МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп`ютерної інженерії та кібербезпеки

**КУРСОВА РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об’єктно-орієнтоване програмування»

на тему:

**«Інформаційно-пошукова система: успішність»**

студента І курсу групи КБ-21-2

спеціальності 125 «Кібербезпека»

\_\_\_\_\_Кучера Віталія Віталійовича\_\_\_\_\_\_\_\_

(прізвище, ім’я та по-батькові)

Керівник ст. викладач кафедри КН\_\_\_\_\_\_\_

, Левківський В.Л.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Л. Левківський \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Марчук .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Граф \_

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра комп`ютерної інженерії та кібербезпеки

Освітній рівень: бакалавр

Спеціальність 125 «Кібербезпека»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри КІ та КБ

\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Єфіменко

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТУ

Кучеру Віталію Віталійовичу

1. Тема роботи: «Розробка інформаційно-пошукової системи: успішність»,

керівник роботи: старший викладач кафедри комп’ютерних наук Левківський Віталій Леонідович.

1. Строк подання студентом: “ 15 ” липня 2022р.
2. Вихідні дані до роботи: Розробити інформаційно-пошукову систему: успішність.
3. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до КР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Посилання на репозиторій: https://gitlab.com/kuchervitalii/course2\_\_\_\_\_\_\_\_\_\_

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 15 ” квітня 2022 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Постановка задачі | 28.04.2022 |  |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 05.05.2022 |  |
| 3 | Формулювання технічного завдання | 30.05.2022 |  |
| 4 | Опрацювання літературних джерел | 03.06.2022 |  |
| 5 | Проектування структури | 09.06.2022 |  |
| 6 | Написання програмного коду | 01.07.2022 |  |
| 7 | Відлагодження | 10.07.2022 |  |
| 8 | Написання пояснювальної записки | 15.07.2022 |  |
| 9 | Захист |  |  |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кучер В.В.

(підпис) (прізвище та ініціали)

**Керівник роботи**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Левківський В.Л.

(підпис) (прізвище та ініціали)

**РЕФЕРАТ**

Завданням на курсову роботу було створення інформаційно-пошукової системи для перегляду поточної успішності.

Пояснювальна записка до курсової роботи на тему «Інформаційно-пошукова система: успішність» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 36 сторінках друкованого тексту.

Пояснювальна записка має 18 сторінок додатків. Список використаних джерел містить 12 найменувань і займає 2 сторінки. В роботі наведено 32 рисунка. Загальний обсяг роботи – 55 сторінок.

Ключові слова: C#, MySQL, WINDOWS FORMS, ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА, УСПІШНІСТЬ, БАЗА ДАНИХ, БД.

**ЗМІСТ**

ВСТУП 6

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДИ, ТА ЗАСОБИ ВИРІШЕННЯ 8

1.1 Аналіз задачі, засобів та методів її вирішення 8

1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи 9

Висновки до першого розділу 11

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 12

2.1 Проектування загального алгоритму роботи програми 12

2.2 Розробка функціональних алгоритмів роботи програми 13

2.3 Розробка програмного забезпечення 19

Висновки до другого розділу 21

РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ 22

3.1 Опис роботи з програмним додатком 22

3.2 Тестування роботи програмного забезпечення 31

Висновки до третього розділу 33

ВИСНОВКИ 34

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 35

ДОДАТКИ 37

# 

# ВСТУП

**Актуальність теми:** пошук інформації є однією з основних складових людської діяльності, з яким ми зіштовхуємося щодня: перегортаючи телефонну книгу, підбираючи в розкладі поїздів електричку, вивчаючи афішу кінотеатра, щоб вибрати по істинні цікавий фільм... Людині, яка в силу своєї професії чи захоплень часто зіштовхується з підбором і пошуком якої-небудь тематичної інформації, рано чи пізно (зі зростанням її обсягу) приходиться застосовувати деякі принципи систематизації і класифікації наявних даних, що забезпечують більш зручний і ефективний пошук. Так, у бібліотеках складають картотеку: відомості про книгу за визначеною схемою записуються на картку, туди ж міститься шифр - кілька букв і цифр, по яких можна визначити місце розташування книги (сховище, стелаж, полицю); картки розставляються в алфавітному чи тематичному порядку. Застосування інформаційно-пошукових систем дає більш широкі можливості для роботи з великими масивами інформації.

**Мета дослідження:** вдосконалення навичок програмування мовою С# та створення програмного забезпечення на тему “Інформаційно-пошукова система: успішність”. Встановлена мета має забезпечити та реалізувати наступні функції:

− Програма має містити реєстрацію та авторизацію, здійснювати перехід між різними формами.

− Виведення даних на екран в зручному вигляді;

− Дані мають зберігатися у базі даних.

− Пошук даних (за всіма параметрами).

− Редагування даних. Перед редагуванням знайти потрібний запис.

− Видалення даних. Перед видаленням знайти потрібний запис.

− Додавання нових записів.

**Об’єкт дослідження:** технологія роботи з формами, класами, базою даних.

**Предмет дослідження:** робота з класами та наслідуванням, базою даних, конструкторами класів, формами, функціями та бібліотеками класів.

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДИ, ТА ЗАСОБИ ВИРІШЕННЯ

## Аналіз задачі, засобів та методів її вирішення

Поставлено завдання створити інформаційно-пошукову систему на тему “ Успішність”. Для зберігання інформації будемо використовувати базу даних, а саме MySql для створення бази абітурієнтів, зареєстрованих користувачів та оцінок. Для виконання цього завдання програма має містити та вміти виконувати такі функції:

− Зчитувати інфомацію з бази даних;

Для реалізації буде створено метод, який прийматиме дані.

− Заносити зміни в базу даних;

Для реалізації під час роботи будемо заносити корективи в базу даних.

Програма має вміти виконувати такі функції з базою даних:

− Виведення даних;

Для реалізації будемо використовувати елемент WindowsForms DataGridView.

− Пошук даних;

Для реалізації буде створено метод, який шукатиме схожу інформацію в таблиці.

− Редагування даних. Перед редагуванням знайти потрібний запис;

Для реалізації буде створено метод, який даватиме користувачеві редагувати конкретний запис.

− Видалення даних. Перед видаленням знайти потрібний запис;

Для реалізації буде створено метод, який видалятиме обраний користувачем запис.

− Додавання нового запису;

Для реалізації буде створено метод, який додаватиме новий запис користувача.

− Реєстрація і авторизація користувача;

Це потрібно для індентифікації користувача і надання йому можливостей: звичайний юзер має можливість тільки переглядати шукати потрібні записи, адмін ж матиме змогу редагувати, видаляти та додавати записи, бачити більш детальну інформацію.

− Зручне переміщення між формами;

Для реалізації буде створено взаємозв`язок між усіма формами.

## 1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи

Аналогічними системами пошуку користуються навчальні заклади. Такі інформаційно-пошукові системи зазвичай використовуються в додатках.

1) Один із прикладів є додаток Єдина Школа, автором якої є компанія TATL technology(https://eschool-ua.com).

Ця система має доволі широкий спектр можливостей: можливість спілкуватися через чати, перегляд поточної успішності, показ статистики по оцінкам, перкгляд розкладу та власного профілю.

Перевагою цієї системи є розширений функціонал, а мінусом – відсутність веб-сайту, перегляд оцінок інших користувачів не є можливим, відсутність сортування та вікна для реєсстрації.

