ТЕХНІЧНИЙ КОЛЕДЖ ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА ПУЛЮЯ

Циклова комісія програмних систем і комплексів

**КУРСОВА РОБОТА**

з дисципліни:

**Об’єктно-орієнтоване програмування**

на тему: **Розробка програмного забезпечення для** **тестової системи**

**« Автошкола»**

Студента  3  курсу групи КН-321 спеціальності 122 «Комп’ютерні науки»

Качуровського Д.З.

(прізвище та ініціали)

Керівник: викладач   Слободян  Р.О.         
Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_\_\_ Р.О. Слободян

(підпис)

\_\_\_\_\_\_\_\_\_ Г.Я. Марціяш

(підпис)

м. Тернопіль – 2019

ЗМІСТ

[ВСТУП 4](#_Toc26434668)

[1 ТЕХНІЧНЕ ЗАВДАННЯ 5](#_Toc26434669)

[1.1 Найменування та область застосування 5](#_Toc26434670)

[1.2 Підстави для розробки 5](#_Toc26434671)

[1.3 Призначення розробки 5](#_Toc26434672)

[1.4 Вимоги до програми чи програмного виробу 5](#_Toc26434673)

[1.5 Вимоги до програмної документації 6](#_Toc26434674)

[1.6 Техніко-економічні показники 6](#_Toc26434675)

[1.7 Стадії та етапи розробки 6](#_Toc26434676)

[1.8 Порядок контролю та прийому 6](#_Toc26434677)

[2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ 7](#_Toc26434678)

[2.1 Розробка загальної структури і варіантів використання програми 7](#_Toc26434679)

[2.2 Розробка системи класів 7](#_Toc26434680)

[2.3 Розробка методів 7](#_Toc26434681)

[2.4 Проєктування і опис інтерфейсу користувача 7](#_Toc26434682)

[2.5 Опис файлової структури програми 7](#_Toc26434683)

[2.6 Опис структури бази даних програми 7](#_Toc26434684)

[3 ТЕСТУВАННЯ ПРОГРАМИ І РЕЗУЛЬТАТИ ЇЇ ВИКОНАННЯ 8](#_Toc26434685)

[ВИСНОВКИ 9](#_Toc26434686)

[ПЕРЕЛІК ПОСИЛАНЬ 10](#_Toc26434687)

[Додаток А Діаграма варіантів використання 11](#_Toc26434688)

[Додаток Б Лістинг файлу «program.h» 12](#_Toc26434689)

[Додаток В Лістинг файлу «program.cpp» 13](#_Toc26434690)

[Додаток Г Компакт-диск із програмним продуктом 14](#_Toc26434691)

ВСТУП Усучасному повсякденному житті використання продуктів програмування вже стало загальновживаним явищем. Тому з кожним днем практика програмування вимагає все більшого вдосконалення прийомів розробки комп’ютерних програм, прискорення та полегшення процесу їх проектування. Важливим фактором є орієнтування програмних засобів на тривалу роботу з користувачем. Як результат, розробник-програміст має завдання дотримуватися вимог, що спрямовані не тільки на технологічні сторони організації інтерфейсу програми, а й на те, як звичайний користувач сприйматиме подану інформацію ( її кількість, системність, логічну послідовність, чіткість, структурування і т.ін.).

У сучасному економічному просторі вже неможливо обійтися без використання спеціалізованого програмного забезпечення. На ринку праці існує чималий попит на фахівців, які здатні розробляти нескладні програмні засоби, орієнтовані на розв’язання вузько спеціалізованих завдань пов’язаних з веденням бухгалтерії, обліку, аудиту, статистикою, менеджментом, маркетингом, проходженням тренувань і, звісно, навчанням. Отож, сучасний освітній менеджмент вимагає підготовки фахівців, які спроможні створювати гнучкі програмні засоби, не витрачаючи на їх розробку та впровадження багато фінансів та часу. Як приклад, в даній курсовій роботі представлено розробку одного із таких спеціалізованих програмних рішень мовою С++ і з використанням принципів об’єктно-орієнтованого програмування. Темою мого проєкту є «Розробка програмного забезпечення для тестової системи «Автошкола». Метою його розробки – створення бази тестових завдань згідно діючих Правил дорожнього руху для проходження тренувальних вправ під час навчання в автошколі. Дана розробка є над актуальною, оскільки у рамках реформування сервісних центрів МВС активно впроваджуються нові стандарти підготовки водіїв та складання іспитів для отримання посвідчення водія. Тестова система «Автошкола» - це шлях оптимізації процесу підготовки майбутніх водіїв під час проходження навчального курсу в автошколі.

Даний курсовий проект побудований на основі об'єктно-орієнтованого програмування. А це - парадигма програмування, заснована на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування. Спочатку я розробляв клас, а на його основі при виконанні програми створював конкретні об'єкти (екземпляри класів). На основі класів можна створювати нові, які розширюють базовий клас і таким чином створюється ієрархія класів. Це дозволило мені швидко конструювати достатньо функціональний програмний засіб.

# ТЕХНІЧНЕ ЗАВДАННЯ

## Найменування та область застосування

Найменування програми – Програма «Тестова система «Автошкола».

Стисла назва програми - «test-system»

Область застосування програми – ведення контролю за проходженням навчального (тренувального) процесу учнями автошколи, що ведеться керівником навчальної групи та підготовка до здачі іспиту на знання Правил дорожнього руху для отримання посвідчення водія.

## Підстави для розробки

Підставами для проведення розробки є індивідуальне завдання підготувати курсову роботу з дисципліни «Об’єктно-орієнтоване програмування».

Найменування теми курсової роботи – «Розробка програмного забезпечення для тестової системи «Автошкола».

Виконавець – студент групи КН-321 Качуровський Дмитро Зіновійович.

## Призначення розробки

Дане програмне забезпечення створене для широкого кола людей різного віку, які проходять навчання в автошколі з метою отримання водійського посвідчення. Проходження тестів на віртуальній машині допомагає не лише оптимізувати процес підготовки майбутніх водіїв, а й зробити прозорою систему роботи навчальних центрів та МВС в цілому. Більше цього, у час цифрових технологій для молодого покоління людей такий спосіб навчання є більш звичним, доступним та економить чимало часу.

