surname	name	test1	test2	test3	test4	test5	test6	test7	test8	test9	test10	test11	login	pass
☐	Edit	Copy	Delete	1	1.jpg	Іванов	Іван	1	1	1	1	1	1	0	0	0	0	1	1
☐	Edit	Copy	Delete	2	2.jpg	Петров	Петр	0	0	1	0	0	0	0	0	0	0	2	2
☐	Edit	Copy	Delete	3	3.jpg	Демідова	Тетяна	0	1	0	0	0	0	0	0	0	0	3	3
☐	Edit	Copy	Delete	4	4.jpg	Неандерталець	Неандерталець	0	1	1	0	0	0	0	0	0	0	4	4
☐	Edit	Copy	Delete	5	5.jpg	Сидоренко	Максим	1	1	1	0	0	0	0	0	0	0	5	5
☐	Edit	Copy	Delete	6	10.jpg	Іванов	Іван	1	1	0	0	0	0	0	0	0	0	6	6
☐	Edit	Copy	Delete	7	9.jpg	Іванов	Іван	1	0	1	0	0	0	0	0	0	0	7	7
☐	Edit	Copy	Delete	8	10.jpg	Іванов	Іван	0	0	0	0	0	0	0	0	0	0	10	10

Рис. 4.2 - Структура БД(quiz)

Підключення до бази даних здійснюється за допомогою функції

```
class DB
{
    MySqlConnection connection = new
    MySqlConnection("server=localhost; port = 3306; username=root;
    password=root; database=quiz");
```

```

        public void openConnection()
        {
            if (connection.State ==
System.Data.ConnectionState.Closed)
                connection.Open();
        }

        public void closeConnection()
        {
            if (connection.State ==
System.Data.ConnectionState.Open)
                connection.Close();
        }

        public MySqlConnection getConnection()
        {
            return connection;
        }
    }

```

4.3. Опис роботи програми

Для запуску програми необхідно перейти наступним шляхом і запустити програму quiz (рис. 4.3)

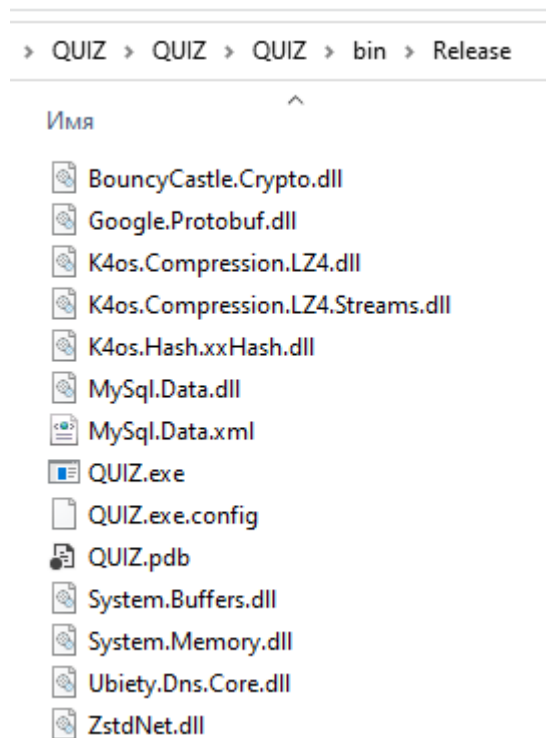


Рис. 4.3 - Шлях до файлу

Запустивши програму, перед вами з'явиться наступне вікно (рис. 4.4)

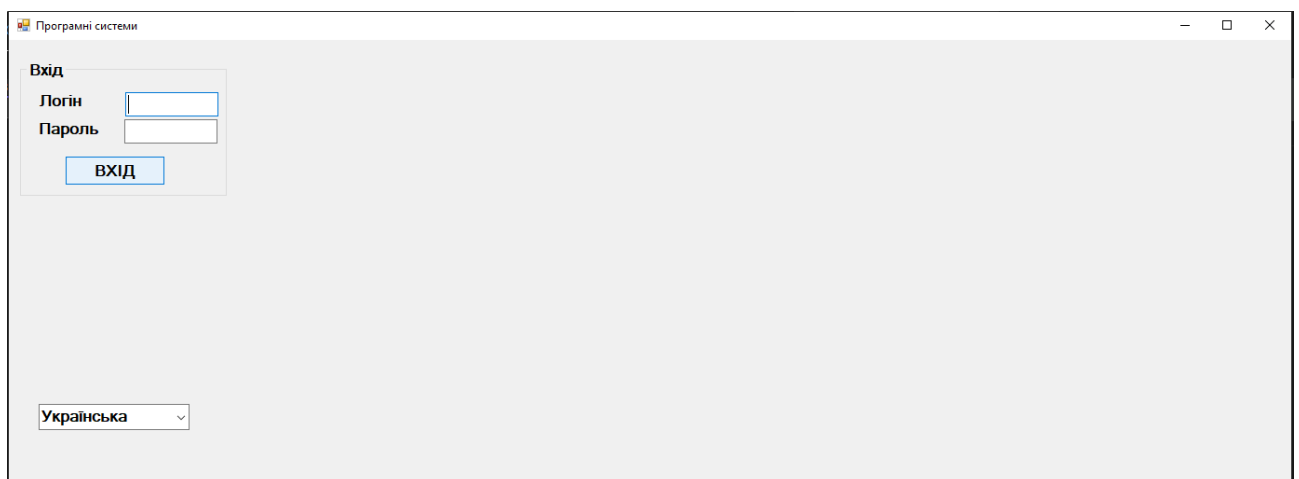


Рис. 4.4 – Стартове вікно програми

Ввівши логін та пароль, ви можете увійти як звичайний користувач (рис. 4.5) так і як адміністратор (рис. 4.6).

