

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ  
ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Циклова комісія програмної інженерії

**КУРСОВИЙ ПРОЕКТ**  
з навчальної дисципліни  
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

на тему: «Програма ведення обліку задач НМТ»

(вказати тему курсового проекту)

Студента IV курсу ПЗ-21-1 групи  
галузь знань 12 «Інформаційні технології»  
спеціальності 121 «Інженерія програмного  
забезпечення»

Гуненко Я.М.

(прізвище та ініціали студента)

Керівник

викладач Гапоненко Н.В.

Національна шкала \_\_\_\_\_

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

Члени комісії \_\_\_\_\_

(підпис)

Любохинець В.М.

(прізвище та ініціали)

Ланська С.С.

(підпис)

(прізвище та ініціали)

Гапоненко Н.В.

(підпис)

(прізвище та ініціали)

м. Дніпро - 2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ  
ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Циклова комісія програмної інженерії

**ЗАТВЕРДЖУЮ**

Голова комісії ПІ

С.С.Ланська

«\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на виконання курсового проекту**

з дисципліни Об'єктно-орієнтоване програмування  
студенту Гуненко Ярославу Максимовичу  
(прізвище, ім'я та по батькові)  
Відділення Комп'ютерної та програмної інженерії  
Спеціальність 121 Інженерія програмного забезпечення  
Курс IV Група (шифр) ПЗ-21-1  
1 Тема проекту «Програма ведення обліку здачі НМТ»

2 Початкові дані Перелік даних про учня (ID учня, Номер паспорту, Тип ,  
Паспорту, ПІБ, Дата народження, Стать, Е-mai, Номер телефону, Номер  
свідоства про освіту, ПІН, Примітка), перелік навчальних закладів(місто,  
область, тип, Е-mail, ПІБ відповідального, Шифр навчального закладу)  
перелік предметів(Код предмету, Назва предмету, Опис, Зразок завдань)  
термін дії, статус) перелік сертифікатів(Номер сертифікату, ID учня, PIN,  
Дата створення, Статус) перелік результатів тесту(Шифр результату,  
ID умови, Отриманий бал, Статус здачі, Дата проходження, Шифр Статус  
навчального закладу) Умови проходження(ID умови, Код предмету,  
Максимальний бал, Мінімальний бал, Мінімальний бал для проходження,  
предмету, Дата ухвалення умов)

Розглянуто і ухвалено на засіданні циклової комісії програмної інженерії  
Протокол № 2 від 21.09.2021 р.

Керівник КП

Н.В.Гапоненко

(підпис)

(ініціали та прізвище)

Завдання до виконання  
одержав студент

(підпис)

(ініціали та прізвище)

Дата видачі

21 вересня 2021 р.

Термін виконання

22 листопада 2021 р.

## ЗМІСТ

ВСТУП.....	4
1 ПОСТАНОВКА ЗАДАЧІ.....	5
2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ .....	7
2.1 Опис середовища програмування .....	7
2.2 Опис мови програмування .....	8
2.3 Опис СКБД .....	10
2.4 Опис основних принципів ООП.....	12
2.5 Опис подібних програмних продуктів.....	14
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ .....	16
3.1 Опис бази даних.....	16
3.2 ПРОЕКТУВАННЯ КОРИСТУВАЧЬКОГО ІНТЕРФЕЙСУ ПРОГРАМИ .....	18

					КП.ПЗ.211.6.ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата	Програма ведення обліку здачі НМТ	Літ.	Арк.	Аркушів
Розроб.		Гуненко Я.М.						
Перевір.		Гапоненко Н.В.					3	24
Реценз.						ФКРКМ ДНУ ім. О.Гончара		
Н. контр.								
Затверд.								

## ВСТУП

У сучасному світі всюди потрібна автоматизація коректного збереження даних та ведення обліку цих даних. Не минула цифрова ера і сферу освіти. Щорічно в Україні після завершення одинадцяти класів учні проходять тестування, що відображає їх рівень знань отриманих за цей час. Наразі на заміну Зовнішньому Національному Оцінюванню через воєнний стан проводиться Національний Мультипредметний Тест, скорочено НМТ.

Існує багато програмного забезпечення для шкільних закладів але ці програми важкі у вивченні та незручні у використанні. Великий відсоток цих програм залежить від мережі інтернет, що не є завжди зручним. Програма повинна бути зручною у використанні, швидкою, легкою у вивченні її засобів та універсальною для багатьох великих та малих центрів оцінювання. Програмою може користуватися навіть не досвідчений користувач ПК. Нею може користуватися як і одна людина(наприклад, у невеликому центрі оцінювання) так і група людей(наприклад, у великому центрі оцінювання розділена робота на декілька посад).

Головною метою курсового проекту є створення зручного та простого додатку як і для великого центру оцінювання так і для малого.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1 ПОСТАНОВКА ЗАДАЧІ

### 1. Організація довідкової системи проходження тестів:

- Коди предметів.
- Опис предметів та зразки завдань.
- Умови проходження тесту:
- Код предмету.
- Мінімальний та максимальний бали.
- Статус обов'язковості тесту (обов'язковий чи ні).

### 2. Організація довідкової системи навчальних закладів:

- Шифр навчального закладу.
- Місто, область, контактна інформація.
- Відповідальний за додавання учнів та відповідальний за фіксацію проведення тесту.