## Вимоги до програми чи програмного виробу

### Вимоги до функціональних характеристик

Програмне забезпечення для тестової системи «Автошкола» повинне забезпечити тренування виконання тестових завдань з ПДР та ведення контролю успішності учнів. Для цього потрібно авторизуватися адміністратору та створити тестові завдання відповідно до теми , що вивчається. Користувачеві потрібно зареєструватися або авторизуватися, якщо він вже був зареєстрований раніше. У вікні користувача потрібно ввести назву тесту. Після цього автоматично відкривається вікно виконання тесту. Кількість тестових завдань може бути необмеженою. Кожне тестове завдання має 4 варіанти відповідей. Користувач вибирає відповідь, яку вважає вірною. Після закінчення виконання усіх завдань з’являється віконечко з кількістю набраних балів.

### Вимоги до часових характеристик

немає

### Вимоги до надійності

Надійне функціонування програми повинне забезпечуватись наявністю блоку перевірки правильності введених даних. Це збереже від помилок, що виникають при виконанні рутинної роботи користувачем. Також необхідно забезпечити достатню стійкість програми до некваліфікованих дій користувача, що досягається супроводом будь-яких операцій в програмі чіткими і зрозумілими повідомленнями.

### Умови до експлуатації

Розроблений програмний продукт по потребуватиме яких-небудь видів обслуговування, для його використання достатньо буде навиків користувача ПК.

Забезпечення учнів автошколи персональними комп’ютерами.

Наявність операційної системи Linux.

### Вимоги до складу і параметрів технічних засобів

В склад технічних засобів повинні входити: монітор (діагоналлю не менше15”), клавіатура, мишка, IBM-сумісний персональний комп’ютер із такими мінімальними характеристиками:процесор –IntelPentium4;оперативна пам’ять -1ГБ;обсяг дискової пам’яті –80ГБ.

### Вимоги до інформаційної і програмної сумісності

Вихідні коди програми повинні бути реалізовані мовою С++. В якості середовища розробки програми повинне використовуватись середовище Qt Creator версії не нижче 5.12. Системні програмні засоби і утиліти, які буде використовувати програма, повинні забезпечуватись операційною системою сімейства Linux версії не нижче Ubuntu 16.04. Вимоги до захисту інформації і програми не пред’являються.

## Вимоги до програмної документації

Закінчивши розробку програмного забезпечення потрібно підготувати таку документацію:

-інструкція інсталяції програми;

-загальні відомості про можливості програми;

-інструкція з експлуатації.

## Техніко-економічні показники

Розрахунок економічної ефективності і вартості розробки програмного продукту не проводився.

Приблизне число використання розробленої програми в рік –1000-1500 разів

## Стадії та етапи розробки

Розробка програми тестування «Автошкола» буде мати такі стадії:

Аналіз. Метою аналізу є максимально повний опис поставлених завдань. На цій стадії аналізується предметна область завдання, виконується об'єктна декомпозиція програми та визначаються особливості поведінки об'єктів. За результатами аналізу розробляється структурна схема програми, описуються об'єкти та події над ними.

Проектування. Проектування поділяється на логічне та фізичне. Логічне полягає у розробці структури класів, коли визначаються поля для збереження складових об'єктів, алгоритми методів, що описують поведінку об'єктів. Фізичне проектування містить об'єднання опису класів у модулі, вибір схеми їх підключення (статична або динамічна), визначення способів взаємодії з обладнанням, операційною системою та іншим програмним забезпеченням (базами даних, мережним оточенням), забезпечення синхронізації процесів для систем рівнобіжної обробки тощо. На етапі проектування використовуються такі прийоми розробки класів як спадкування, композиція, наповнення. Результатом етапу проектування є створення ієрархії класів та їх опис.

Еволюція. Еволюція системи представляє етап послідовної реалізації та підключення класів до проекту. Процес починається із створення основного інтерфейсу проєкту, бібліотеки класів, описуються тип поля класу, розробляються алгоритми методів, уточнюються вимоги до програми та вносяться певні зміни. Одержаний прототип продукту тестується й налагоджується.

## Порядок контролю та прийому

Прийом розробленого програмного забезпечення повинен відбуватися на об’єкті Замовника в терміни, які зазначені в індивідуальному завданні. Для прийому роботи Виконавець повинен представити:

діючу програму, яка повністю відповідає даному технічному завданню;

вихідний програмний код, записаний разом із програмою на оптичний носій інформації.

Прийом програмного забезпечення повинен відбуватися перед комісією з двох чоловік(один з яких – Замовник)у такій послідовності:

доповідь Виконавця про виконану роботу;

демонстрація Виконавцем роботи програми;

контрольні випробовування роботи програми;

відповіді на запитання і зауваження комісії.

# РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

## Розробка загальної структури і варіантів використання програми

Тест-програма призначена для створення та редагування тестів викладачем, та відповідно проходження студентами. Є два діючих суб’єкта (більше не передбачено): користувач (студент автошколи), та адміністратор (викладач). Адміністратор повинен авторизуватись, створити тест, при створенні повинен вказати кількість питань, та назву тесту. Щоб вказати яке з питань правильне потрібно в вікні адміністратора вибрати ʼредагувати тестʼ, та вписати ключове слово, за яким і буде виконуватись пошук правильної відповіді. Для того, щоб подивитись користувачів та їх бали адміністратору потрібно вибрати ‘показати користувачів’, та ‘показати оцінки’ відповідно. Для видалення тесту адміністратор повинен вибрати ‘видалити тест’ та вказати назву тесту, після чого вся інформація буде видалена. Користувач повинен створити свій профіль вказавши ім’я та прізвище і придумати пароль, після чого ці дані вносяться в базу даних, або авторизуватись, якщо до цього користувач вже був зареєстрований. У вікні користувача, щоб пройти тест потрібно ввести назву тесту, після чого з’являється  вікно проходження тесту, у якому потрібно вибирати одну з чотирьох варіантів відповіді. Тест проходить до кількості питань які були вказані при створенні тесту. Після закінчення тесту виводиться бал який набрав користувач, і закривається вікно проходження тесту, після чого можна подивитись всі свої попередні бали.