Програмні системи

Вхід

Логін: 1

Пароль: 1

ВХІД

Пошук

Прізвище: Іванов

Пошук

Українська

Тести Рейтинг

1. Скільки існує парадигм програмування?

☐ 3

☐ 4

☐ 5

☐ 6

НАСТУПНЕ ПЕРЕВІРИТИ ВИЙТИ

Рис. 4.5.1 – Вид програми звичайного користувача

Програмні системи

Вхід

Логін: 1

Пароль: 1

ВХІД

Пошук

Прізвище: Іванов

Пошук

Українська

Тести Рейтинг

№	Зображення	Прізвище	Ім'я	Тест1	Тест2	Тест3
1		Іванов	Іван	1	1	1
2		Петров	Петр	0	0	1
3		Демідова	Тетяна	0	1	0
4		Неандерталець	Неандерталець	0	1	1
5		Сидоренко	Максим	1	1	1
6		Іванов	Іван	1	1	0

Оновити

Рис. 4.5.2 – Вид програми звичайного користувача

Програмні системи

Вхід

Логін: 12345

Пароль: 12345

ВХІД

Пошук

Прізвище: Іванов

Пошук

Українська

Тести Рейтинг

№	Зображення	Прізвище	Ім'я	Тест1	Тест2	Тест3
1		Іванов	Іван	1	1	1
2		Петров	Петр	0	0	1
3		Демідова	Тетяна	0	1	0
4		Неандерталець	Неандерталець	0	1	1
5		Сидоренко	Максим	1	1	1
6		Іванов	Іван	1	1	0

Оновити

Додати користувача

Логін: 10

Пароль: 10

Зображення: 10.jpg

Прізвище: Іванов

Ім'я: Іван

Додати

Видалити за номером

Видалити №: 2

Видалити

Рис. 4.6 – Вид програми адміністратора

Для пошуку користувача необхідно ввести прізвище, і пошукова система знайде відповідних клієнтів (рис. 4.7)

The screenshot shows a web application window titled 'Програмні системи'. It contains several sections:

- Вхід (Login):** Fields for 'Логін' (12345) and 'Пароль' (12345), and a 'ВХІД' button.
- Пошук (Search):** A 'Прізвище' field with 'Іванов' entered and a 'Пошук' button.
- Тести / Рейтинг:** A table with columns: №, Зображення, Прізвище, Ім'я, Тест1, Тест2, Тест3. It lists four users with the surname 'Іванов'.
- Додати користувача (Add user):** Fields for 'Логін' (10), 'Пароль' (10), 'Зображення' (10.jpg), 'Прізвище' (Іванов), and 'Ім'я' (Іван), with a 'Додати' button.
- Видалити за номером (Delete by number):** A 'Видалити №' field with '2' entered and a 'Видалити' button.

№	Зображення	Прізвище	Ім'я	Тест1	Тест2	Тест3
1		Іванов	Іван	0	1	0
6		Іванов	Іван	1	1	0
7		Іванов	Іван	1	0	1
8		Іванов	Іван	0	0	0

Рис. 4.7 – пошук користувача

Для видалення клієнта з бази даних необхідно ввести номер користувача, який хочете видалити у відповідне поле (рис. 4.8)

This is a close-up of the 'Видалити за номером' section. It features a label 'Видалити №' followed by a text input field containing the number '2'. Below the input field is a grey button labeled 'Видалити'.

Рис. 4.8 – Видалення користувача

Для додавання користувача необхідно заповнити поля та натиснути на кнопку «Додати» (рис. 4.9)

Додати користувача

Логин	10
Пароль	10
Зображення	10.jpg
Прізвище	Іванов
Ім'я	Іван

Додати

Рис. 3.9 – Додавання клієнта

4.4. Дослідження можливостей програмної реалізації

Під час тестування, всі помилки, що виникали, були виправлені. Нижче наведено приклади тестування програми.

Програмні системи

Enter

Login 1

Password 1

Enter

Search

Surname Ivanov

Search

English

Tests Rating

2. Hybrid programming languages include

☒ C++

☐ C#

☒ Java

☐ Object Pascal

NEXT CHECK EXIT Almost right

Рис. 4.10 – тестування програми

Enter

Login1

Password1

Enter

Search

SernameIvanov

Search

English

TestsRating

5. What will be displayed on the screen when the following code is compiled?

```

delegate void Printer()

static void Main()
{
    List<Printer> printers=new List<Printer>();
    for (int i=0; i<10; i++)
    {
        printers.Add(delegate{Console.WriteLine(i)});
    }

    foreach (var printer in printers)
    {
        printer();
    }
}

```

0123456789

NEXT

CHECK

EXIT

True

Рис. 4.11 – тестування програми

Enter

Login1

Password1

Enter

Search

SernameIvanov

Search

English

TestsRating

№	Image	Sername	Name	Test1	Test2	Tes
1		Іванов	Іван	1	1	1
2		Петров	Петр	0	0	1
3		Демідова	Тетяна	0	1	0
4		Неандерталець	Неандерталець	0	1	1
5		Сидоренко	Максим	1	1	1
6		Іванов	Іван	1	1	0

Refresh

Рис. 4.12 – тестування програми

ВИСНОВКИ

В рамках виконання дипломної роботи побудовано алгоритм та створено програмну реалізацію навчального тренажеру з теми «Методологія програмування» англomовного дистанційного навчального курсу «Теорія програмування».