### 3. Оформлення результатів тестування для студентів:

Реєстрація учня:

- Ідентифікаційні дані (номер паспорту, ПІБ, дата народження, ІПН).
- Контактні дані (телефон, email).
- Номер свідоцтва про освіту.

Внесення результатів тесту:

- Отриманий бал.
- Шифр навчального закладу та дата проходження.
- Генерація сертифікатів:
- Номер сертифікату.
- Термін дії та статус.

### 4. Вимоги до програмної системи:

- Операційна система: Windows 10 x64.
- Процесор: Intel Celeron або еквівалент.
- Графічна підсистема: DirectX 10 і вище.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Виділення інформаційних об'єктів предметної області

Учень	ID учня, номер паспорту, тип паспорту, ПІБ, дата народження, стать, E-mail, контактний номер телефону, Номер свідоцтва про освіту, Примітка, ПІН
Навчальний заклад	Шифр навчального закладу, Місто, Область, Тип, E-mail, Піб Відповідального за проведення
Предмет	Код предмету, Назва предмету, Опис, Зразок завдань
Умови проходження тестування з предмету	ID умови, Код предмету, Максимальний бал, Мінімальний бал, Мінімальний бал для проходження, Статус, Дата укладання вимог
Результат тестування	Шифр результату, ID умови, ID учня, отриманий бал, Статус здачі, Дата проходження, Шифр навчального закладу
Сертифікат учасника НМТ	Номер сертифікату, ID учня, PIN-код, дата створення, Термін дії, Статус дійсності

	Вик.	Гуменко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

### 2.1 Опис середовища програмування

C++ Builder— програмний продукт, інструмент швидкої розробки додатків (RAD), інтегроване середовище розробки (IDE), система, яка використовується програмістами для розробки програмного забезпечення на мові програмування C++.

На поверхні візуального дизайнера C++ Builder можна створювати графічні інтерфейси користувача, перетягуючи компоненти з палітри інструментів на форму. За допомогою дизайнерів можна створювати додатки Windows Forms, які використовують велику бібліотеку візуальних компонентів (VCL).

C++ Builder автоматично генерує більшу частину коду програми, як тільки ви починаєте роботу над проектом. Щоб допомогти вам завершити решту логіки програми, текстовий редактор коду надає такі функції, як рефакторинг, синхронізоване редагування, завершення коду, записані макроси натискання клавіш і призначені для користувача поєднання клавіш. Усередині IDE ви можете встановити параметри компілятора, скомпілювати і запустити додаток, а також переглянути повідомлення компілятора. В C++ Builder інтегровано MSBuild в якості рушія збірки, і команди компіляції та збірки викликають MSBuild.

RAD Studio забезпечує підтримку наступних мов програмування:

- Delphi
- C++

Відмінні риси C++ Builder та його можливості:

- Глибока інтеграція з C++:

Повний доступ до C++: C++ Builder надає безпосередній доступ до всіх можливостей мови C++, дозволяючи розробникам писати високопродуктивний і ефективний код.

- Підтримка стандартів:

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		



Середовище підтримує останні стандарти C++, забезпечуючи сучасний і ефективний розвиток програмного забезпечення.

– Оптимізація коду:

Вбудований компілятор здійснює агресивну оптимізацію коду, що призводить до високої швидкості виконання програм.

– Високопродуктивні бібліотеки:

VCL (Visual Component Library): Набір компонентів для створення багатих графічних інтерфейсів користувача. VCL забезпечує високу швидкість роботи і надійність.

FireDAC: Сучасна бібліотека доступу до даних, що підтримує широкий спектр баз даних.

RTL (Runtime Library): Бібліотека рутинного часу виконання, що містить безліч корисних функцій для роботи з рядками, файлами, математичними операціями та ін.

– Кросплатформна розробка:

Підтримка різних платформ: C++ Builder дозволяє створювати додатки для Windows, macOS, Android та iOS, використовуючи єдину кодову базу. Використання нативних компонентів: Програми, створені в C++ Builder, використовують нативні компоненти платформи, що забезпечує високу продуктивність і інтеграцію з операційною системою.

## 2.2 Опис мови програмування

У проєкті використано наступні мови програмування:

– C++.

C++ – це потужна, універсальна та широко використовувана мова програмування, яка забезпечує високий рівень контролю над ресурсами комп'ютера та дозволяє розробляти ефективні та надійні програмні системи.

Ключові особливості C++:

	Вик.	Гуменко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктно-орієнтоване програмування: C++ підтримує класичну модель ООП з класами, об'єктами, успадкуванням, поліморфізмом та абстракцією. Це дозволяє створювати модульні, розширювані та добре структуровані програми.

Підтримка процедурного програмування: Мова зберігає сумісність з C, дозволяючи використовувати процедурний стиль програмування для простих задач.

Шаблони: Шаблони (templates) забезпечують механізм генерації типобезпечного коду на основі параметрів типу. Це дозволяє створювати універсальні алгоритми та структури даних.

Стандартна бібліотека шаблонів (STL): STL надає багатий набір контейнерів (вектори, списки, карти тощо), алгоритмів (сортування, пошук, перетворення) та ітераторів, що спрощує розробку складних програм.