З цього випливає, що діючий суб’єкт адміністратор буде виконувати такі дії:

- авторизуватись;

- добавляти тести;

- редагувати тести;

- видаляти тести;

- виводити бали користувачів;

- виводити усіх користувачів.

Діючий суб’єкт користувач буде виконувати такі дії:

- авторизовуватись;

- проходити тест;

- виводити свої бали.

Таким чином, діаграму функцій адміністратора та користувача, які отримуються внаслідок переліку його дій, можна подати так як це зображено на рисунку 2.1.

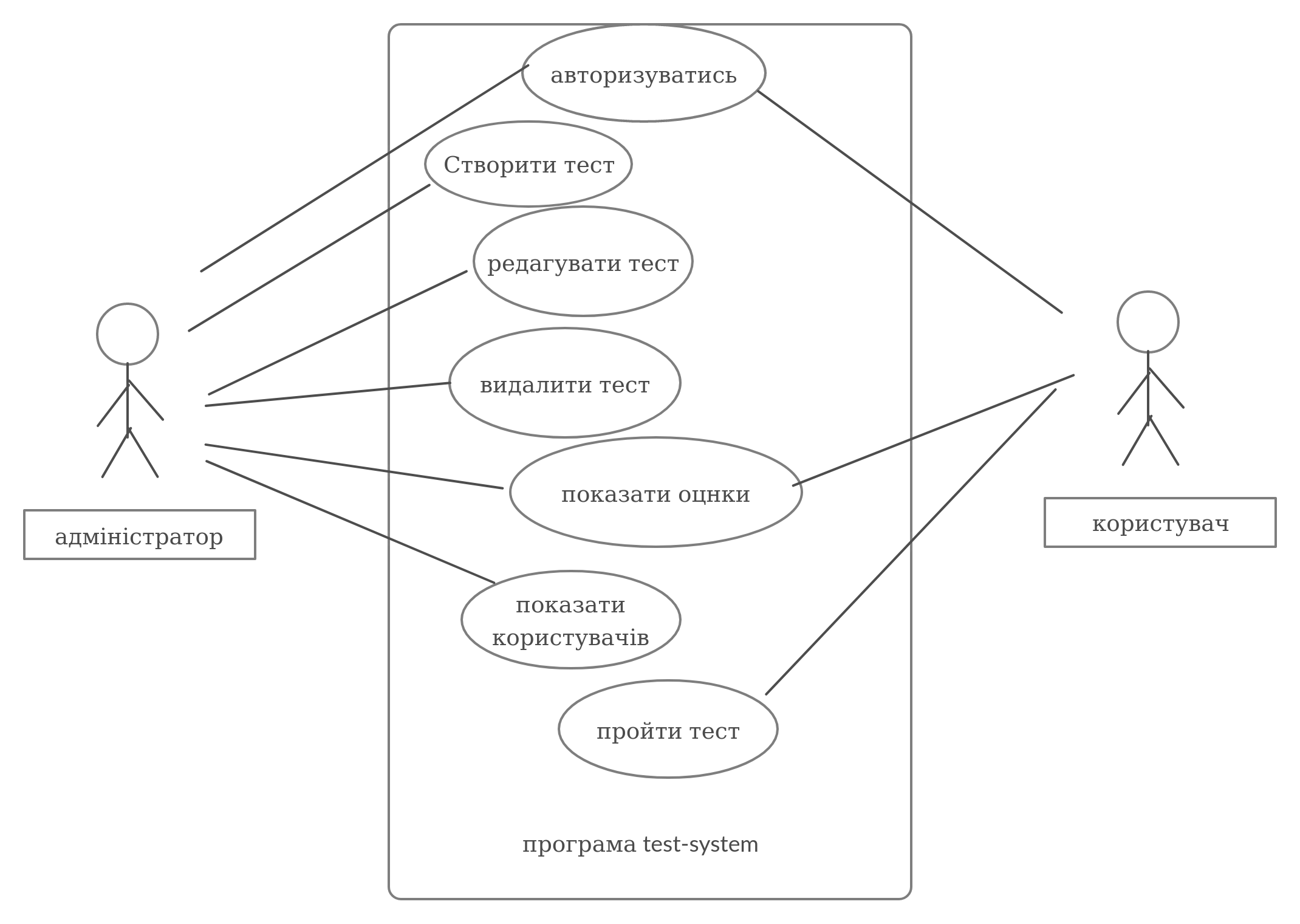


Рисунок 2.1 - Діаграма варіантів використання програми TestSystem

**Авторизуватись.** Коли запускається програма на екран виводиться вікно авторизації, у поля які називаються ‘Ім’я’ та ‘Пароль’ потрібно ввести: admin, для того, щоб авторизуватись як адміністратор. Якщо користувач реєструється в перше, то потрібно ввести ім’я яке не заняте, інакше у вікні виведеться помилка ‘проблема авторизації’. Якщо користувач введе ім’я та пароль які співпадають з якимось користувачем то він авторизується.

**Добавити тест.** В інтерфейсі адміністратора потрібно натиснути на кнопку з назвою ‘створити тест’, після цього відкриється вікно створення тесту, у якому потрібно ввести назву тесту та кількість питань, при натисканні на кнопку створити, з’являється поле у якому потрібно ввести умову тесту, та 4 поля з варіантами відповіді, які потрібно заповнити та натиснути кнопку ок.

**Редагувати тест.** В інтерфейсі адміністратора потрібно натиснути кнопку з назвою ‘редагувати тест’, після цього відкриється вікно редагування тесту, в поле потрібно ввести назву тесту, та натиснути кнопку ‘редагувати’ після чого в полі буде відкритий файл з тестом, після закінчення редагування потрібно натиснути кнопку ‘зберегти’.

**Видалити тест.** В інтерфейсі адміністратора потрібно натиснути на кнопку з назвою ‘видалити тест’, після чого відкриється вікно видалення тесту, у якому потрібно ввести назву тесту який потрібно видалити та натиснути кнопку ‘видалити’.

**Подивитись оцінки.** В інтерфейсі адміністратора потрібно натиснути на кнопку ‘показати всі оцінки’, після чого в таблиці яка розташована нижче буде виведено імена користувачів,які здавали тест, та оцінка з назвою тесту.