Для досягнення мети були реалізовані наступні завдання:

- опрацьовано методичні рекомендації до виконання дипломної роботи;
- сформульовано постановку задачі;
- виконано огляд літературних джерел, що використовується для вивчення даної теми, а також здійснено огляд існуючих програмних продуктів, які використовуються при вивченні різних дисциплін;
- опрацьовано теоретичний матеріал з теми;
- розроблено алгоритм навчального тренажеру;
- побудовано блок–схеми алгоритму тренажеру;
- здійснено вибір оптимальної платформи для програмної реалізації тренажеру;
- реалізована програма-тренажер;
- здійснено опис програми.

Для створення програми було обрано мову програмування C# у середовищі Visual Studio 2022. Даний тренажер впроваджено для вивчення теми «Методологія програмування» англomовного дистанційного навчального курсу «Теорія програмування» в Полтавському університеті економіки та торгівлі.

Результати дипломної роботи опубліковані у Збірнику наукових статей магістрів Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі»

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабій М. С. Теорія програмування. Суми: Вид-во СумДУ, 2009. 181 с.
2. Вінник В. Ю. Алгоритмічні мови та основи програмування: мова С. Житомир: ЖДТУ, 2007. 328 с.
3. Дейтел Х. М. Как программировать на С++. М.: Бином-Пресс, 2005. 1248 с.
4. Зыков С. В. Современные языки программирования. Ч. I. Функциональный подход к программированию. М.: МИФИ, 2003. 230 с.
5. Ковалюк Т. В. Алгоритмізація та програмування: підручник з грифом МОН України. Львів: Магнолія-2006, 2013. 400 с.
6. Мартынов Н. Н. Программирование для Windows на С/С++ М.: Бином-Пресс, 2004. 528 с.
7. Нікітченко М. С. Теоретичні основи програмування. Київ: КНУ ім. Т. Г. Шевченка, 2009. 200 с.
8. Пышкин Е. В. Структуры данных и алгоритмы: реализация на С/С++. СПб.: ФТК СПбГПУ, 2009. 200 с.
9. Трофименко О. Г. С++. Теорія та практика. 587 с.
10. Макконнелл, З. Ідеальний код. Майстер-клас/С. Макконнелл. Пров. з англ. - М.: Видавництво «Російська редакція», 2010. - 896 стор.
11. Документація Visual Studio [електронний ресурс]. Режим доступу: <https://docs.microsoft.com/ru-ru/visualstudio/ide/?view=vs-2017>
12. Введення мови програмування С і С++ [електронний ресурс]/режим доступу: <http://www.intuit.ru/studies/courses/1039/231/info>
13. Верифікація програмного забезпечення [електронний ресурс]/режим доступу: <http://www.intuit.ru/studies/courses/1040/209/info>
14. Інструменти, алгоритми та структури даних [електронний ресурс] / режим доступу: <http://www.intuit.ru/studies/courses/683/539/info>

ДОДАТКИ

Додаток А(DB.cs)

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace library
{
    class DB
    {
        MySqlConnection connection = new
        MySqlConnection("server=localhost; port = 3306; username=root;
        password=root; database=quiz");

        public void openConnection()
        {
            if (connection.State ==
            System.Data.ConnectionState.Closed)
                connection.Open();
        }

        public void closeConnection()
        {
            if (connection.State ==
            System.Data.ConnectionState.Open)
                connection.Close();
        }

        public MySqlConnection getConnection()
        {
            return connection;
        }
    }
}
```

Додаток Б(Form1.cs)

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```



```

using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QUIZ
{
    public partial class Form1 : Form
    {
        int count = 0;
        string login;
        string password;
        string t1 = "0";
        string t2 = "0";
        string t3 = "0";
        string t4 = "0";
        string t5 = "0";
        string t6 = "0";
        string t7 = "0";
        string t8 = "0";
        string t9 = "0";
        string t10 = "0";

        public Form1()
        {
            InitializeComponent();
            if (comboBox1.SelectedIndex == 0)
            {
                label1.Text = "1/10. Обрати правильне визначення  

                дефініції «Програмні системи».";
                radioButton1.Text = "це ідентична форма фіксації  

                програмних понять яка дозволяє визначити різницю понять," +  

                Environment.NewLine + "що задаються сукупністю маси масуджень.";
                radioButton2.Text = "це специфічна форма  

                фіксації рівня абстракції програмних понять. Вона дозволяє визначити  

                сукупність понять," + Environment.NewLine + "яка характеризує  

                рівень, та їх властивості, що задаються сукупністю суджень.";
                radioButton3.Text = "це нейтральна форма  