Низькорівневий доступ: C++ надає прямий доступ до пам'яті та апаратних ресурсів, що робить його ідеальним для системного програмування, драйверів пристроїв та інших програм, які вимагають високої продуктивності.

Основні концепції C++:

Типи даних: C++ має широкий спектр вбудованих типів даних (цілі числа, числа з плаваючою точкою, символи, булеві значення) та дозволяє створювати користувацькі типи даних за допомогою класів.

Змінні: Змінні використовуються для зберігання даних. Кожна змінна має свій тип, який визначає, які значення вона може приймати.

Оператори: Оператори використовуються для виконання різних дій над даними, таких як арифметичні операції, порівняння, логічні операції тощо.

Умови та цикли: Умови (if, else) дозволяють виконувати різні блоки коду залежно від умов, а цикли (for, while) дозволяють повторювати виконання блоків коду.

Функції: Функції використовуються для розбиття програми на окремі, повторно використовувані блоки коду.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Класи та об'єкти: Класи описують структуру даних та поведінку об'єктів, а об'єкти є екземплярами класів.

Переваги використання C++:

Висока продуктивність: C++ дозволяє створювати високопродуктивні програми, які ефективно використовують ресурси комп'ютера.

Контроль: Програміст має повний контроль над процесом виконання програми.

Портативність: Програми, написані на C++, можуть бути скомпільовані та виконані на різних платформах.

Широке спільнота: C++ має велику та активну спільноту розробників, що забезпечує велику кількість бібліотек, фреймворків та інструментів.

Довговічність: C++ є мовою з довгою історією, що гарантує її стабільність і розвиток.

Сфери застосування C++:

Системне програмування: Операційні системи (Windows, Linux), драйвери пристроїв, вбудовані системи.

Розробка ігор: Движки ігор (Unreal Engine, Unity), графічні бібліотеки (OpenGL, DirectX).

Наукові обчислення: Симуляції, моделювання, аналіз даних.

Бази даних: Системи управління базами даних (MySQL, PostgreSQL), сервери баз даних.

Веб-розробка: Сервери, фреймворки (Qt, Boost).

## 2.3Опис СКБД

MySQL — це система керування реляційними базами даних (СКБД) з відкритим вихідним кодом, яка широко використовується для створення, підтримки та управління базами даних. Вона була створена в 1995 році шведською компанією MySQL AB, заснованою Майклом Віденіусом, Девідом Аксмарксом і Аланом Ларсоном. Спочатку MySQL була розроблена як альтернатива комерційним СКБД, і завдяки своїй простоті використання, високій

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

продуктивності та надійності швидко здобула популярність серед розробників. У 2008 році MySQL була придбана компанією Sun Microsystems, а в 2010 році — Oracle Corporation, яка продовжує розвивати систему.

Основні можливості MySQL включають підтримку транзакцій, реплікації, шардінгу та масштабування. Вона дозволяє обробляти одночасні запити від тисяч користувачів і підтримує різні типи таблиць і механізми зберігання даних, зокрема InnoDB та MyISAM. MySQL також підтримує широкий спектр мов програмування, таких як PHP, Java, Python, C++, що робить її ідеальним вибором для створення динамічних веб-застосунків та корпоративних систем. Вона відповідає стандарту SQL (Structured Query Language), що дозволяє ефективно працювати з реляційними даними та виконувати складні запити.

З випуском MySQL 5.5 система отримала кілька суттєвих поліпшень, спрямованих на підвищення масштабованості та швидкодії:

InnoDB за замовчуванням: InnoDB став основним рушієм збереження даних за замовчуванням, що забезпечує підтримку транзакцій, цілісність даних через механізм зовнішніх ключів та кращу продуктивність для великих баз даних.

Напівсинхронна реплікація: додано підтримку напівсинхронного механізму реплікації, заснованого на патчах від Google, що підвищує надійність реплікації в кластерних середовищах.

Партіціювання даних: поліпшені функції з партіціювання даних, включаючи розбиття великих таблиць на кілька частин (partitioning) з підтримкою RANGE, LIST і оптимізації «partition pruning», що дозволяє ефективно працювати з великими обсягами даних.

Оптимізація запитів: нові механізми оптимізації вкладених запитів та операцій JOIN значно підвищують продуктивність при роботі з великими базами даних.

Блокування та багатопоточність: перероблена система внутрішніх блокувань та підтримка багатоядерних процесорів забезпечує більш ефективну роботу в багатопоточних середовищах.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

MySQL продовжує розвиватися і залишається однією з найбільш популярних СКБД у світі завдяки своїй гнучкості, швидкодії, надійності та сумісності з сучасними хмарними технологіями.

## 2.4 Опис основних принципів ООП

Об'єктно-орієнтоване програмування (ООП) — це підхід до програмування, що базується на концепції об'єктів і взаємодії між ними. Основними принципами ООП є інкапсуляція, наслідування, поліморфізм і абстракція. У мові програмування C++ ці принципи широко використовуються для побудови структурованих та гнучких програмних рішень. Розглянемо кожен із принципів докладніше.