**Подивитись користувачів.** В інтерфейсі адміністратора потрібно натиснути на кнопку ‘показатти користувачів’, після чого в таблиці яка розташована нижче буде виведено усіх користувачів які були заре’єстровані.

**Пройти тест.** В інтерфейсі користувача потрібно вписати у поле назву тесту, та натиснути кнопку ‘почати тест’, після чого відкриється вікно проходження тесту, у якому потрібно вибирати одну з чотирьох правильних відповідей. Після того, як тест був пройдений буде виведена кількість набраних балів.

**Подивитись оцінки.** В інтерфейсі користувача потрібно натиснути ‘показати оцінки’, після чого у нижній частині вікна з’явиться таблиця, в якій виведуться усі оцінки користувача.

## Розробка системи класів

Програма ‘test-system’ складаається з дев’яти явно визначених класів:

1. вікно авторизації;

2. інтерфейс адміністратора;

3. вікно створення тесту;

4. вікно редагування тесту;

5. вікно видалення тесту;

6. інтерфейс користувача;

7. вікно проходження тесту;

8. QmainWindow;

9. налаштування бази даних userTest.db.

**Визначення атрибутів.** У класі mainWindow або клас авторизації, визначено атрибути для роботи з базою даних, та атрибут ім’я користувача.

У класі інтерфейсу адміністратора admininterface, використано чотири атрибути, два з них для роботи з базою даних, атрибут який відповідає за відображення елементів у вікні, та один для стилізування таблиці.

У класі створення тесту — використано три атрибути, один для роботи з базою даних, для переліку кількості питань та атрибут вікна createtest.ui.

У класі редагування тесту — використано три атрибути, для роботи з моделею таблиці, для роботи з базою даних та атрибут вікна edittest.ui.

У класі видалення тесту — використано три атрибути, для роботи з моделею таблиці, для роботи з базою даних та атрибут вікна removetest.ui.

У класі інтерфейсу користувача userInterface — використано шість атрибутів: для роботи з моделею таблиці, атрибут вікна userinterface.ui, для роботи з базою даних, та три атрибута типу Qstring, які відповідають за ім’я користувача, оцінку та порядковий номер користувача.

У класі проходження тесту, або completetest — п’ятнадцять атрибутів, дев’ять з них для передачі, питань з файлу та підрахунку балів і питань. Інші використовуються для роботи з базою даних, та вставкою даних в БД.

У класі налаштування бази даних використано один атрибут для створення бази даних.

**Перехід від дієслів до повідомлень.** У вікні авторизації визначено два поля: ім’я та прізвище і пароль, в які потрібно ввести дані для авторизаці, після чого дані заносяться в таблицю, якщо такого користувача ще немає, інакше, користувач ввівши ім’я та пароль які є в таблиці — авторизується. У вікні адміністратора потрібно вибрати потрібну кнопку, для реалізації дії (видалення, створення, редагування, вивід). У вікні створення тесту, потрібно ввести ім’я та кількість питань тесту, якщо дані введені вірно у вікні з’явиться п’ять полів, в які відповідно потрібно ввести дані. У вікні редагування є одне поле, в яке потрібно ввести назву тесту, для початку редагування, та сама ситуація з вікном видалення тесту. У вікні користувача, є поле в яке потрібно ввести назву тесту для його проходждення. У вікні проходження тесту — потрібно слідувати інструкції вказаній в підсказці.

**Побудова діаграм класів.** Знаючи, які класи буде включено в розроблювальну програму і як вони пов’язані між собою, можна побудувати спрощену діаграму класів, що зображуватиме взаємозв’язки між класами. Дану діаграму зображено на рисунку 2.2.