                фіксації рівня абстракції програмних понять.";
                label2.Text = "";
            }
            if (comboBox1.SelectedIndex == 1)
            {
                label1.Text = "1/10. Choose the correct  

                definition of the definition «Software systems».";
                radioButton1.Text = "this is an identical form  

                of fixation of program concepts that allows to determine the  

                difference of concepts," + Environment.NewLine + "which are given by  

                the set of mass masses.";
                radioButton2.Text = "this is a specific form of  

                fixing the level of abstraction of program concepts. It allows you

```

```

to define a set of concepts," + Environment.NewLine + "which
characterizes the level and their properties given by a set of
judgments.";
        radioButton3.Text = "it is a neutral form of
fixing the level of abstraction of program concepts.";
        label2.Text = "";
    }
}

private void button4_Click(object sender, EventArgs e)
{
    tabControll1.Visible = true;
    login = textBox1.Text;
    password = textBox2.Text;
    count = 0;
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "1/10. Обрати правильне визначення
дефініції «Програмні системи».";
        radioButton1.Text = "це ідентична форма фіксації
програмних понять яка дозволяє визначити різницю понять," +
Environment.NewLine + "що задаються сукупністю маси масуджень.";
        radioButton2.Text = "це специфічна форма
фіксації рівня абстракції програмних понять. Вона дозволяє визначити
сукупність понять," + Environment.NewLine + "яка характеризує
рівень, та їх властивості, що задаються сукупністю суджень.";
        radioButton3.Text = "це нейтральна форма
фіксації рівня абстракції програмних понять.";
        label2.Text = "";
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "1/10. Choose the correct
definition of the definition «Software systems».";
        radioButton1.Text = "this is an identical form
of fixation of program concepts that allows to determine the
difference of concepts," + Environment.NewLine + "which are given by
the set of mass masses.";
        radioButton2.Text = "this is a specific form of
fixing the level of abstraction of program concepts. It allows you
to define a set of concepts," + Environment.NewLine + "which
characterizes the level and their properties given by a set of
judgments.";
        radioButton3.Text = "it is a neutral form of
fixing the level of abstraction of program concepts.";
        label2.Text = "";
    }

    if (textBox1.Text == "12345" && textBox2.Text ==
"12345")
    {
        groupBox2.Visible = true;
    }
}

```

```

        groupBox3.Visible = true;
        groupBox4.Visible = true;
    }
    else
    {
        groupBox2.Visible = false;
        groupBox3.Visible = false;
        groupBox4.Visible = true;
    }
    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);

    dataGridView1.Rows.Clear();

    dataGridView1.RowCount = table.Rows.Count;
    for (int i = 0; i < table.Rows.Count; ++i)
    {
        Image image = Image.FromFile("../\\..\\img\\" +
table.Rows[i].ItemArray[1].ToString());

        dataGridView1.Rows[i].Cells[0].Value =
table.Rows[i].ItemArray[0].ToString();
        dataGridView1.Rows[i].Cells[1].Value = image;
        dataGridView1.Rows[i].Cells[2].Value =
table.Rows[i].ItemArray[2].ToString();
        dataGridView1.Rows[i].Cells[3].Value =
table.Rows[i].ItemArray[3].ToString();
        dataGridView1.Rows[i].Cells[4].Value =
table.Rows[i].ItemArray[4].ToString();
        dataGridView1.Rows[i].Cells[5].Value =
table.Rows[i].ItemArray[5].ToString();
        dataGridView1.Rows[i].Cells[6].Value =
table.Rows[i].ItemArray[6].ToString();
        dataGridView1.Rows[i].Cells[7].Value =
table.Rows[i].ItemArray[7].ToString();
        dataGridView1.Rows[i].Cells[8].Value =
table.Rows[i].ItemArray[8].ToString();
        dataGridView1.Rows[i].Cells[9].Value =
table.Rows[i].ItemArray[9].ToString();
        dataGridView1.Rows[i].Cells[10].Value =
table.Rows[i].ItemArray[10].ToString();
        dataGridView1.Rows[i].Cells[11].Value =
table.Rows[i].ItemArray[11].ToString();
    }

```

```

        dataGridView1.Rows[i].Cells[12].Value           =
table.Rows[i].ItemArray[12].ToString();
        dataGridView1.Rows[i].Cells[13].Value           =
table.Rows[i].ItemArray[13].ToString();
    }

}

private void button3_Click_1(object sender, EventArgs e)
{

    if (comboBox1.SelectedIndex == 0)
        label2.Text = "НеВipHO";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Wrong";
    label2.ForeColor = Color.Red;

    if (radioButton3.Checked && count == 0)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t1 = "1";
    }
    if (radioButton3.Checked && count == 1)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t2 = "1";
    }
    if (radioButton1.Checked && count == 2)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t3 = "1";
    }
    if (radioButton2.Checked && count == 3)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t4 = "1";
    }
}

```

```

}
if (radioButton1.Checked && count == 4)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "Bipho";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Right";
    label2.ForeColor = Color.Green;
    t5 = "1";
}
if (radioButton3.Checked && count == 5)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "Bipho";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Right";
    label2.ForeColor = Color.Green;
    t6 = "1";
}
if (radioButton1.Checked && count == 6)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "Bipho";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Right";
    label2.ForeColor = Color.Green;
    t7 = "1";
}
if (radioButton3.Checked && count == 7)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "Bipho";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Right";
    label2.ForeColor = Color.Green;
    t8 = "1";
}
if (radioButton1.Checked && count == 8)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "Bipho";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Right";
    label2.ForeColor = Color.Green;
    t9 = "1";
}
if (radioButton2.Checked && count == 9)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "Bipho";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Right";
}

```