### 1. Інкапсуляція

Інкапсуляція — це процес об'єднання даних і методів, які працюють з цими даними, в один об'єкт. Основна ідея полягає в приховуванні деталей реалізації від зовнішнього світу, що дозволяє захистити внутрішній стан об'єкта від некоректних змін. У C++ це реалізується через використання модифікаторів доступу:

`private`: доступ до даних і методів обмежений лише межами класу, що дозволяє приховати їх від зовнішнього використання.

`protected`: доступний в межах класу та його нащадків.

`public`: доступний для всіх об'єктів і функцій.

Інкапсуляція дозволяє:

Забезпечити безпеку даних, приховавши їх від прямого доступу зовнішніх об'єктів.

Зменшити залежність між компонентами програми, що полегшує її підтримку та модифікацію.

Контролювати доступ до змінних через інтерфейси класу (публічні методи).

### 2. Наслідування

Наслідування — це механізм, що дозволяє створювати нові класи на основі існуючих, успадковуючи їхні властивості та методи. У C++ наслідування дозволяє

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

повторно використовувати код і спрощує створення ієрархій класів. Нові класи, які наслідують інші, називаються похідними, а класи, від яких вони наслідують, — базовими.

Ключові аспекти наслідування:

Повторне використання коду: похідні класи можуть успадковувати і використовувати методи та атрибути базових класів, що дозволяє уникнути дублювання коду.

Розширення функціональності: похідні класи можуть додавати нові методи або перевизначати існуючі, щоб змінювати поведінку базового класу.

Типи наслідування в C++:

public: всі публічні та захищені члени базового класу залишаються такими ж у похідному класі.

protected: публічні члени базового класу стають захищеними у похідному.

private: всі члени базового класу стають приватними у похідному.

Наслідування дозволяє будувати логічні зв'язки між класами, створюючи гнучкі й масштабовані ієрархії.

### 3. Поліморфізм

Поліморфізм дозволяє методам і об'єктам виконувати одну й ту саму операцію по-різному, залежно від контексту. У C++ поліморфізм може бути реалізований двома способами: компіляційний (статичний) та рантаймовий (динамічний).

Статичний поліморфізм: це реалізується через перевантаження методів і операторів. Наприклад, функції з однією назвою можуть мати різні параметри і виконувати різні дії залежно від того, як вони викликаються.

Динамічний поліморфізм: досягається через використання віртуальних функцій та успадкування. Він дозволяє об'єктам різних типів обробляти виклики однаково через спільний інтерфейс (зазвичай це базовий клас). В C++ віртуальні функції дозволяють вирішити, яку реалізацію методу викликати під час виконання програми залежно від типу об'єкта.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Поліморфізм забезпечує:

Гнучкість у реалізації та використанні методів.

Можливість створення узагальнених рішень, які можуть працювати з різними типами даних або об'єктів.

Спрощення підтримки та розширення системи за рахунок можливості підміни методів у похідних класах.

#### 4. Абстракція

Абстракція полягає у відокремленні важливих деталей реалізації від зовнішнього інтерфейсу об'єкта. Це дозволяє програмістам працювати лише з тими аспектами об'єкта, які мають значення для виконання конкретного завдання, приховуючи другорядні або складні деталі. В C++ абстракція реалізується через абстрактні класи та чисті віртуальні функції.

Абстрактний клас — це клас, який не може бути інстанційованим (не можна створити об'єкт цього класу), оскільки він містить хоча б одну чисту віртуальну функцію, яка не має реалізації. Похідні класи повинні реалізовувати ці чисті віртуальні функції.

Абстракція дозволяє:

Спрощувати взаємодію між компонентами програми, визначаючи лише важливі аспекти взаємодії.

Полегшувати зміну реалізації без впливу на інші частини програми.

Забезпечувати чіткий контракт між базовими і похідними класами.

#### 2.5 Опис подібних програмних продуктів

Схожою за функціоналом є система [my.testportal.gov.ua](http://my.testportal.gov.ua).

[my.testportal.gov.ua](http://my.testportal.gov.ua) — це веб-портал, призначений для організації та обліку результатів національних тестів. Система працює виключно онлайн, що означає, що будь-яка перерва в доступі до інтернету може спричинити затримку в роботі, оскільки програма не підтримує роботу в автономному режимі.

Платформа використовує власну базу даних для збереження інформації про учнів, навчальні заклади та результати тестувань. Проте, для більш гнучкої

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

обробки даних немає можливості прямої інтеграції з СКБД на стороні користувача. Дані можна експортувати у форматі PDF, але це обмежує можливості подальшого аналізу та структуризації інформації.

Платформа також вимагає значного часу на налаштування, особливо для нових користувачів або навчальних закладів. Потрібен спеціаліст для розподілу ролей та доступів між адміністраторами, вчителями та учнями, оскільки самостійно це зробити складно без відповідного досвіду.

Висновок: [my.testportal.gov.ua](http://my.testportal.gov.ua) підходить для великих організацій, що мають відповідні ресурси для підтримки та адміністрування системи. Однак, для невеликих закладів або окремих користувачів платформа може виявитися занадто складною та вимагати значних зусиль для ефективного використання.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		



## 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

### 3.1 Опис бази даних

Для збереження великих об'ємів інформації потрібно розробити низку правил та відношень між цими даними. Відношення між таблицями показані на ER-діаграмі. Скрипт створення БД показаний у додатку Б. Зміст таблиць бази даних показаний у таблицях 3.2-3.9.