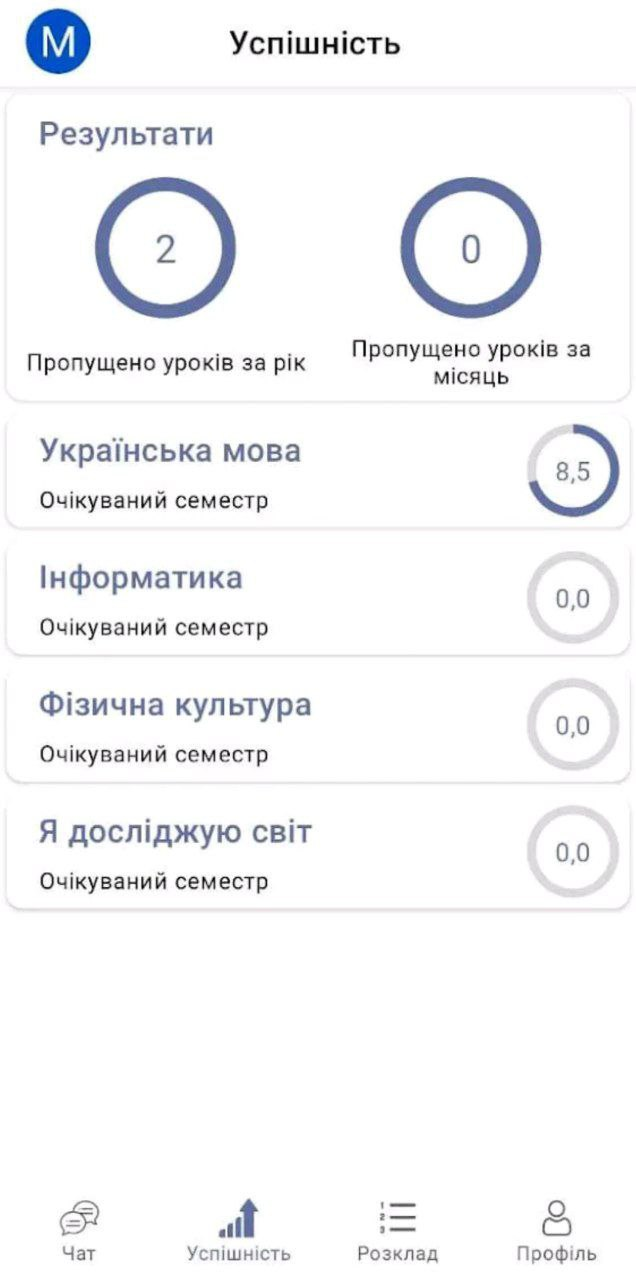


Рис. 1.1. Інтерфейс системи пошуку – “Успішність”

2) Іншим аналогом можна вважати інформаційну систему пошуку для перегляду успішності HІТ, автором якого є компанія Lionwood.Software(https://nit.school).

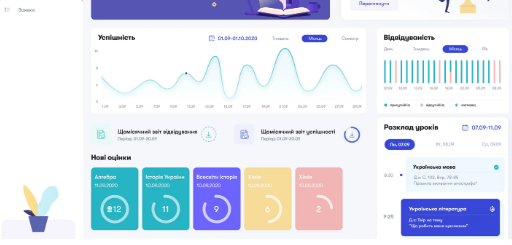


Рис. 1.2. Інтерфейс інформаційно-пошукової системи – “ HІТ ”

Ця система пошуку містить аналітику успішності,електронний щоденник, динаміку успішності попредметно, реєстрацію та авторизацію.

Основною перевагою цієї програми є зручне управління, яке дає змогу швидко адаптуватися до функціоналу. Недоліком ж цієї системи пошуку є відсутність сортування та залежність від інтернету.

3) Третім аналогом систему пошуку ain, автором якої є компанія ain [( https://ain.ua/special/meet-eschool/).](https://www.gputracker.eu/en/terms)

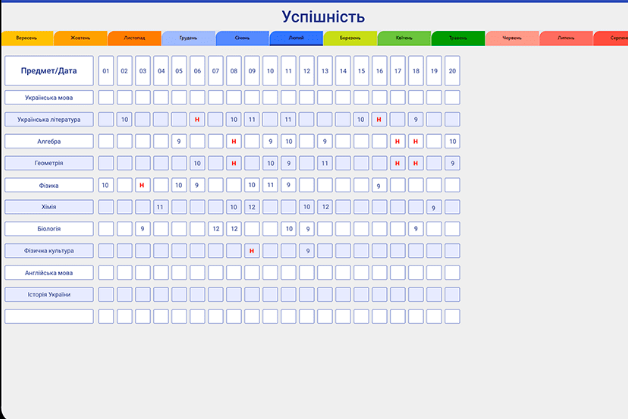


Рис. 1.3. Інтерфейс системи пошуку – “Ain”

Cистема має доволі широкий спектр можливостей: можливість переглядати поточну успішність, перегляд всіх оцінок помісячно.

Перевагою цієї системи є доволі простий функціонал, мінусом ж відсутність веб-сайту, відсутність сортування, підрахунок середньго балу та вікна для реєсстрації .

Отже, основним напрямом розробки програмного продукту є створення інформаційно-пошукової системи для роботи з базою даних успішності.

## Висновки до першого розділу

Таким чином, було визначено завдання і проведено його аналіз. Знайдено методи вирішення задачі та можливість їх реалізації. Проведено аналіз аналогів, знайдено їх недоліки та переваги, завдяки цьому визначено напрям розробки програмного продукту.

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1 Проектування загального алгоритму роботи програми

Відкривавши програму, користувача зустріне вікно авторизації, де зможе авторизуватися або зареєструватися та виконувати всі потрібні дії в подальшому (рис. 2.1).



Рис. 2.1. Загальна схема робота програми

На схемі зображено дії, які користувач може виконувати. Зустрічає його панель авторизації, з якої він може авторизуватися перейти до панелі реєстрації. Якщо користувача буде авторизовано він перейде до панелі користувача або панелі адміна. Адміністратор може редагувати, переглядати, видаляти, шукати та створювати записи, переглядати список зареєстрованих користувачів. А ось звичайний користувач переглядати детальну інформацію щодо поточної успішності та шукати необхідну йому інформацію.

Для кращого розуміння переходу між функціоналом складемо детальну схему варіантів використання програми (рис. 2.2). На цій схемі детальніше розписано шляхи та умови переходу між формами.

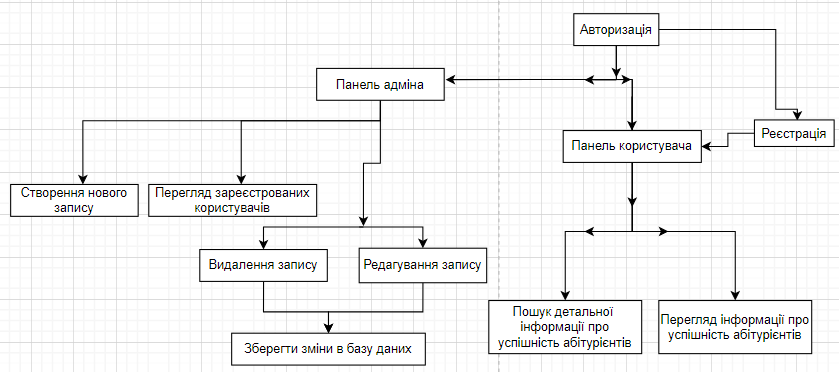


Рис. 2.2. Схема роботи програми

## 2.2 Розробка функціональних алгоритмів роботи програми

В першу чергу створимо діаграму класів для візуалізації взаємозв`язків класів у нашій програмі (рис. 2.3).