```

        label2.ForeColor = Color.Green;
        t10 = "1";
    }
}

private void button1_Click_1(object sender, EventArgs e)
{
    count++;
    if (count == 1)
    {
        if (comboBox1.SelectedIndex == 0)
        {
            label1.Text = "2/10. Функціонально  
номінативна система:";
            radioButton1.Text = "алгебра даних: V -  
носіє цієї алгебри, які задаються іменами з K за допомогою  
відображення dec";
            radioButton2.Text = "алгебра функцій  
задаються іменами з Z за допомогою відображення bek";
            radioButton3.Text = "алгебра даних: D -  
носіє цієї алгебри, а операціями алгебри є функції," +  
Environment.NewLine + "які задаються іменами з Z за допомогою  
відображення den";
            label2.Text = "";
            radioButton1.Checked = true;
        }
        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "2/10. Functionally  
nominative system:";
            radioButton1.Text = "data algebra: V is the  
carrier of this algebra, which are specified by names from K using  
the dec mapping";
            radioButton2.Text = "the algebra of  
functions are given names from Z using the bek mapping";
            radioButton3.Text = "data algebra: D is the  
carrier of this algebra, and algebra operations are functions," +  
Environment.NewLine + "which are specified by names from Z using the  
den mapping";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
    if (count == 2)
    {
        if (comboBox1.SelectedIndex == 0)
        {
            label1.Text = "3/10. Продовжіть речення.  
Наявність композицій дозволяє:";
            radioButton1.Text = "ієрархічну побудову  
програм із деяких базових функцій багатократним застосуванням  
композицій.";

```

```

        radioButton2.Text = "екстраполювати
програми із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton3.Text = "будувати програми із
деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "3/10. Continue the sentence.
The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 3)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "4/10. Продовжіть речення.
Наявність композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову
програм із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton2.Text = "екстраполювати
програми із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton3.Text = "будувати програми із
деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "4/10. Continue the sentence.
The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";

```

```

        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 4)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "5/10. Продовжіть речення.
Наявність композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову
програм із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton2.Text = "екстраполювати
програми із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton3.Text = "будувати програми із
деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "5/10. Continue the sentence.
The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 5)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "6/10. Продовжіть речення.
Наявність композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову
програм із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton2.Text = "екстраполювати
програми із деяких базових функцій багатократним застосуванням
композицій.";

```



```

        radioButton3.Text = "будувати програми із
деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "6/10. Continue the sentence.
The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 6)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "7/10. Продовжіть речення.
Наявність композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову
програм із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton2.Text = "екстраполювати
програми із деяких базових функцій багатократним застосуванням
композицій.";
        radioButton3.Text = "будувати програми із
деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "7/10. Continue the sentence.
The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}

```

```

    }
}
if (count == 7)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "8/10. Продовжіть речення.  

Наявність композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову  

програм із деяких базових функцій багатократним застосуванням  

композицій.";
        radioButton2.Text = "екстраполювати  

програми із деяких базових функцій багатократним застосуванням  

композицій.";
        radioButton3.Text = "будувати програми із  

деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "8/10. Continue the sentence.  

The presence of compositions allows:";
        radioButton1.Text = "hierarchical  

construction of programs from some basic functions by repeated  

application of compositions.";
        radioButton2.Text = "to extrapolate  

programs from some basic functions by repeatedly applying  

compositions.";
        radioButton3.Text = "build programs from  

some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 8)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "9/10. Продовжіть речення.  

Наявність композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову  

програм із деяких базових функцій багатократним застосуванням  

композицій.";
        radioButton2.Text = "екстраполювати  

програми із деяких базових функцій багатократним застосуванням  

композицій.";
        radioButton3.Text = "будувати програми із  

деяких базових функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}

```

```

        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "9/10. Continue the sentence.  
The presence of compositions allows:";
            radioButton1.Text = "hierarchical  
construction of programs from some basic functions by repeated  
application of compositions.";
            radioButton2.Text = "to extrapolate  
programs from some basic functions by repeatedly applying  
compositions.";
            radioButton3.Text = "build programs from  
some basic functions by repeatedly applying compositions.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
    if (count == 9)
    {
        if (comboBox1.SelectedIndex == 0)
        {
            label1.Text = "10/10. Продовжіть речення.  
Наявність композицій дозволяє:";
            radioButton1.Text = "ієрархічну побудову  
програм із деяких базових функцій багатократним застосуванням  
композицій.";
            radioButton2.Text = "екстраполювати  
програми із деяких базових функцій багатократним застосуванням  
композицій.";
            radioButton3.Text = "будувати програми із  
деяких базових функцій багатократним застосуванням композицій.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "10/10. Continue the sentence.  
The presence of compositions allows:";
            radioButton1.Text = "hierarchical  
construction of programs from some basic functions by repeated  
application of compositions.";
            radioButton2.Text = "to extrapolate  
programs from some basic functions by repeatedly applying  
compositions.";
            radioButton3.Text = "build programs from  
some basic functions by repeatedly applying compositions.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
}

```