Таблиця 3.2 – Поля таблиці «Студент» (Student)

Назва поля	Опис	Тип	Розмір	Ключ
Student_id	Унікальний ідентифікатор студента	Числовий		Первинний
Passport_num	Номер паспорта	Символьний	До 20 символів	
Passport_type	Тип паспорта	Символьний	До 10 символів	
PIB	Прізвище, ім'я, по-батькові	Символьний	До 100 символів	
Birth_date	Дата народження	DATE	Стандарт ISO	
Gender	Стать	Символьний	1 символ	
E-mail	Електронна пошта	Символьний	До 100 символів	
Phone_num	Номер телефону	Символьний	До 15 символів	
EduCerf_num	Номер освітнього сертифіката	Символьний	До 30 символів	
PN	Ідентифікаційний код	Символьний	10 символів	
Additional	Додаткова інформація	Текстовий		

Таблиця 3.3 – Поля таблиці «Користувачі» (Users)

Назва поля	Опис	Тип	Розмір	Ключ
Login	Логін користувача	Символьний	До 50 символів	Первинний
Password	Пароль користувача	Символьний	До 50 символів	
Role	Роль користувача	Символьний	До 20 символів	

Таблиця 3.4 – Поля таблиці «Результат» (Result)

Назва поля	Опис	Тип	Розмір	Ключ
Res_id	Унікальний код результату	Числовий	Макс. значення – 2147483647	Первинний
Subj_id	Код предмета	Числовий		Зовнішній
Condition_id	Код умов	Числовий		Зовнішній
Student_id	Код студента	Числовий		Зовнішній
Reached_score	Досягнутий бал	Числовий		
Status	Статус	Логічний		
School_id	Код навчального закладу	Числовий		Зовнішній
Attemp_date	Дата спроби	DATE	Стандарт ISO	

Таблиця 3.5 – Поля таблиці «Сертифікат» (Certificate)

Назва поля	Опис	Тип	Розмір	Ключ
Cerf_num	Номер сертифіката	Числовий	Макс. значення – 2147483647	Первинний
Student_id	ID студента	Числовий		Зовнішній
PIN	PIN-код	Символьний	До 15 символів	
Creation_date	Дата створення	DATE	Стандарт ISO	
Efect_time	Термін дії	DATE	Стандарт ISO	
Status	Статус сертифіката	Логічний		

Таблиця 3.6 – Поля таблиці «Умови тестування» (Condition)

Назва поля	Опис	Тип	Розмір	Ключ
Condition_id	Унікальний код умови	Числовий	Макс. значення – 2147483647	Первинний
Subject_id	Код предмета	Числовий		Зовнішній
Max_point	Максимальний бал	Числовий		
Min_r_point	Мінімальний бал для проходження	Числовий		
Min_point	Мінімальний бал	Числовий		
Status	Обов'язкова умова	Логічний		
Date	Дата	DATE	Стандарт ISO	

Таблиця 3.7 – Поля таблиці «Умови тестування» (Condition)

Назва поля	Опис	Тип	Розмір	Ключ
School_id	Унікальний код навчального закладу	Числовий	Макс. значення – 2147483647	Первинний
City	Місто	Символьний	До 50 символів	
Region	Область	Символьний	До 50 символів	
Type	Тип навчального закладу	Символьний	До 30 символів	
E-mail	Електронна пошта	Символьний	До 100 символів	
Teacher_PIB	Прізвище, ім'я, по-батькові відповідального	Символьний	До 100 символів	

Таблиця 3.8– Поля таблиці «Предмети» (Subject)

Назва поля	Опис	Тип	Розмір	Ключ
Subject_id	Унікальний код предмета	Числовий	Макс. значення – 2147483647	Первинний
Name	Назва предмета	Символьний	До 100 символів	
Description	Опис предмета	Текстовий		

ER-діаграма зв'язку таблиць бази даних наведена нижче на рисунку 3.1.