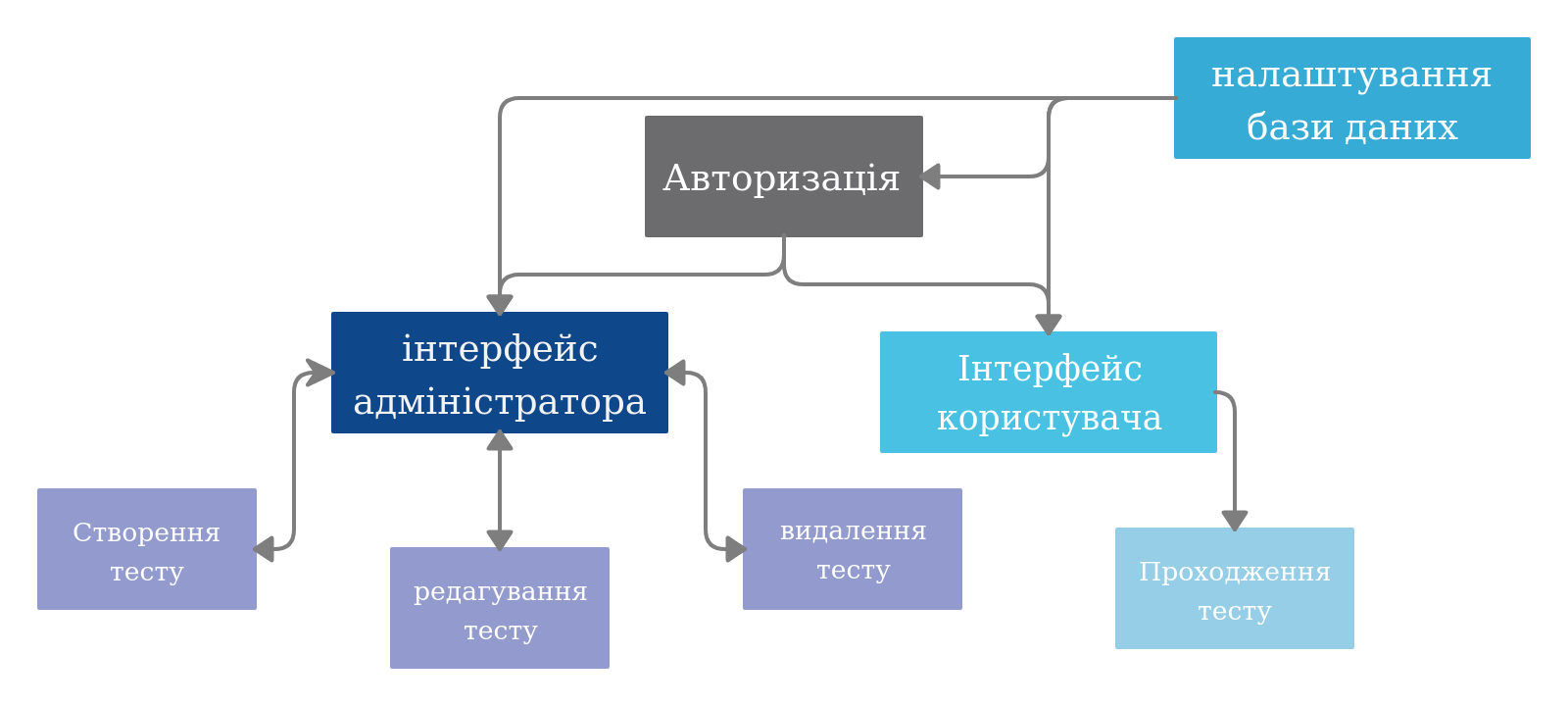
****

Рисунок 2.2 - Діаграма класів для програми test-system

## Розробка методів

## Проєктування і опис інтерфейсу користувача

При запуску програми, відкривається вікно авторизації mainwindow.ui, в якому потрібно ввести ім’я та пароль, якщо ввести дані `admin; admin` - відкриється вікно інтерфейсу адміністратора admininterface.ui, у якому при натисканні на кнопку ‘створити тест’ відкриється вікно створення тесту createtest.ui, у якому потрібно вводити дані (назва тесту, кількість питань, умова, варіанти відповідей), при натисканні на кнопку ‘редагувати тест’ відкриється вікно редагування edittest.ui, в якому потрібно ввести назву тесту, та після змін зберегти, натиснувши на потрібну кнопку, при натисканні на кнопку ‘видалити тест’ відкривається вікно видалення тесту removetest.ui, у якому потрібно ввести назву тесту та нажати кнопку ‘видалити’. Якщо під час авторизації увійти як простий користувач — відкриється вікно інтерфейсу користувача userinterface.ui в якому потрібно ввести в поле назву тесту та натиснути на кнопку ‘почати тест’, тоді відкриється вікно проходження тесту completetest.ui.

## Опис файлової структури програми

Головна функція програми main() зберігаємо в одному коротенькому файлі «main.cpp».Лістинг даного файлу представлено у додатку А. У функції main() створюється об'єкт MainWindow і викликається метод show().

У файлі MainWindow.h під’єднані бібліотеки для роботи з головним вікном та створюється клас головного вікна, у якому оголошені методи для роботи з головним вікном. Лістинг даного файлу представлено у додатку Б. У файлі mainwindow.cpp проходить реалізація методів які були оголошені у заголовочному файлі MainWindow.h. Лістинг даного файлу представлено у додатку В. Файл mainwindow.ui містить форму, для авторизації, та поля, логіка яких реалізована у файлі mainwindow.cpp.

У файлі userdatabase.h оголошено константні змінні з даними про базу данних та таблиці, та оголошено декілька методів для роботи з базами даних.Також підключено бібліотеки для роботи з БД. Лістинг даного файлу представлено у додатку Г. У файлі userdatabase.cpp реалізовано методи для роботи з базами, також створюється база даних у заданій директорії. Лістинг даного файлу представлено у додатку Д.

У файлі admininerface.h під’єднані бібліотеки для роботи з таблицями. Також оголошені методи для реплізації вигляду та констроювання таблиці, оголошені атрибути, для роботи з базами даних, та моделею таблиці, та слоти. Лістинг даного файлу представлено у додатку Е. Файл admininterface.cpp слугує для реалізації методів, та слотів які оголошені у файлі admininterface.h. Лістинг даного файлу представлено у додатку Є.

У файлі createtest.h підключені бібліотеки для роботи з файлами, базами даних, оголошено атрибути для роботи з базами даних, та приватні слоти. Лістинг даного файлу представлено у додатку Ж. У файлі createtest.cpp реалізований алгоритм створення тесту, та слоти які були оголошені у заголовочному файлі, також реалізована робота з файловою системою та базами даних. Лістинг даного файлу представлено у додатку З. У файлі createtest.ui створено форму створення тесту, реалізація логіки поведінки елементів форми, знаходиться у файлі createtest.cpp.

У файлі edittest.h підключено бібліотеки для роботи з базами даних та таблицями, також оголошено методи для відображення таблиці, та атрибути для роботи з базою даних. Лістинг даного файлу представлено у додатку И. У файлі edittest.cpp реалізовано методи та слоти оголошені у заголовочному файлі, також реалізовано методи для редагування файлу у вікні програми, та збереження змін і вигляд таблиці. Лістинг даного файлу представлено у додатку І. У файлі edittest.ui створено форму, поведінку елементів реалізовано у файлі edittest.cpp.

У файлі removetest.h підключено бібліотеки для роботи з файлами, таблицями та базами даних, також оголошено методи для роботи з таблицями, та атрибути, для роботи з базами даних. Лістинг даного файлу представлено у додатку К. Файл removetest.cpp реалізує методи, слоти з заголовочного файлу, та реалізує алгоритм видалення тесту, одразу з файлової системи та з таблиці в базі даних. Лістинг даного файлу представлено у додатку Л. У файлі removetest.ui створено форму,а поведінку елементів форми реалізовано у файлі removetest.cpp

У файлі userinterface.h підключено бібліотеки для роботи з файлами, базами даних, таблицями. Оголошено методи, слоти, та атрибути для роботи з базами та файлами. Лістинг даного файлу представлено у додатку М. Файл userinterface.cpp реалізує методи та слоти заголовочного файлу. Лістинг даного файлу представлено у Н. У файлі userinterface.ui створено форму, поведінку елементів реалізовано у файлі userinterface.cpp.

У файлі completetest.h підключено бібліотеки для роботи з файлами, таблицями та базами даних, оголошено методи та атрибути для реалізації алгоритму проходження тесту. Лістинг даного файлу представлено у додатку П. Файл completetest.cpp реалізує алгоритм проходження тесту, та оголошені в заголовочному файлі методи, слоти. Лістинг даного файлу представлено у додатку Р. У файлі completetest.ui створено форму, поведінку елементів реалізовано у файлі completetest.cpp.