```
private void button2_Click_1(object sender, EventArgs e)
```

```

    {
        Close();
    }

private void button5_Click(object sender, EventArgs e)
{
    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);

    int id = table.Rows.Count + 1;
    string img = textBox5.Text;
    string sername = textBox6.Text;
    string name = textBox7.Text;
    string login = textBox3.Text;
    string pass = textBox4.Text;

    DB db2 = new DB();

    DataTable table2 = new DataTable();

    MySqlDataAdapter adapter2 = new MySqlDataAdapter();

    MySqlCommand command2 = new MySqlCommand("INSERT
INTO `quiz` (`id`, `fotourl`, `sername`, `name`, `test1`, `test2`,
`test3`, `test4`, `test5`, `test6`, `test7`, `test8`, `test9`,
`test10`, `login`, `pass`) VALUES (@id, @img, @sername, @name,
@test1, @test2, @test3, @test4, @test5, @test6, @test7, @test38,
@test9, @test10, @login, @pass)", db.getConnection());
    command2.Parameters.Add("@id",
MySqlDbType.VarChar).Value = id;
    command2.Parameters.Add("@img",
MySqlDbType.VarChar).Value = img;
    command2.Parameters.Add("@sername",
MySqlDbType.VarChar).Value = sername;
    command2.Parameters.Add("@name",
MySqlDbType.VarChar).Value = name;
    command2.Parameters.Add("@test1",
MySqlDbType.VarChar).Value = t1;
    command2.Parameters.Add("@test2",
MySqlDbType.VarChar).Value = t2;
    command2.Parameters.Add("@test3",
MySqlDbType.VarChar).Value = t3;

```

```

        command2.Parameters.Add("@test4",
MySqlDbType.VarChar).Value = t4;
        command2.Parameters.Add("@test5",
MySqlDbType.VarChar).Value = t5;
        command2.Parameters.Add("@test6",
MySqlDbType.VarChar).Value = t6;
        command2.Parameters.Add("@test7",
MySqlDbType.VarChar).Value = t7;
        command2.Parameters.Add("@test8",
MySqlDbType.VarChar).Value = t8;
        command2.Parameters.Add("@test9",
MySqlDbType.VarChar).Value = t9;
        command2.Parameters.Add("@test10",
MySqlDbType.VarChar).Value = t10;


        command2.Parameters.Add("@login",
MySqlDbType.VarChar).Value = login;
        command2.Parameters.Add("@pass",
MySqlDbType.VarChar).Value = pass;
        adapter2.SelectCommand = command2;
        adapter2.Fill(table2);


        DB db1 = new DB();

        DataTable table1 = new DataTable();

        MySqlDataAdapter adapter1 = new MySqlDataAdapter();

        MySqlCommand command1 = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

        adapter1.SelectCommand = command1;
        adapter1.Fill(table1);

        dataGridView1.Rows.Clear();

        dataGridView1.RowCount = table1.Rows.Count;
        for (int i = 0; i < table1.Rows.Count; ++i)
        {
            Image image = Image.FromFile("../..\\img\\" +
table1.Rows[i].ItemArray[1].ToString());

            dataGridView1.Rows[i].Cells[0].Value =
table1.Rows[i].ItemArray[0].ToString();
            dataGridView1.Rows[i].Cells[1].Value = image;
            dataGridView1.Rows[i].Cells[2].Value =
table1.Rows[i].ItemArray[2].ToString();
            dataGridView1.Rows[i].Cells[3].Value =
table1.Rows[i].ItemArray[3].ToString();
            dataGridView1.Rows[i].Cells[4].Value =
table1.Rows[i].ItemArray[4].ToString();

```

```

        dataGridView1.Rows[i].Cells[5].Value =
table1.Rows[i].ItemArray[5].ToString();
        dataGridView1.Rows[i].Cells[6].Value =
table1.Rows[i].ItemArray[6].ToString();
        dataGridView1.Rows[i].Cells[7].Value =
table1.Rows[i].ItemArray[7].ToString();
        dataGridView1.Rows[i].Cells[8].Value =
table1.Rows[i].ItemArray[8].ToString();
        dataGridView1.Rows[i].Cells[9].Value =
table1.Rows[i].ItemArray[9].ToString();
        dataGridView1.Rows[i].Cells[10].Value =
table1.Rows[i].ItemArray[10].ToString();
        dataGridView1.Rows[i].Cells[11].Value =
table1.Rows[i].ItemArray[11].ToString();
        dataGridView1.Rows[i].Cells[12].Value =
table1.Rows[i].ItemArray[12].ToString();
        dataGridView1.Rows[i].Cells[13].Value =
table1.Rows[i].ItemArray[13].ToString();
    }
}

private void button6_Click(object sender, EventArgs e)
{
    string id = textBox8.Text;
    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("DELETE FROM
`quiz` WHERE `quiz`.`id` = @id", db.getConnection());
    command.Parameters.Add("@id",
MySqlDbType.VarChar).Value = id;

    adapter.SelectCommand = command;
    adapter.Fill(table);

    DB db1 = new DB();

    DataTable table1 = new DataTable();