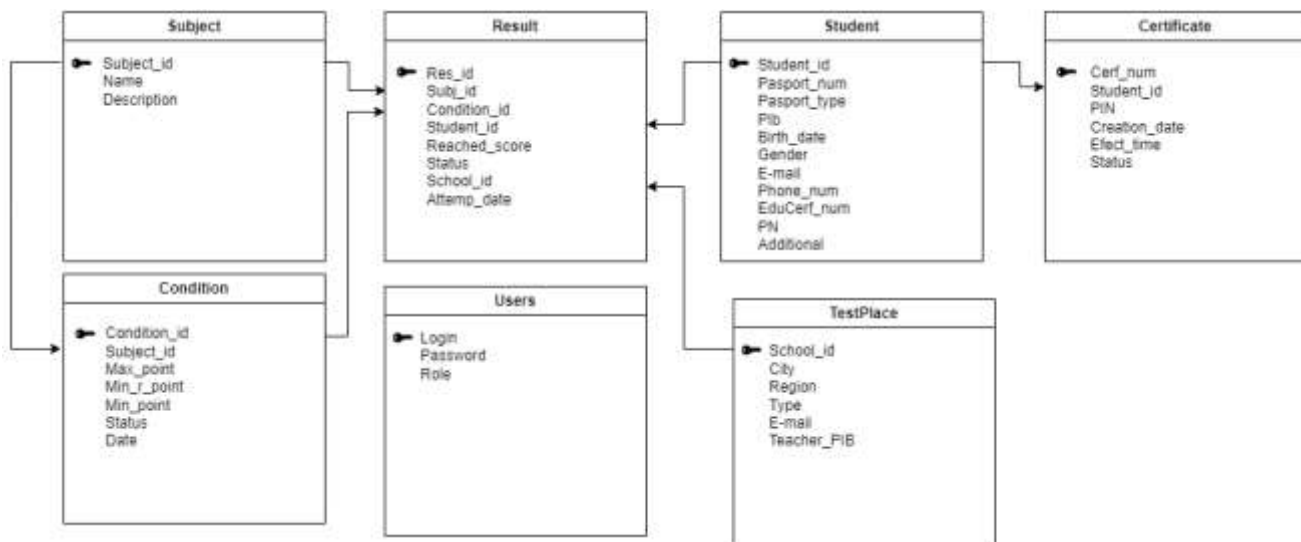


Рисунок 3.1 – ER діаграма бази даних

### 3.2 Проектування користувацького інтерфейсу програми

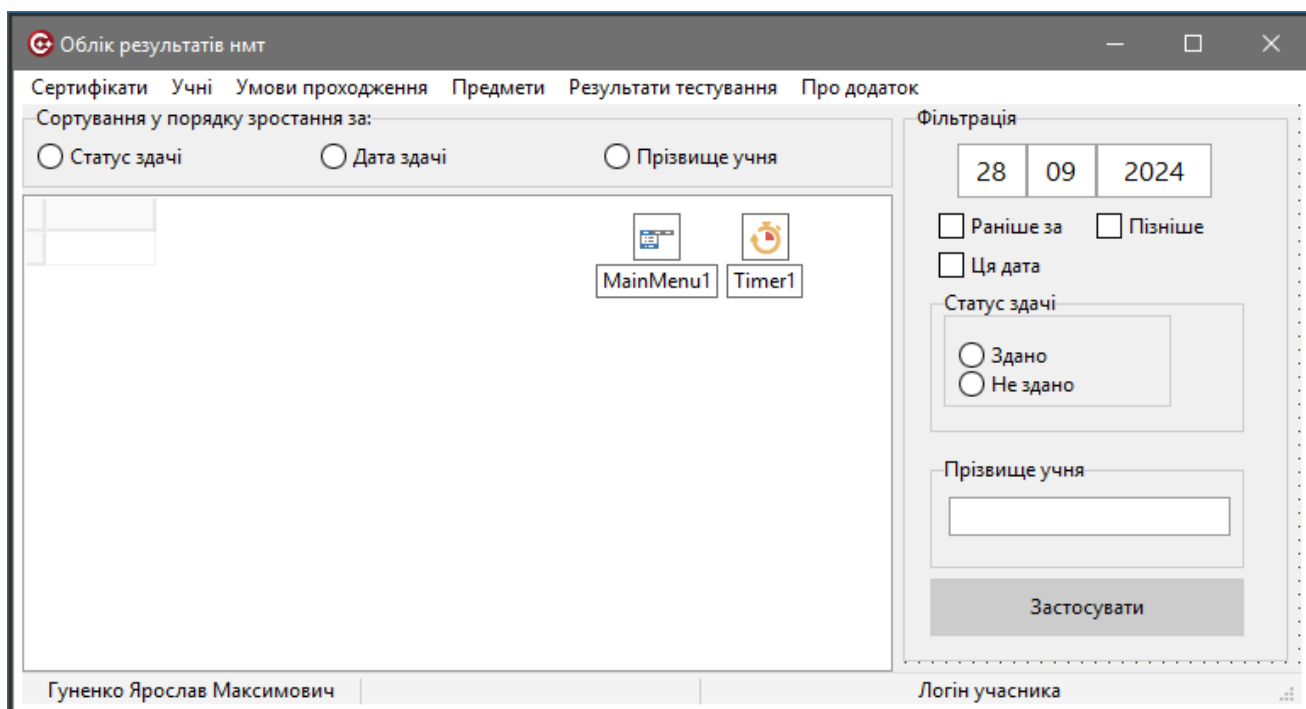


Рисунок 3.2 – Дизайн головного вікна

Додавання інформації про учня

Сертифікати
Учні
Умови проходження
Предмети
Результати тестування
Про додаток

Поля для введення інформації про студента

ID учня(студента)

0

Стать

☐ Чоловіча
☐ Жіноча

Дата народження

28

09

2024

Помітка(\*)

Додати

Номер паспорту

ІПН

Но MainMenu1 про освіту

E-mail

Контактний телефон

Рисунок 3.3 – Додавання інформації про учнів

Редагування інформації про учнів

Сертифікати
Учні
Умови проходження
Предмети
Результати тестування
Про додаток

Поля для введення інформації про студента

ID учня(студента)

Стать

☐ Чоловіча
☐ Жіноча

Дата народження

28

09

2024

Помітка(\*)

Внести зміни

Номер паспорту

ІПН

Но MainMenu1 про освіту

E-mail

Контактний телефон

Рисунок 3.4 – Редагування інформації про учнів

Вик.	Гуненко Я.М.				КП.ПЗ.211.6.ПЗ	Арк.
Пер.	Гапоненко Н.В.					19
Змн.	Арк.	№ докум.	Підпис	Дата		