## Опис структури бази даних програми

База даних у програмі ‘test-sytem’ реалізована у вигляді одного файлу userTest.db. У базі даних є три таблиці: users, marks, testTable.

Таблиця users. При створені нового користувача його дані заносяться у таблицю, якщо користувач з такими даними вже існує, то дані повторно не заносяться, також таблиця users зв’язана полем id з таблицею marks.

Таблиця testTable. При створені тесту дані такі як : назва тесту, кількість питань та прямя ссилка до файлу заносяться у таблицю. І використовуються при проходженні тесту, видалення тесту та редагування тесту.

Таблиця marks. Прив’язана своїм полем user\_id до таблиці users. При проходженні тесту користувачем бал, який він набрав по закінченню теста, та назва тесту заносяться до таблиці, в поле mark, та привязується по id до таблицю users.

# ТЕСТУВАННЯ ПРОГРАМИ І РЕЗУЛЬТАТИ ЇЇ ВИКОНАНН

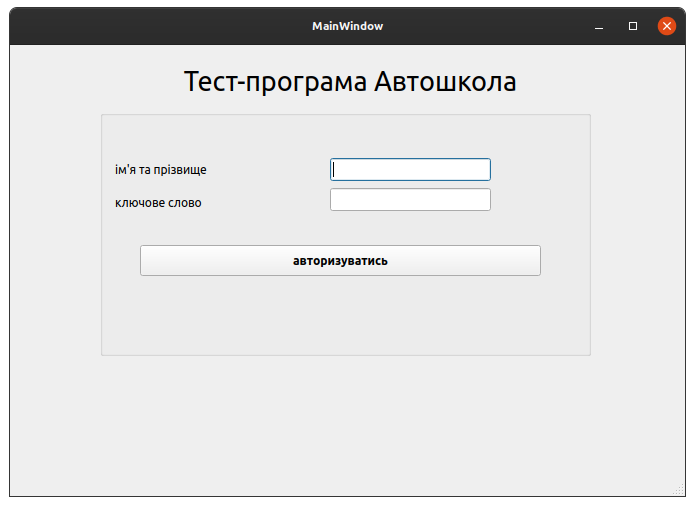
В ході тестування програми було виявлено івиправлено незначні помилки.

Запустивши програму на екрані монітора відображається вікно авторизації, яке представлено на рисунку 3.1. З його допомогою можна виконати такі дії:

- авторизуватись як адміністратор;

- авторизуватись як користувач;

- створити нового користувача.

Рисунок 3.1 — Вікно авторизації

Якщо авторизуватись як адміністратор - відкриється вікно інтерфейсу адміністратора, яке представлено на рисунку 3.2. З його допомогою можна виконати такі дії:

- вивести усіх користувачів;

- вивести оцінки користувачів;

- створити тест;

- редагувати тест;

- видалити тест.

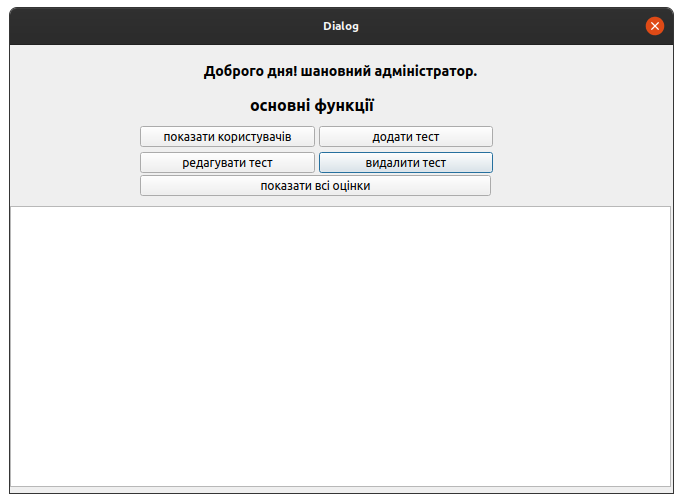
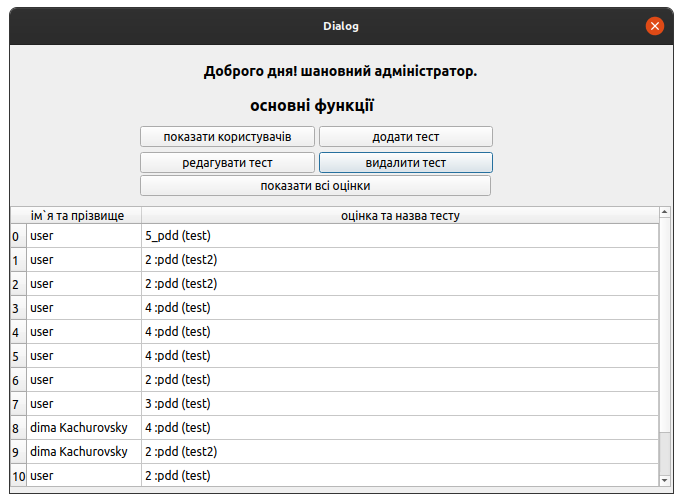
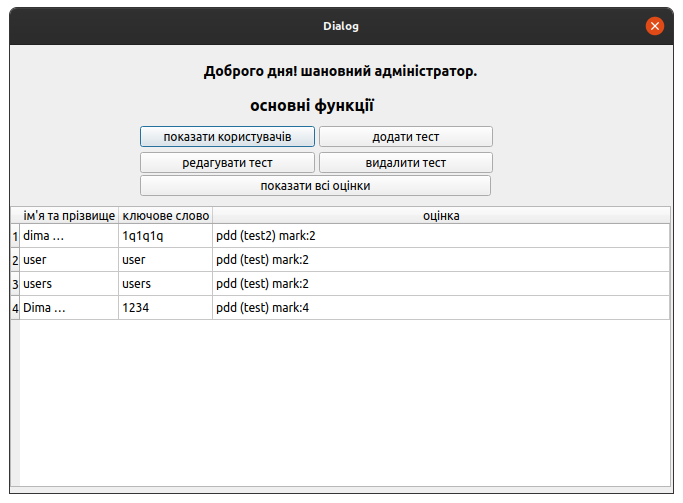