    MySqlDataAdapter adapter1 = new MySqlDataAdapter();

    MySqlCommand command1 = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

    adapter1.SelectCommand = command1;
    adapter1.Fill(table1);

    dataGridView1.Rows.Clear();

```

```

        dataGridView1.RowCount = table1.Rows.Count;
        for (int i = 0; i < table1.Rows.Count; ++i)
        {
            Image image = Image.FromFile("../\\..\\img\\" +
table1.Rows[i].ItemArray[1].ToString());

            dataGridView1.Rows[i].Cells[0].Value =
table1.Rows[i].ItemArray[0].ToString();
            dataGridView1.Rows[i].Cells[1].Value = image;
            dataGridView1.Rows[i].Cells[2].Value =
table1.Rows[i].ItemArray[2].ToString();
            dataGridView1.Rows[i].Cells[3].Value =
table1.Rows[i].ItemArray[3].ToString();
            dataGridView1.Rows[i].Cells[4].Value =
table1.Rows[i].ItemArray[4].ToString();
            dataGridView1.Rows[i].Cells[5].Value =
table1.Rows[i].ItemArray[5].ToString();
            dataGridView1.Rows[i].Cells[6].Value =
table1.Rows[i].ItemArray[6].ToString();
            dataGridView1.Rows[i].Cells[7].Value =
table1.Rows[i].ItemArray[7].ToString();
            dataGridView1.Rows[i].Cells[8].Value =
table1.Rows[i].ItemArray[8].ToString();
            dataGridView1.Rows[i].Cells[9].Value =
table1.Rows[i].ItemArray[9].ToString();
            dataGridView1.Rows[i].Cells[10].Value =
table1.Rows[i].ItemArray[10].ToString();
            dataGridView1.Rows[i].Cells[11].Value =
table1.Rows[i].ItemArray[11].ToString();
            dataGridView1.Rows[i].Cells[12].Value =
table1.Rows[i].ItemArray[12].ToString();
            dataGridView1.Rows[i].Cells[13].Value =
table1.Rows[i].ItemArray[13].ToString();
        }
    }

    private void button7_Click(object sender, EventArgs e)
    {
        string search = textBox9.Text;

        DB db = new DB();

        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();

        MySqlCommand command = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

        adapter.SelectCommand = command;
        adapter.Fill(table);
    }

```

```

        dataGridView1.Rows.Clear();

        int count = 0;
        for (int i = 0; i < table.Rows.Count; ++i)
            if (search ==
table.Rows[i].ItemArray[2].ToString())
                count++;

        dataGridView1.RowCount = count == 0 ? 1 : count;
        count = 0;
        for (int i = 0; i < table.Rows.Count; ++i)
        {
            if (search ==
table.Rows[i].ItemArray[2].ToString())
            {
                Image image =
Image.FromFile("../..\\img\\"
table.Rows[i].ItemArray[1].ToString());

                dataGridView1.Rows[count].Cells[0].Value =
table.Rows[i].ItemArray[0].ToString();
                dataGridView1.Rows[count].Cells[1].Value =
image;
                dataGridView1.Rows[count].Cells[2].Value =
table.Rows[i].ItemArray[2].ToString();
                dataGridView1.Rows[count].Cells[3].Value =
table.Rows[i].ItemArray[3].ToString();
                dataGridView1.Rows[count].Cells[4].Value =
table.Rows[i].ItemArray[4].ToString();
                dataGridView1.Rows[count].Cells[5].Value =
table.Rows[i].ItemArray[5].ToString();
                dataGridView1.Rows[count].Cells[6].Value =
table.Rows[i].ItemArray[6].ToString();
                dataGridView1.Rows[count].Cells[7].Value =
table.Rows[i].ItemArray[7].ToString();
                dataGridView1.Rows[count].Cells[8].Value =
table.Rows[i].ItemArray[8].ToString();
                dataGridView1.Rows[count].Cells[9].Value =
table.Rows[i].ItemArray[9].ToString();
                dataGridView1.Rows[count].Cells[10].Value =
table.Rows[i].ItemArray[10].ToString();
                dataGridView1.Rows[count].Cells[11].Value =
table.Rows[i].ItemArray[11].ToString();
                dataGridView1.Rows[count].Cells[12].Value =
table.Rows[i].ItemArray[12].ToString();
                dataGridView1.Rows[count].Cells[13].Value =
table.Rows[i].ItemArray[13].ToString();
                count++;
            }
        }

```



```

    }

    private void button8_Click(object sender, EventArgs e)
    {
        DB db = new DB();

        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();

        MySqlCommand command = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

        adapter.SelectCommand = command;
        adapter.Fill(table);

        DB db2 = new DB();

        DataTable table2 = new DataTable();

        MySqlDataAdapter adapter2 = new MySqlDataAdapter();