Рисунок 3.5 – Видалення інформації про учнів

Рисунок 3.6 – Додавання інформації про предмети

Рисунок 3.7 – Редагування інформації про предмети

	Вик.	Гуменко Я.М.			КП.ПЗ.211.6.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

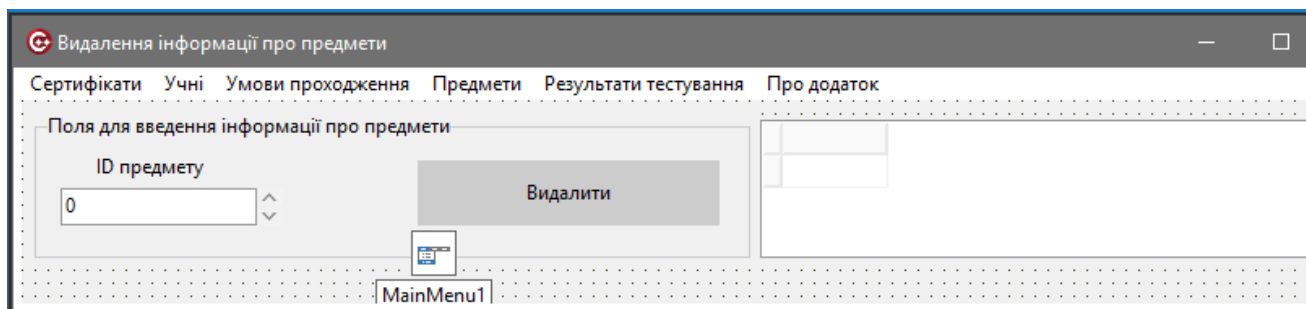


Рисунок 3.8 – Видалення інформації про предмети

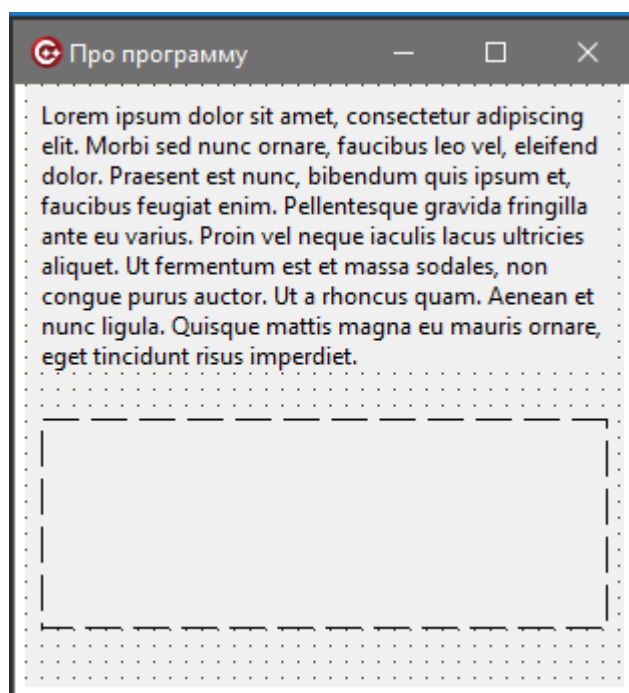


Рисунок 3.9 – Інформація про додаток

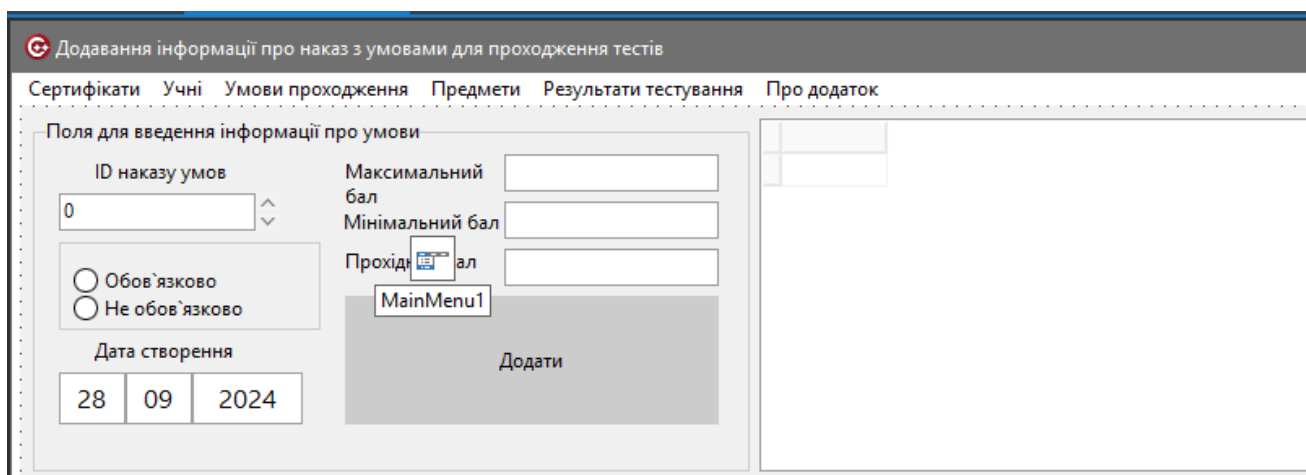


Рисунок 3.10 – Додавання інформації про умови проходження тестів

Редагування інформації про наказ з умовами для проходження тестів

Сертифікати Учні Умови проходження Предмети Результати тестування Про додаток

Поля для введення інформації про умови

ID наказу умов

0

Максимальний бал

Мінімальний бал

Прохідний бал

Обов'язково

Не обов'язково

Дата створення

28 09 2024

Замінити

Рисунок 3.10 – Редагування інформації про умови проходження тестів

Редагування інформації про наказ з умовами для ...