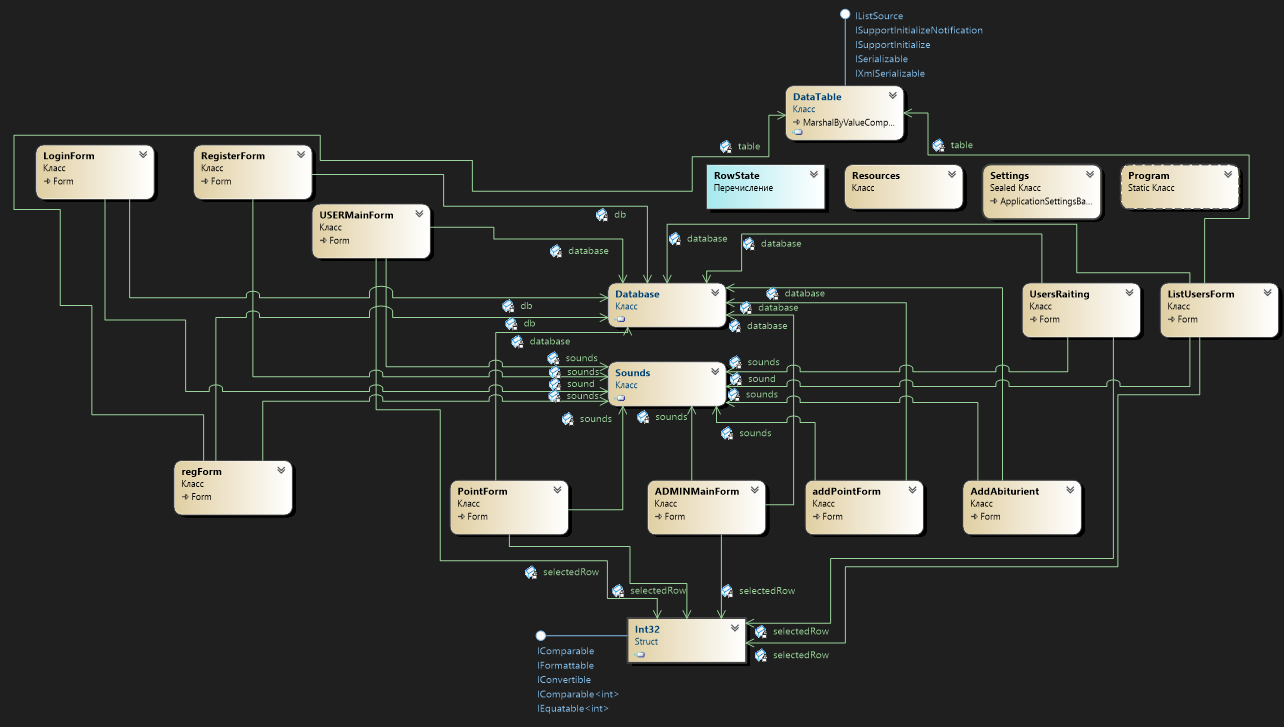


Рис. 2.3. Діаграма класів

На діаграмі класів ми бачимо, що всі форми успадковані від основої форми, ще бачимо що форми містять у собі поле класу Database.cs та Sounds.sc. А також перехування RowState, статичний клас Program і клас ресурсу зі строгою типізацією для пошуку локалізованих рядків Resources.

Далі складемо блок-схеми методів програми:

1. Метод зчитування з бази даних (рис. 2.4). В ньому буде реалізовано створення команди MySql, для передачі інструкції базі даних. А також реалізовано команду зчитування з бази даних усіх записів, після чого буде викликаний метод для заповнення стовпців DataGridView.

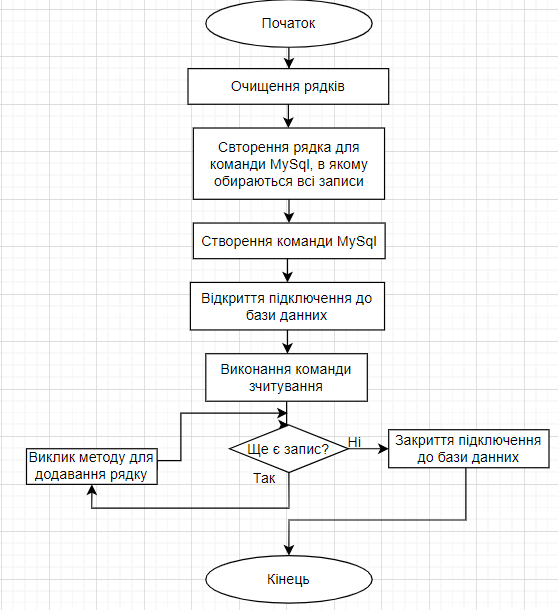


Рис. 2.4. Блок-схема для зчитування з бази даних

1. Метод пошуку (рис. 2.5). В ньому буде відбуватися вибірка записів з бази даних та подальший їх вивід.

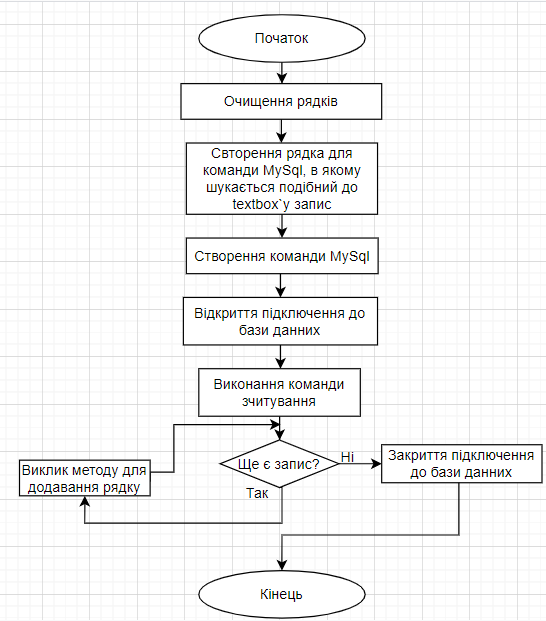


Рис. 2.5. Блок-схема методу пошуку

1. Метод оновлення DataGridView (рис. 2.6). В ньому буде відбуватися ново вивід записів з бази даних.

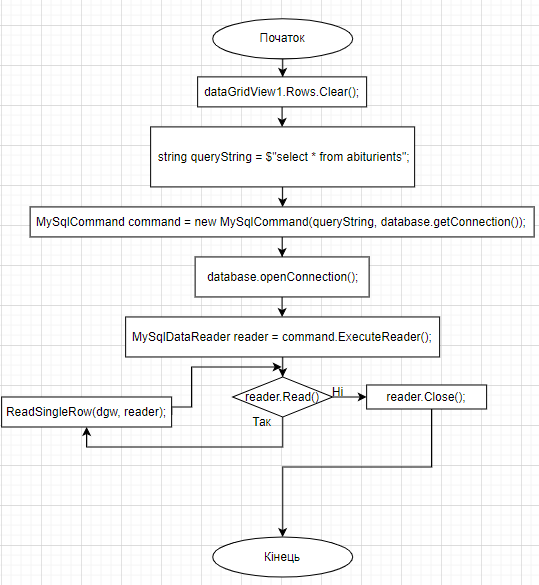


Рис. 2.6. Блок-схема методу оновлення DataGridView

1. Метод видалення (рис. 2.7). В ньому буде знайдено індекс елемента, який потрібно видалити.

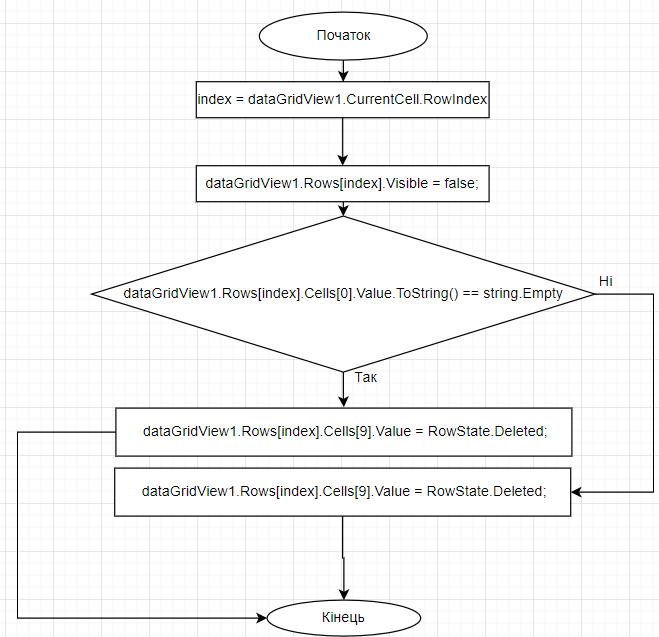


Рис. 2.7. Блок-схема методу видалення

1. Метод редагування (рис. 2.8). В ньому буде перезаписано усі поля запису.