Рисунок 3.2 — Інтерфейс адміністратора

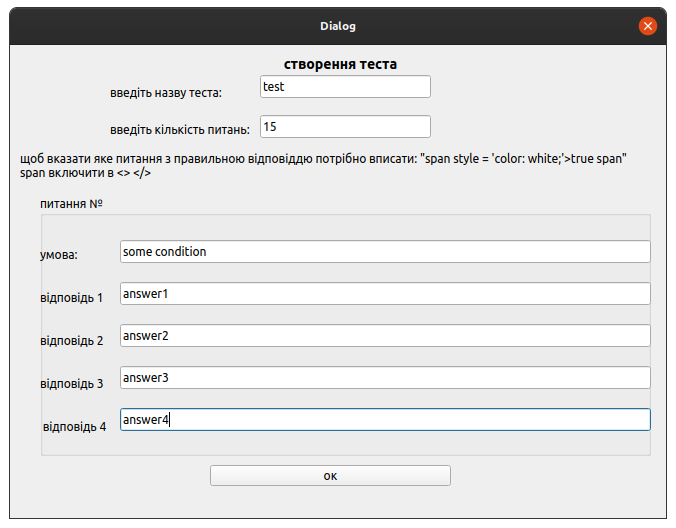
При натисканні на кнопку ‘показати всі оцінки’ - в таблицю розташовану нижче виведуться усі оцінки користувачів які проходили тести. Виведення таблиці показано на рисунку 3.3.

Рисунок 3.3 - Вивід в таблицю оцінок користувачів

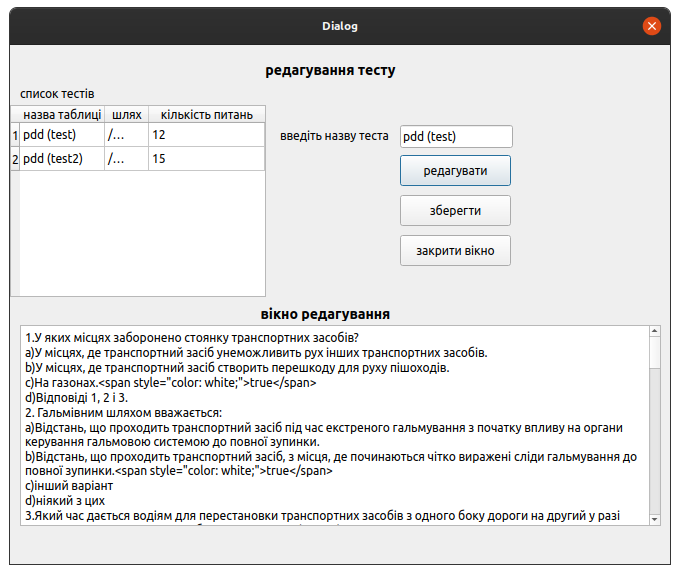
При натисканні на кнопку ‘показати користувачів’ в таблицю виведуться усі зареєстровані користувачі. Виведення таблиці показано на рисунку 3.4.

Рисунок 3.4 - Виведення зареєстрованих користувачів

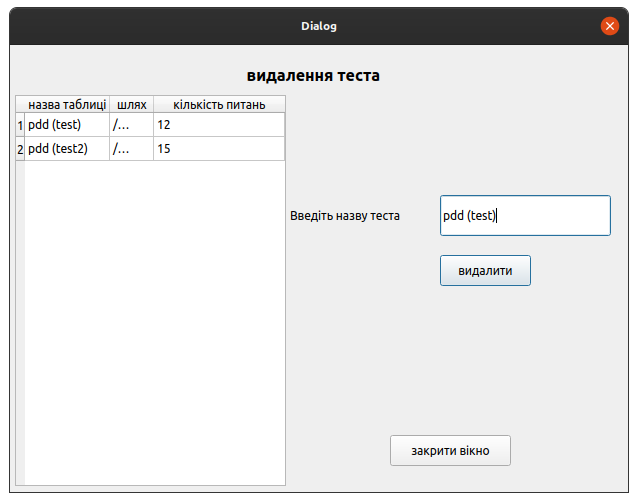
При натисканні на кнопку ‘додати тест’, після чого відкриється вікно створення тесту, яке зображено на рисунку 3.5. З його допомогою користувач може створити тест, з зазначеною кількістю питань.

Рисунок 3.5 — Вікно створення тесту

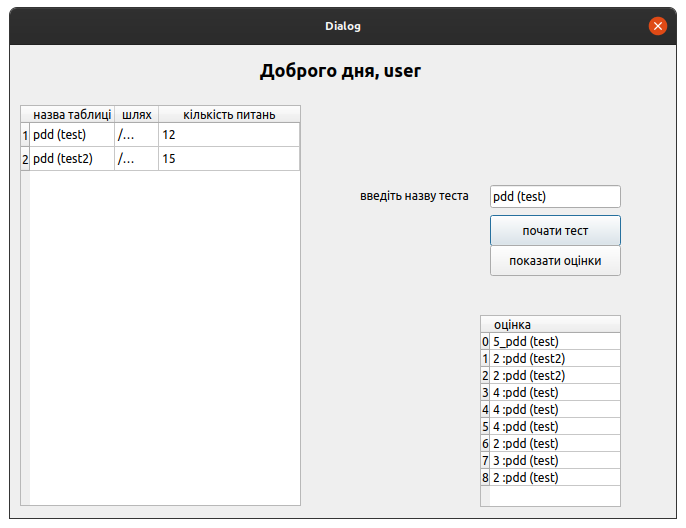
При натисканні на кнопку ‘редагувати тест’ відкриється вікно редагування. Після чого потрібно ввести назву тесту, які зображенні в таблиці зліва та натиснути кнопку ‘редагувати’, після чого знизу вікна з’явиться поле в якому можна редагувати тест, після редагування тесту, потрібно натиснути на кнопку ‘зберегти’, для збереження файлу. Процес редагування тесту зображенно на рисунку 3.6. З допомогою вікна редагування користувач отримує змогу відредагувати тест.