Сертифікати Учні Умови проходження Предмети

Результати тестування Про додаток

Поля для введення інформації про умови

ID наказу умов

0

Видалити

MainMenu1

Рисунок 3.11 – Видалення інформації про умови проходження тестів

Вик.	Гуненко Я.М.				КП.ПЗ.211.6.ПЗ	Арк.
Пер.	Гапоненко Н.В.					22
Змн.	Арк.	№ докум.	Підпис	Дата		

Додавання інформації про результати

Сертифікати
Учні
Умови проходження
Предмети
Результати тестування
Про додаток

Поля для введення інформації про результати

ID результату

0

ID предмета

0

ID учня(студента)

0

ID умов

0

MainMenu1

ID учня(студента)

0

Здано
Не здано

Дата складання

28

09

2024

Додати

Рисунок 3.11 – Додавання інформації про результати тестів

Редагування інформації про результати

Сертифікати
Учні
Умови проходження
Предмети
Результати тестування
Про додаток

Поля для введення інформації про результати

ID результату

0

ID предмета

0

ID учня(студента)

0

ID умов

0

MainMenu1

ID учня(студента)

0

Здано
Не здано

Дата складання

28

09

2024

Додати

Рисунок 3.12 – Редагування інформації про результати тестів



Видалення інформації про результати

Сертифікати   Учні   Умови проходження   Предмети   Результати тестування   Про додаток

Поля для введення інформації про результати

ID результату

0

Видалити

MainMenu1

Рисунок 3.13 – Видалення інформації про результати тестів

Сертифікати

Сертифікати   Учні   Умови проходження   Предмети   Результати тестування   Про додаток

MainMenu1

Поля для введення інформації про сертифікати

ID наказу умов

0

PIN-код

Термін дії

ID студента

0

Дата створення

28 09 2024

Додати

Сгенерувати сертифікат

ID студента

0

Згенерувати

Рисунок 3.14 – Видалення інформації про результати тестів

Вик.	Гуненко Я.М.				КП.ПЗ.211.6.ПЗ	Арк.
Пер.	Гапоненко Н.В.					24
Змн.	Арк.	№ докум.	Підпис	Дата		



```

CREATE DATABASE NMT_results;
USE NMT_results;
-- Створення таблиці "Student"
CREATE TABLE Student (
    Student_id INT PRIMARY KEY,
    Passport_num VARCHAR(20),
    Passport_type VARCHAR(10),
    PIB VARCHAR(100),
    Birth_date DATE,
    Gender CHAR(1),
    Email VARCHAR(100),
    Phone_num VARCHAR(15),
    EduCerf_num VARCHAR(30),
    PN VARCHAR(10),
    Additional TEXT
);

-- Створення таблиці "Users"
CREATE TABLE Users (
    Login VARCHAR(50) PRIMARY KEY,
    Password VARCHAR(50),
    Role VARCHAR(20)
);

-- Створення таблиці "Result"
CREATE TABLE Result (
    Res_id INT PRIMARY KEY,
    Subj_id INT,
    Condition_id INT,
    Student_id INT,
    Reached_score INT,
    Status BOOLEAN,
    School_id INT,
    Attemp_date DATE,
    FOREIGN KEY (Subj_id) REFERENCES Subject(Subject_id),
    FOREIGN KEY (Condition_id) REFERENCES Condition(Condition_id),
    FOREIGN KEY (Student_id) REFERENCES Student(Student_id),
    FOREIGN KEY (School_id) REFERENCES School(School_id)
);

-- Створення таблиці "Certificate"
CREATE TABLE Certificate (
    Cerf_num INT PRIMARY KEY,
    Student_id INT,
    PIN VARCHAR(15),
    Creation_date DATE,
    Efect_time DATE,
    Status BOOLEAN,
    FOREIGN KEY (Student_id) REFERENCES Student(Student_id)
);

-- Створення таблиці "Condition"
CREATE TABLE Condition (
    Condition_id INT PRIMARY KEY,
    Subject_id INT,
    Max_point INT,

```

```

        Min_r_point INT,
        Min_point INT,
        Status BOOLEAN,
        Date DATE,
        FOREIGN KEY (Subject_id) REFERENCES Subject(Subject_id)
    );

-- Створення таблиці "School"
CREATE TABLE School (
    School_id INT PRIMARY KEY,
    City VARCHAR(50),
    Region VARCHAR(50),
    Type VARCHAR(30),
    Email VARCHAR(100),
    Teacher_PIB VARCHAR(100)
);

-- Створення таблиці "Subject"
CREATE TABLE Subject (
    Subject_id INT PRIMARY KEY,
    Name VARCHAR(100),
    Description TEXT
);

```