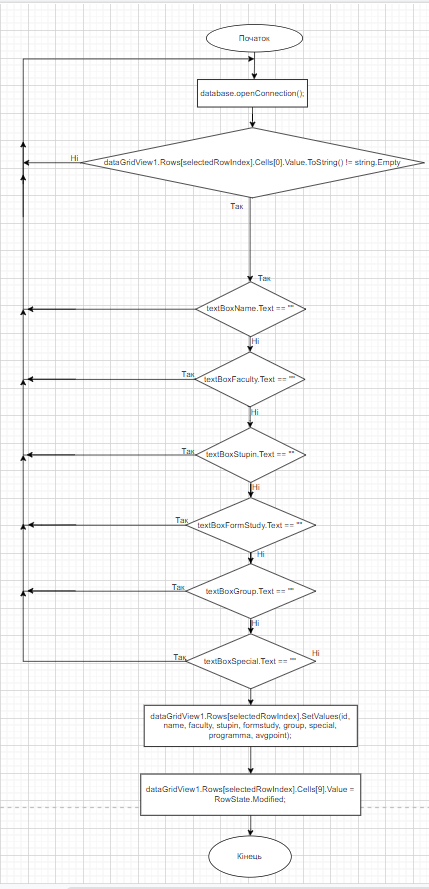


Рис. 2.8. Блок-схема методу редагування

1. Метод оновлення бази даних (рис. 2.9). В ньому буде передано базі даних зміни в записах.

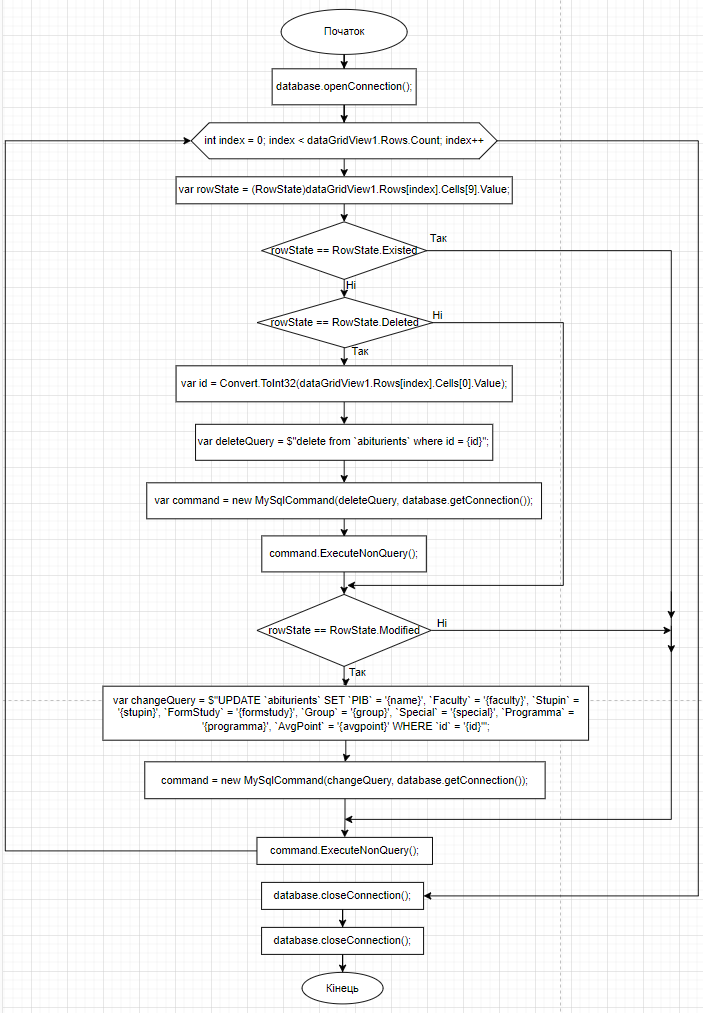


Рис. 2.9. Блок-схема оновлення бази даних

## 2.3 Розробка програмного забезпечення

Спочатку створимо клас, в якому будуть знаходитися методи для підключення до бази даних.

Лістинг:

public class Database

{

//Команда для підключення до бази данних MySql

MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=termpaper");

public void openConnection() //Функція яка відкриває з'єднання з базою данних

{

if (connection.State == System.Data.ConnectionState.Closed) //Перевірка: Якщо стан бази даних закритий, то

connection.Open(); // Відкриваємо з'єднання з базою даних

}

public void closeConnection() //Функція яка закриває з'єднання з базою данних

{

if (connection.State == System.Data.ConnectionState.Open) //Перевірка: Якщо стан бази даних відкритий, то

connection.Close(); // Закриваємо з'єднання з базою даних

}

//Функція, яка повертає з'єднання з базою даних

public MySqlConnection getConnection() //Ця функція повертає коткретний об'єкт від класу MySqlConnection(Саме цей об'єкт ми повертаємо)

{

return connection; //Повертається об'єкт з назвою connection

}

}

Далі реалізуємо вивід даних у DataGridView.

1. Створимо стовпці:

Лістинг:

public void CreateColumns()

{

dataGridView1.Columns.Add("id", "id");//0

dataGridView1.Columns.Add("PIB", "Прізвище, Ім'я, По-батькові");//1

dataGridView1.Columns.Add("Faculty", "Факультет");//2

dataGridView1.Columns.Add("Stupin", "Освітній ступінь");//3

dataGridView1.Columns.Add("FormStudy", "Форма навчання");//4

dataGridView1.Columns.Add("Group", "Група");//5

dataGridView1.Columns.Add("Special", "Спеціальність");//6

dataGridView1.Columns.Add("Programma", "Освітня програма");//7

dataGridView1.Columns.Add("AvgPoint", "Рейтинговий бал");//8

dataGridView1.Columns.Add("isNew", String.Empty);

}

1. Заповнимо їх:

Лістинг:

public void ReadSingleRow(DataGridView dgw, IDataRecord record)

{

dgw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetString(2), record.GetString(3), record.GetString(4), record.GetString(5), record.GetString(6), record.GetString(7), record.GetInt32(8), RowState.ModifiedNew);

}

Пошук в програмі буде відбуватися за допомогою оператора like в мові MySql.

Лістинг:

private void Search(DataGridView dgw)

{

dgw.Rows.Clear();

string searchString = $"select \* from `abiturients` where concat (`id`, `PIB`, `Faculty`, `Stupin`, `FormStudy`, `Group`, `Special`, `Programma`, `AvgPoint`) like '%" + textBoxSearch.Text + "%'";

MySqlCommand com = new MySqlCommand(searchString, database.getConnection());

database.openConnection();

MySqlDataReader read = com.ExecuteReader();

while(read.Read())

ReadSingleRow(dgw, read);

read.Close();

} //Пошук

Дані для редагування та видалення будуть відбуватися в основних формах ADMINForm.cs, PointForm.cs та ListUsersForm.cs. Детальніше методи для реалізації цього будуть розписані в додатку А.

Для обирання користувачем потрібного абітурієнта, інформацію про якого потрібно показати детальніше, було створенно подію CellClick.

Лістинг:

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

selectedRow = e.RowIndex;

if (e.RowIndex >= 0)

{

DataGridViewRow row = dataGridView1.Rows[selectedRow];

textBox\_id.Text = row.Cells[0].Value.ToString();

textBoxLogin.Text = row.Cells[1].Value.ToString();

textBoxPass.Text = row.Cells[2].Value.ToString();

reads();

}

}

## Висновки до другого розділу

Отже, створено загальний алгоритм роботи програми. Описано алгоритм роботи основних методів. Розроблено та наведено діаграму класів програми. Представлено та описано основні реалізовані методи.

# РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

## 3.1 Опис роботи з програмним додатком

Після запуску, користувача зустрічатиме вікно “Авторизації” на якій він може авторизуватися або перейти до вікна реєстрації для створення нового аккаунту. Користувач, можна бачити список записів та знайти якийсь з них, може зареєструватися або авторизуватися (рис. 3.1).

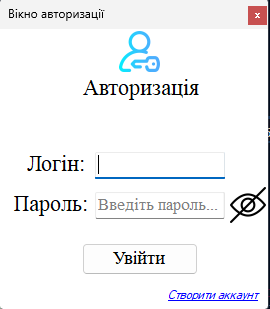


Рис. 3.1. Вікно авторизації програми

Якщо користувач вирішить зареєструватися, то він просто може натиснути на відповідний елемент - лінклейбл та перейти до вікна реєстрації, також при необхідності користувач може знову повернутись до вікна авторизації за допомогою того ж самого лінклейблу(рис. 3.2).

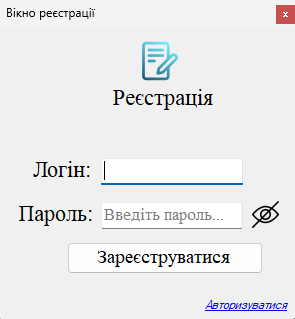


Рис. 3.2. Вікно реєстрації

Після успішної авторизації або реєстрації звичайний користувач побачити “Панель користувача”, яку ми можемо побачити на рисунку 3.3

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3.3. Панель користувача програми

Авторизований користувач має можливість до зручного перегляду інформації, після натиснення на відповідний рядок, а також доступ до пошуку та переходу до перегляду таблиці оцінок (функціонал залишається незмінним) з різних предметів,.

Почнемо з зручного перегляду, для нього користувач має натиснути лівою клавішою миші на потрібний рядок. Після цього, нижче, в відповідних текстбоксах з’явиться інформація про абітурієнта.(рис. 3.4).

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3.4. Перегляд записів у програмі

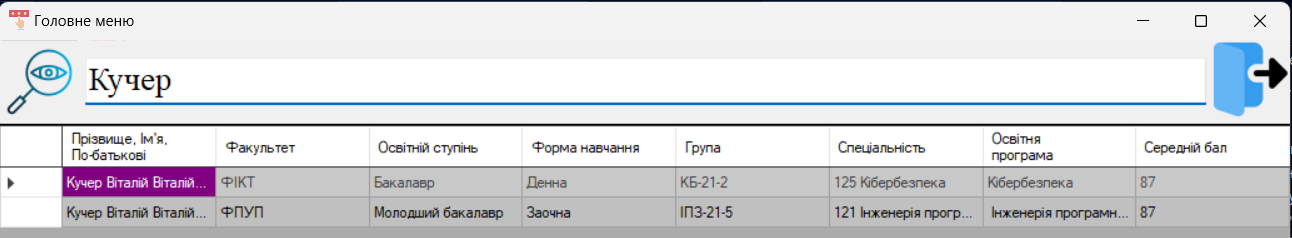
Далі перейдемо до пошуку, якщо користувач вирішить знайти потрібний йому запис, то він може просто вписати потрібну або приблизну інформацію у текстбокс, пошук відбувається по всім полям в цій таблиці. (рис 3.5) 

Рис. 3.5. Пошук запису в програмі

Якщо користувач захоче вийти з облікового запису, то після натиснення на відповідний пікчербокс в верхньому правому кутку головного вікна. Спочатку з'явиться меседжбокс з відповідним повідомленням(рис 3.6), після натиснення на кнопку «ДА» користувач потрапить на вікно авторизації (рис. 3.7).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.6. Меседжбокс при виході з облікового запису програми

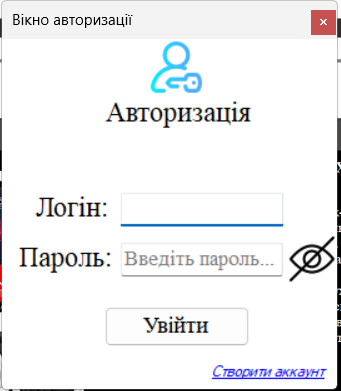


Рис. 3.7. Вікно після виходу з облікового запису програми

Користувач також має можливість перейти до таблиці з оцінками натиснувши на відповідну кнопку. Після натиснення на кнопку, користувача, як не дивно, зустрічає вікно з оцінками, де функціонал залишається незмінним(рис 3.8).

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3.8. Таблиця з оцінками програми

Якщо користувач є адміном, то він побачить вікно “Володарювання над формою” з іншим функціоналом (рис. 3.9).

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3.9. Панель адміна в програмі

Нижче ми бачимо додаткові пікчербокси “Видалити запис”, “Додати запис”, “Зберегти” та “Змінити”. Натиснувши на пікчербокс “Додати запис”, користувач побачить “Вікно додавання абіурієнта” (рис. 3.10).

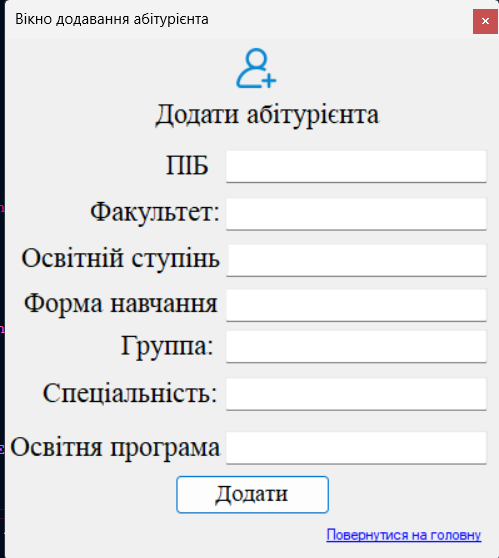


Рис. 3.10. Вікно додавання запису

Для редагування або видалення користувачу потрібно двічі клікнути на комірку таблиці з потрібним записом. Він побачить, що в панелі з назвою “Запис” заповняться текстбокси відповідно до вибраного рядка, де зможе відредагувати або видалити запис (рис. 3.11).



Рис. 3.11. Панелі редагування запису

В програмі передбачено виставлення оцінкок вже існуючим абітурієнтам (рис. 3.12).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.12. Список існуючих абітурієнтів

Якщо адмін при спробі виставити оцінку абітурієнту, оцінка якому вже виставлна, з’явиться текстбокс типу question. (рис. 3.13).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.13. Питання “Оцінки цьому абітурієнту вже вситавлені, впевнені що хочете продовжити?”

Після натиснення кнопки “Нет”, адміністратор повернеться до форми, де зможе вибрати іншого абітурієнта для виставлення оцінки. Якщо ж адмін все-таки впевнений в виставленні оцінки, з’явиться текстбокс з відповідним повідомленням . (рис. 3.14).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.14. Повідомлення “Оцінка виставлена! ”

Адміністртор також може перейти до списку створених аккаунтів, натиснувши на кнопку “Користувачі”, де матиме можливість змінювати логін та пароль, додавати та видаляти аккаунти самостійно.(рис 3.15)

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3.15 Меню “Список користувачів”

## 3.2 Тестування роботи програмного забезпечення

Програма передбачає помилки при введені користувачем даних для запобіження помилок. Наприклад не можна при реєстрації вказати логін, який уже є в базі даних (рис. 3.16).

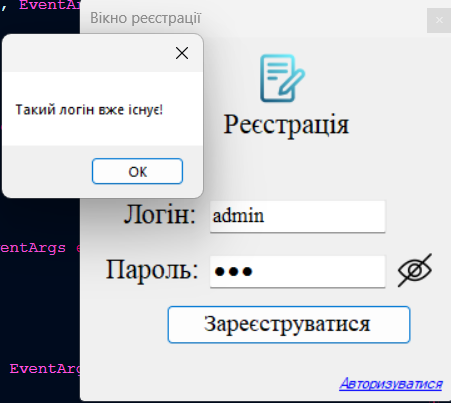


Рис. 3.16 Помилка “Такий логін вже існує”

Якщо користувач вирішить при редагуванні або додавані запису залишить який-небудь текстбокс порожнім, він також зустріне помилку. (рис 3.17, рис 3.18, рис 3.19)

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.17. Помилка “Поле “Факультет” повинно бути заповненим!”