 Рисунок 3.6 - Вікно редагування тесту

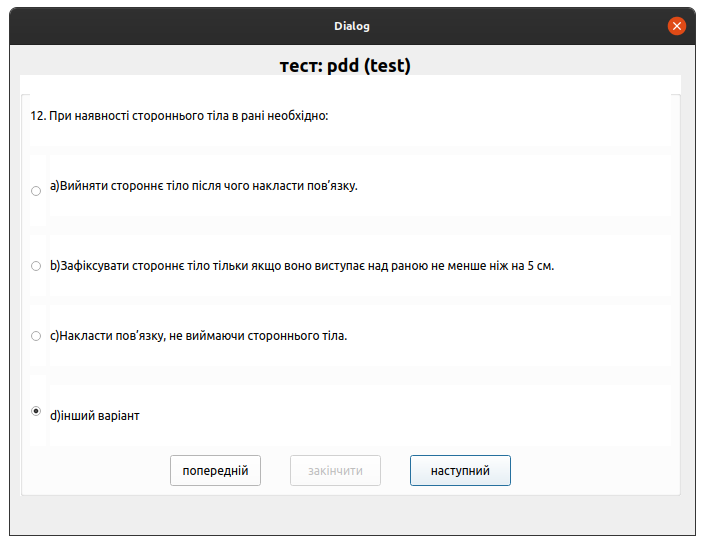
При натисканні на кнопку ‘видалити тест’ відкриється вікно видалення тесту. В якому потрібно ввести назву тесту з таблиці, яка розташована зліва, та натиснути кнопку ‘видалити’, після чого тест буде видалено. На рисунку 3.7 показано процес видалення тесту.

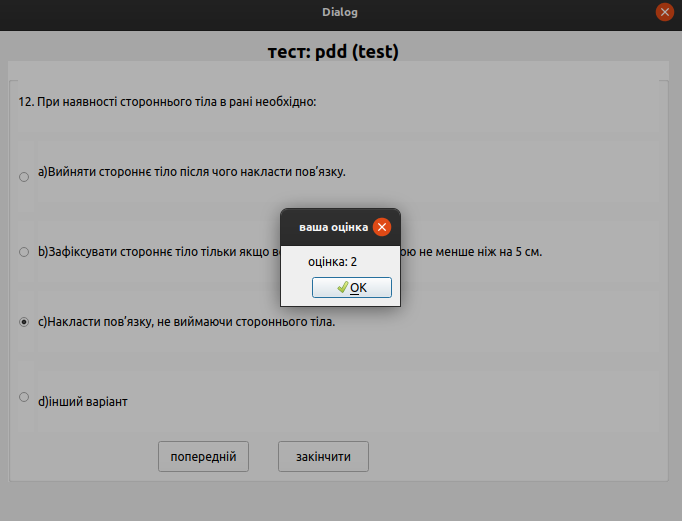
Рисунок 3.7 — Видалення тесту

Якщо авторизуватись, як користувач, чи створити нового користувача, відкриється вікно інтерфейсу користувача, в якому користувач, може пройти тести, та подивитись свої оцінки за проходження тестів. Вікно інтерфейсу зображено на рисунку 3.8. Щоб почати проходження тесту потрібно вписати ім’я тесту, які зображені у таблиці зліва, та натиснути кнопку ‘почати тест’, для виведення своїх оцінок потрібно, натиснтуи на кнопку ‘показати оцінки’, після чого зправа знизу з’явитьтся список, з оцінками.

Рисунок 3.8 — Інтерфейс користувача

Вікно проходження тесту складається з: умови, 4 варіантів відповіді, та трьх кнопок, ‘наступний’ - при натисканні користувач переходить до наступного питання, ‘попередній’ - при натисканні користувач переходить до попереднього питання, ‘закінчити’ - ця кнопка не доступна до того моменту поки користувач не дійде до останнього питання — при натисненні тест завершується, виводиться вікно з набраним балом, та вікно проходження закривається. Процес проходження зображено на рисунках 3.9 та 3.10.

Рисунок 3.9 — проходження тесту

Рисунок 3.10 — виведення результату

ВИСНОВКИ

У даному курсовому проєкті було описано етапи розробки програми “test-system”, яка призначена для створення бази тестових завдань згідно діючих Правил дорожнього руху для проходження тренувальних вправ під час навчання в автошколі та ведення контролю за якістю знань з боку адміністратора.

Розроблена програма відповідає усім поставленим вимогам технічного завдання. Під час виконання курсової роботи було закріплено і поглиблено знання, отримані під час вивчення предмету «Об’єктно-орієнтоване програмування», більш детально вивчено мову програмування С++ і практично застосовано бібліотеку STL. Для розробки даного програмного забезпечення використовувались принципи і концепції об’єктно-орієнтованого програмування. Використане середовище програмування VisualStudio2012, його особливі можливості і переваги, що дозволило швидко конструювати програмний засіб. Також під час розробки використано стандартну бібліотеку STL, зокрема контейнери типу вектор і множина, що дозволило в значній мірі пришвидшити розробку даної програми.

ПЕРЕЛІК ПОСИЛАНЬ

1. Грицюк Ю.І. Рак Т.Є. Об’єктно-орієнтоване програмування мовою С++ : навч. посіб. Львів : Вид-во Львівського ДУ БЖД, 2011. 404 с.
2. Вайсфельд М. Объектно-ориентированное мышление. СПб. : Питер, 2014. 304 с.
3. Build with Qt. URL: https://www1.qt.io/built-with-qt (дата звернення: 15.12.2019).
4. EVILEG : вебсайт. URL: https://evileg.com/uk/knowledge/qt (дата звернення: 15.12.2019).

Додаток А  
Діаграма варіантів використання

Додаток Б  
Лістинг файлу «program.h»

/////////////// Лістинги програм оформляти цим шрифтом!!!! ///////////////

/////////////////////підключення бібліотек//////////////////////

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <graphics.h>

#include <dos.h>

#include <math.h>

#include <dir.h>

/////////////// Лістинги програм оформляти цим шрифтом!!!! ///////////////

. . .

/////////////////////Оголошення класу ХХХ/////////////////////

. . .

Додаток В  
Лістинг файлу «program.cpp»

Додаток Г  
Компакт-диск із програмним продуктом