        MySqlCommand command2 = new MySqlCommand("UPDATE
`quiz` SET `test1` = @test1, `test2` = @test2, `test3` = @test3,
`test4` = @test4, `test5` = @test5, `test6` = @test6, `test7` =
@test7, `test8` = @test8, `test9` = @test9, `test10` = @test10 WHERE
`quiz`.`login` = @login AND `quiz`.`pass` = @pass",
db.getConnection());
        command2.Parameters.Add("@test1",
MySqlDbType.VarChar).Value = t1;
        command2.Parameters.Add("@test2",
MySqlDbType.VarChar).Value = t2;
        command2.Parameters.Add("@test3",
MySqlDbType.VarChar).Value = t3;
        command2.Parameters.Add("@test4",
MySqlDbType.VarChar).Value = t4;
        command2.Parameters.Add("@test5",
MySqlDbType.VarChar).Value = t5;
        command2.Parameters.Add("@test6",
MySqlDbType.VarChar).Value = t6;
        command2.Parameters.Add("@test7",
MySqlDbType.VarChar).Value = t7;
        command2.Parameters.Add("@test8",
MySqlDbType.VarChar).Value = t8;
        command2.Parameters.Add("@test9",
MySqlDbType.VarChar).Value = t9;
        command2.Parameters.Add("@test10",
MySqlDbType.VarChar).Value = t10;
        command2.Parameters.Add("@login",
MySqlDbType.VarChar).Value = login;
    }

```

```

        command2.Parameters.Add("@pass",
MySQLDbType.VarChar).Value = password;
        adapter2.SelectCommand = command2;
        adapter2.Fill(table2);

        DB db1 = new DB();

        DataTable table1 = new DataTable();

        MySqlDataAdapter adapter1 = new MySqlDataAdapter();

        MySqlCommand command1 = new MySqlCommand("SELECT *
FROM `quiz`", db.getConnection());

        adapter1.SelectCommand = command1;
        adapter1.Fill(table1);

        dataGridView1.Rows.Clear();

        dataGridView1.RowCount = table1.Rows.Count;
        for (int i = 0; i < table1.Rows.Count; ++i)
        {
            Image image = Image.FromFile("../\\..\\img\\" +
table1.Rows[i].ItemArray[1].ToString());

            dataGridView1.Rows[i].Cells[0].Value =
table.Rows[i].ItemArray[0].ToString();
            dataGridView1.Rows[i].Cells[1].Value = image;
            dataGridView1.Rows[i].Cells[2].Value =
table.Rows[i].ItemArray[2].ToString();
            dataGridView1.Rows[i].Cells[3].Value =
table.Rows[i].ItemArray[3].ToString();
            dataGridView1.Rows[i].Cells[4].Value =
table.Rows[i].ItemArray[4].ToString();
            dataGridView1.Rows[i].Cells[5].Value =
table.Rows[i].ItemArray[5].ToString();
            dataGridView1.Rows[i].Cells[6].Value =
table.Rows[i].ItemArray[6].ToString();
            dataGridView1.Rows[i].Cells[7].Value =
table.Rows[i].ItemArray[7].ToString();
            dataGridView1.Rows[i].Cells[8].Value =
table.Rows[i].ItemArray[8].ToString();
            dataGridView1.Rows[i].Cells[9].Value =
table.Rows[i].ItemArray[9].ToString();
            dataGridView1.Rows[i].Cells[10].Value =
table.Rows[i].ItemArray[10].ToString();
            dataGridView1.Rows[i].Cells[11].Value =
table.Rows[i].ItemArray[11].ToString();
            dataGridView1.Rows[i].Cells[12].Value =
table.Rows[i].ItemArray[12].ToString();
            dataGridView1.Rows[i].Cells[13].Value =
table.Rows[i].ItemArray[13].ToString();

```

```

    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex == 0)
    {
        groupBox1.Text = "Вхід";
        label3.Text = "Логін";
        label4.Text = "Пароль";
        button4.Text = "Вхід";
        groupBox4.Text = "Пошук";
        label11.Text = "Прізвище";
        button7.Text = "Пошук";
        textBox9.Text = "Іванов";
        button1.Text = "НАСТУПНЕ";
        button3.Text = "ПЕРЕВІРИТИ";
        button2.Text = "ВИЙТИ";
        tabPage1.Text = "Тести";
        tabPage2.Text = "Рейтинг";
        groupBox2.Text = "Додати користувача";
        groupBox3.Text = "Видалити за номером";
        label5.Text = "Логін";
        label6.Text = "Пароль";
        label7.Text = "Зображення";
        label8.Text = "Прізвище";
        label9.Text = "Ім'я";
        label10.Text = "Видалити №";
        button8.Text = "Оновити";
        button5.Text = "Додати";
        button6.Text = "Видалити";
    }
    if (comboBox1.SelectedIndex == 1)
    {
        groupBox1.Text = "Enter";
        label3.Text = "Login";
        label4.Text = "Password";
        button4.Text = "Enter";
        groupBox4.Text = "Search";
        label11.Text = "Sername";
        button7.Text = "Search";
        textBox9.Text = "Ivanov";
        button1.Text = "NEXT";
        button3.Text = "CHECK";
        button2.Text = "EXIT";
        tabPage1.Text = "Tests";
        tabPage2.Text = "Rating";
        groupBox2.Text = "Add user";
        groupBox3.Text = "Delete by number";
        label5.Text = "Login";
        label6.Text = "Pass";
    }
}

```

```
label7.Text = "Image";  
label8.Text = "Sername";  
label9.Text = "Name";  
label10.Text = "Delete №";  
button8.Text = "Refresh";  
button5.Text = "Add";  
button6.Text = "Delete";
```

```
}
```

```
}
```

```
}
```

```
}
```