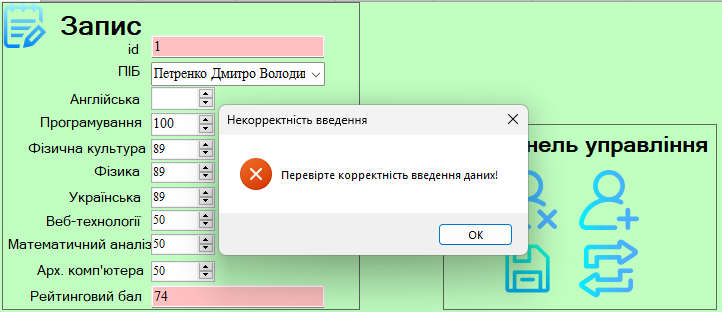


Рис. 3.18. Помилка “Перевірте коректність введення даних!”

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.19. Помилка “Поле ‘Пароль’ повинно бути заповненим!”

При введені неіснуючого логіну або невірного паролю під час авторизації, користувача про це сповістить програма (рис. 3.20).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.20. Помилка “Невірний логін або пароль!”

## Висновки до третього розділу

Отже, в межах даного розділу було повністю описано роботу з інтерфейсом та функціоналом програми від моменту запуску до завершення роботи. Також було проаналізовано та протестовано помилки, які можуть виникнути під час роботи програми.

# ВИСНОВКИ

Отже, під час виконання курсової роботи було вдосконалено навички програмування мовою С#. Проведено роботу з формами, класами та базою даних та поглиблено знання з ООП. Було створено програмне забезпечення для програми на тему “Інформаційно-пошукова система: сучасні моделі відеокарт”.

У програмі було реалізовано увесь потрібний та запланований функціонал:

− Програма містить реєстрацію та авторизацію та здійснює перехід між різними формами.

− Виведення даних на екран в зручному вигляді;

− Дані зберігаються в базі даних.

− Пошук даних (за всіма параметрами).

− Редагування даних.

− Видалення даних.

− Додавання нових записів.

Таким чином, програма відповідає усім поставленим вимогам та задачам. В подальшому може бути покращенна шляхом збільшення та доповнення функціоналу.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Эндрю Т. Язык Программирования C# 5.0 и платформа .NET 4.5 / Троелсен Эндрю., 2013. – 1168 с. – (6-е издание).

2. Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Джеффри Рихтер. – Питер, 2013. – 896 с. – (4-е изд).

3. Фленов М. Библия C# / Михаил Фленов. – Питер: БХВ-Петербург, 2011. – 541 с. – (2-е изд). .

4. Admin. Создание меню MenuStrip [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/5.2.php>.

5. Admin. Контекстное меню ContextMenuStrip [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/5.4.php>.

6. Admin. Все операции с БД в графическом приложении [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/adonet/3.5.php>.

7. Admin. Операции со строками [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/7.2.php>.

8. Аdmin. MS Access и C# [Електронний ресурс] / admin. – 2018. – Режим доступу до ресурсу: <https://vscode.ru/prog-lessons/ms-access-i-c-sharp-rabotaem-sbd.html>9. Шилдт Герберт C# 4.0: полное руководство. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.

10. Admin. Добавление форм. Взаимодействие между формами [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/2.3.php>

11. Warren G. Подключение к данным в базе данных Access (Windows Forms) [Електронний ресурс] / Genevieve Warren. – 2019. – Режим доступу до ре сурсу: <https://docs.microsoft.com/ru-ru/visualstudio/data-tools/connect-to-data-in-anaccess-database-windows-forms?view=vs-2019>.

12. Admin. Отношения между классами и объектами [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: https://metanit.com/sharp/patterns/1.2.php

# ДОДАТКИ

Додаток А

Бібліотека класів MyLib

Лістинг db.cs:

using MySql.Data.MySqlClient;

namespace MyLib

{

public class Database

{

//Команда для підключення до бази данних MySql

MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=termpaper");

public void openConnection() //Функція яка відкриває з'єднання з базою данних

{

if (connection.State == System.Data.ConnectionState.Closed) //Перевірка: Якщо стан бази даних закритий, то

connection.Open(); // Відкриваємо з'єднання з базою даних

}

public void closeConnection() //Функція яка закриває з'єднання з базою данних

{

if (connection.State == System.Data.ConnectionState.Open) //Перевірка: Якщо стан бази даних відкритий, то

connection.Close(); // Закриваємо з'єднання з базою даних

}

//Функція, яка повертає з'єднання з базою даних

public MySqlConnection getConnection() //Ця функція повертає коткретний об'єкт від класу MySqlConnection(Саме цей об'єкт ми повертаємо)

{

return connection; //Повертається об'єкт з назвою connection

}

}

}

Лістинг Sounds.cs:

using System.Media;

namespace MyLib

{

public class Sounds

{

public void Hovering()//Відтворення звукового еффекту при наведенні на кнопку

{

SoundPlayer hovering = new SoundPlayer(@"C:\Users\admin\source\repos\КУРСОВА\SoundEffects\HoveringOverTheButton.wav");

hovering.Play();

}

public void Press()//Відтворення звукового еффекту при натисканні на кнопку

{

SoundPlayer press = new SoundPlayer(@"C:\Users\admin\source\repos\КУРСОВА\SoundEffects\PressOnButton.wav");

press.Play();

}

public void Invalid()//Відтворення звукового еффекту при неправильному введенні

{

SoundPlayer invalid = new SoundPlayer(@"C:\Users\admin\source\repos\КУРСОВА\SoundEffects\InvalidEnter.wav");

invalid.Play();

}

public void Close()//Відтворення звукового еффекту при закритті форми

{

SoundPlayer close = new SoundPlayer(@"C:\Users\admin\source\repos\КУРСОВА\SoundEffects\CloseMenuButton.wav");

close.Play();

}

}

}

Проект WinForms

Лістинг AdminForm.cs:

using MySql.Data.MySqlClient;

using System;

using System.Data;

using System.Windows.Forms;

using MyLib;

namespace Успішність

{

public partial class ADMINMainForm : Form

{

Sounds sounds = new Sounds();

Database database = new Database();

int selectedRow;

public ADMINMainForm()

{

InitializeComponent();

}

public void reads(){

if (textBoxAvgPoints.Text == "")

{

pictureBoxChange.Visible = false;

textBoxName.ReadOnly = true;

textBoxFaculty.ReadOnly = true;

textBoxStupin.ReadOnly = true;

textBoxFormStudy.ReadOnly = true;

textBoxGroup.ReadOnly = true;

textBoxSpecial.ReadOnly = true;

textBoxProgramma.ReadOnly = true;

}

else

{

pictureBoxChange.Visible = true;

textBoxName.ReadOnly = false;

textBoxFaculty.ReadOnly = false;

textBoxStupin.ReadOnly = false;

textBoxFormStudy.ReadOnly = false;

textBoxGroup.ReadOnly = false;

textBoxSpecial.ReadOnly = false;

textBoxProgramma.ReadOnly = false;

}

}

public void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

selectedRow = e.RowIndex;

if (e.RowIndex >= 0)

{

DataGridViewRow row = dataGridView1.Rows[selectedRow];

textBox\_id.Text = row.Cells[0].Value.ToString();

textBoxName.Text = row.Cells[1].Value.ToString();

textBoxFaculty.Text = row.Cells[2].Value.ToString();

textBoxStupin.Text = row.Cells[3].Value.ToString();

textBoxFormStudy.Text = row.Cells[4].Value.ToString();

textBoxGroup.Text = row.Cells[5].Value.ToString();

textBoxSpecial.Text = row.Cells[6].Value.ToString();

textBoxProgramma.Text = row.Cells[7].Value.ToString();

textBoxAvgPoints.Text = row.Cells[8].Value.ToString();

}

reads();

}

public void CreateColumns()

{

dataGridView1.Columns.Add("id", "id");//0

dataGridView1.Columns.Add("PIB", "Прізвище, Ім'я, По-батькові");//1

dataGridView1.Columns.Add("Faculty", "Факультет");//2

dataGridView1.Columns.Add("Stupin", "Освітній ступінь");//3

dataGridView1.Columns.Add("FormStudy", "Форма навчання");//4

dataGridView1.Columns.Add("Group", "Група");//5

dataGridView1.Columns.Add("Special", "Спеціальність");//6

dataGridView1.Columns.Add("Programma", "Освітня програма");//7

dataGridView1.Columns.Add("AvgPoint", "Рейтинговий бал");//8

dataGridView1.Columns.Add("isNew", String.Empty);

}

public void ReadSingleRow(DataGridView dgw, IDataRecord record)

{

dgw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetString(2), record.GetString(3), record.GetString(4), record.GetString(5), record.GetString(6), record.GetString(7), record.GetInt32(8), RowState.ModifiedNew);

}

public void RefreshDataGrid(DataGridView dgw)

{

dgw.Rows.Clear();

string queryString = $"select \* from abiturients";

MySqlCommand command = new MySqlCommand(queryString, database.getConnection());

database.openConnection();

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

ReadSingleRow(dgw, reader);

reader.Close();

}

private void ClearFields()

{

textBox\_id.Text = "";

textBoxName.Text = "";

textBoxFaculty.Text = "";

textBoxStupin.Text = "";

textBoxFormStudy.Text = "";

textBoxGroup.Text = "";

textBoxSpecial.Text = "";

textBoxProgramma.Text = "";

textBoxAvgPoints.Text = "";

}

private void deleteRow()

{

int index = dataGridView1.CurrentCell.RowIndex;

dataGridView1.Rows[index].Visible = false;

if(dataGridView1.Rows[index].Cells[0].Value.ToString() == string.Empty)

{

dataGridView1.Rows[index].Cells[9].Value = RowState.Deleted;

return;

}

dataGridView1.Rows[index].Cells[9].Value = RowState.Deleted;

}

private void UpDate()

{

database.openConnection();

for(int index = 0; index < dataGridView1.Rows.Count; index++)

{

var rowState = (RowState)dataGridView1.Rows[index].Cells[9].Value;

if (rowState == RowState.Existed)

continue;

if (rowState == RowState.Deleted)

{

var id = Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value);

var deleteQuery = $"delete from `abiturients` where id = {id}";

var command = new MySqlCommand(deleteQuery, database.getConnection());

command.ExecuteNonQuery();

}

if (rowState == RowState.Modified)

{

var id = dataGridView1.Rows[index].Cells[0].Value.ToString();

var name = dataGridView1.Rows[index].Cells[1].Value.ToString();

var faculty = dataGridView1.Rows[index].Cells[2].Value.ToString();

var stupin = dataGridView1.Rows[index].Cells[3].Value.ToString();

var formstudy = dataGridView1.Rows[index].Cells[4].Value.ToString();

var group = dataGridView1.Rows[index].Cells[5].Value.ToString();

var special = dataGridView1.Rows[index].Cells[6].Value.ToString();

var programma = dataGridView1.Rows[index].Cells[7].Value.ToString();

var avgpoint = dataGridView1.Rows[index].Cells[8].Value.ToString();

var changeQuery = $"UPDATE `abiturients` SET `PIB` = '{name}', `Faculty` = '{faculty}', `Stupin` = '{stupin}', `FormStudy` = '{formstudy}', `Group` = '{group}', `Special` = '{special}', `Programma` = '{programma}', `AvgPoint` = '{avgpoint}' WHERE `id` = '{id}'";

var command = new MySqlCommand(changeQuery, database.getConnection());

command.ExecuteNonQuery();

}

}

database.closeConnection();

}

private void pictureBoxRemoveUser\_Click(object sender, EventArgs e)

{

reads();

sounds.Press();

DialogResult result = MessageBox.Show("Ви дійсно хочете видалити запис?", "Видалення запису", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

sounds.Close();

deleteRow();

ClearFields();

pictureBoxChange.Visible = false;

}

else

{

sounds.Press();

return;

}

}

private void pictureBoxExit\_Click(object sender, EventArgs e)

{

if (pictureBoxChange.Visible == false)

pictureBoxChange.Visible = true;

reads();

sounds.Press();

DialogResult result = MessageBox.Show("Ви дійсно хочете вийти з облікового запису?", "Вихід", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

sounds.Close();

this.Hide(); //Звертаємось до вікна RegisterForm та ховаємо його, тобто ховаєто вікно авторизації

LoginForm autorizationForm = new LoginForm();//Звертаємось до класу LoginForm та створюємо на його основі деякий об'єкт autorizationForm і вижіляємо під нього пам'ять

autorizationForm.Show();//Звертаємось до об'єкту autorizationForm і функції Show(), яка дозволить нам відкрити вікно реєстрації loginForm.cs

}

else

{

sounds.Press();

return;

}

}

private void ADMINMainForm\_Load(object sender, EventArgs e)

{

reads();

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

CreateColumns();

RefreshDataGrid(dataGridView1);

dataGridView1.Columns["isNew"].Visible = false;

}

private void pictureBoxAddUser\_Click(object sender, EventArgs e)

{

reads();

sounds.Press();

this.Hide();//Звертаємось до вікна ADMINMainForm та ховаємо його

AddAbiturient abitForm = new AddAbiturient();//Звертаємось до класу ADMINMainForm та створюємо на його основі деякий об'єкт mainForm і вижіляємо під нього пам'ять

abitForm.Show(); //Звертаємось до об'єкту mainForm і функції Show(), яка дозволить нам відкрити головне вікно AddAbiturientForm

}

private void Search(DataGridView dgw)

{

dgw.Rows.Clear();

string searchString = $"select \* from `abiturients` where concat (`id`, `PIB`, `Faculty`, `Stupin`, `FormStudy`, `Group`, `Special`, `Programma`, `AvgPoint`) like '%" + textBoxSearch.Text + "%'";

MySqlCommand com = new MySqlCommand(searchString, database.getConnection());

database.openConnection();

MySqlDataReader read = com.ExecuteReader();

while(read.Read())

ReadSingleRow(dgw, read);

read.Close();

} //Пошук

private void textBoxSearch\_TextChanged(object sender, EventArgs e)

{

reads();

Search(dataGridView1);

}

private void pictureBoxRefresh\_Click(object sender, EventArgs e)

{

if (pictureBoxChange.Visible == false)

pictureBoxChange.Visible = true;

reads();

sounds.Press();

RefreshDataGrid(dataGridView1);

}

private void pictureBoxSave\_Click(object sender, EventArgs e)

{

if (pictureBoxChange.Visible == false)

pictureBoxChange.Visible = true;

reads();

sounds.Press();

UpDate();

}

private void Change()

{

var selectedRowIndex = dataGridView1.CurrentCell.RowIndex;

var id = textBox\_id.Text;

var name = textBoxName.Text;

var faculty = textBoxFaculty.Text;

var stupin = textBoxStupin.Text;

var formstudy = textBoxFormStudy.Text;

var group = textBoxGroup.Text;

var special = textBoxSpecial.Text;

var programma = textBoxProgramma.Text;

int avgpoint = Convert.ToInt32(textBoxAvgPoints.Text);

if (dataGridView1.Rows[selectedRowIndex].Cells[0].Value.ToString() != string.Empty)

{

if (textBoxName.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'ПІБ' повинно бути заповним!");

sounds.Press();

}

else

if (textBoxFaculty.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'Факультет' повинно бути заповним!");

sounds.Press();

}

else

if (textBoxStupin.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'Освітній ступінь' повинно бути заповним!");

sounds.Press();

}

else

if (textBoxFormStudy.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'Форма навчання' повинно бути заповним!");

sounds.Press();

}

else

if (textBoxGroup.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'Группа' повинно бути заповним!");

sounds.Press();

}

else

if (textBoxSpecial.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'Спеціальність' повинно бути заповним!");

sounds.Press();

}

else

if (textBoxProgramma.Text == "")

{

sounds.Invalid();

MessageBox.Show("Поле 'Освітня программа' повинно бути заповним!");

sounds.Press();

}

else

{

dataGridView1.Rows[selectedRowIndex].SetValues(id, name, faculty, stupin, formstudy, group, special, programma, avgpoint);

dataGridView1.Rows[selectedRowIndex].Cells[9].Value = RowState.Modified;

}

}

}

private void pictureBoxChange\_Click(object sender, EventArgs e)

{

sounds.Press();

Change();

ClearFields();

reads();

}

private void buttonUsers\_Click(object sender, EventArgs e)

{

if (pictureBoxChange.Visible == false)

pictureBoxChange.Visible = true;

sounds.Press();

this.Hide();

ListUsersForm users = new ListUsersForm();

users.Show();

}

private void buttonPoints\_Click(object sender, EventArgs e)

{

if (pictureBoxChange.Visible == false)

pictureBoxChange.Visible = true;

sounds.Press();

this.Hide();

PointForm poi = new PointForm();

poi.Show();

}

private void MainForm\_FormClosed(object sender, FormClosedEventArgs e)

{

sounds.Close();

DialogResult res = MessageBox.Show("Ви впевненні?", "Вихід", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (res == DialogResult.No)

{

ADMINMainForm point = new ADMINMainForm();

point.Show();

}

else

Application.Exit();//Якщо вікно закривається, то відбувається завершення програми

}

private void buttonUsers\_MouseHover(object sender, EventArgs e)

{

sounds.Hovering();

}

private void buttonPoints\_MouseHover(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxRemoveUser\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxAddUser\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxSave\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxChange\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void buttonUsers\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void buttonPoints\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxRefresh\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxExit\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBox5\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

sounds.Invalid();

}

}

}

Лістинг Point.cs:

using System;

using System.Data;

using System.Windows.Forms;

using MyLib;

using MySql.Data.MySqlClient;

namespace Успішність

{

public partial class PointForm : Form

{

Database database = new Database();

Sounds sounds = new Sounds();

int selectedRow;

public PointForm()

{

InitializeComponent();

Fillcombo();

}

private void PointForm\_Load(object sender, EventArgs e)

{

reads();

CreateColumns();

RefreshDataGrid(dataGridView1);

Comparisonbutton.Visible = false;

Comparison.Visible = false;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

dataGridView1.Columns["check"].Visible = false;

dataGridView1.Columns["isNew"].Visible = false;

}

public void reads()

{

if (textBoxid.Text == "")

{

pictureBoxChange.Visible = false;

comboBoxPIB.Visible = false;

comboBox1.Visible = true;

EnglishUpDown.ReadOnly = true;

ProgrammingUpDown.ReadOnly = true;

CultureUpDown.ReadOnly = true;

PhysicUpDown.ReadOnly = true;

UkrainianUpDown.ReadOnly = true;

WEBUpDown.ReadOnly = true;

MathUpDown.ReadOnly = true;

ComputerUpDown.ReadOnly = true;

}

else

{

pictureBoxChange.Visible = true;

comboBoxPIB.Visible = true;

comboBox1.Visible = false;

EnglishUpDown.ReadOnly = false;

ProgrammingUpDown.ReadOnly = false;

CultureUpDown.ReadOnly = false;

PhysicUpDown.ReadOnly = false;

UkrainianUpDown.ReadOnly = false;

WEBUpDown.ReadOnly = false;

MathUpDown.ReadOnly = false;

ComputerUpDown.ReadOnly = false;

}

}

public void CheckToEmpty()

{

if (EnglishUpDown.Text == "")

EnglishUpDown.Text = "0";

if (ProgrammingUpDown.Text == "")

ProgrammingUpDown.Text = "0";

if (CultureUpDown.Text == "")

CultureUpDown.Text = "0";

if (PhysicUpDown.Text == "")

PhysicUpDown.Text = "0";

if (UkrainianUpDown.Text == "")

UkrainianUpDown.Text = "0";

if (WEBUpDown.Text == "")

WEBUpDown.Text = "0";

if (MathUpDown.Text == "")

MathUpDown.Text = "0";

if (ComputerUpDown.Text == "")

ComputerUpDown.Text = "0";

}

public void CreateColumns()

{

dataGridView1.Columns.Add("id", "№");//0

dataGridView1.Columns.Add("PIB", "Прізвище, Ім'я, По-батькові");//1

dataGridView1.Columns.Add("English", "Англійська");//2

dataGridView1.Columns.Add("Programing", "Програмування");//3

dataGridView1.Columns.Add("PhysicalCulture", "Фізкультура");//4

dataGridView1.Columns.Add("Physic", "Фізика");//5

dataGridView1.Columns.Add("Ukrainian", "Українська");//6

dataGridView1.Columns.Add("WEB", "Веб-технології");//7

dataGridView1.Columns.Add("Math", "Математика");//8

dataGridView1.Columns.Add("Computer", "Архітектура комп'ютера");//9

dataGridView1.Columns.Add("isNew", String.Empty);//10

DataGridViewCheckBoxColumn dgvcheckbox = new DataGridViewCheckBoxColumn();

dgvcheckbox.ValueType = typeof(bool);

dgvcheckbox.Name = "check";

dgvcheckbox.HeaderText = "Порівняння";

dataGridView1.Columns.Add(dgvcheckbox);

}

public void ReadSingleRow(DataGridView dgw, IDataRecord record)

{

dgw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetInt32(2), record.GetInt32(3), record.GetInt32(4), record.GetInt32(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetInt32(9), RowState.ModifiedNew);

}

public void RefreshDataGrid(DataGridView dgw)

{

dgw.Rows.Clear();

string queryString = $"select \* from `points`";

MySqlCommand command = new MySqlCommand(queryString, database.getConnection());

database.openConnection();

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

ReadSingleRow(dgw, reader);

reader.Close();

}

private void ClearFields()

{

textBoxid.Text = "";

comboBoxPIB.Text = "";

EnglishUpDown.Text = "";

ProgrammingUpDown.Text = "";

CultureUpDown.Text = "";

PhysicUpDown.Text = "";

UkrainianUpDown.Text = "";

WEBUpDown.Text = "";

MathUpDown.Text = "";

ComputerUpDown.Text = "";

}

private void buttonAbiturients\_Click(object sender, EventArgs e)

{

sounds.Press();

this.Hide();

ADMINMainForm adm = new ADMINMainForm();

adm.Show();

}

private void buttonUsers\_Click(object sender, EventArgs e)

{

sounds.Press();

this.Hide();

ListUsersForm list = new ListUsersForm();

list.Show();

}

private void pictureBoxRefresh\_Click(object sender, EventArgs e)

{

reads();

RefreshDataGrid(dataGridView1);

sounds.Press();

}

private void pictureBoxExit\_Click(object sender, EventArgs e)

{

reads();

sounds.Press();

DialogResult result = MessageBox.Show("Ви дійсно хочете вийти з облікового запису?", "Вихід", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

sounds.Close();

this.Hide(); //Звертаємось до вікна RegisterForm та ховаємо його, тобто ховаєто вікно авторизації

LoginForm autorizationForm = new LoginForm();//Звертаємось до класу LoginForm та створюємо на його основі деякий об'єкт autorizationForm і вижіляємо під нього пам'ять

autorizationForm.Show();//Звертаємось до об'єкту autorizationForm і функції Show(), яка дозволить нам відкрити вікно реєстрації loginForm.cs

}

else

{

sounds.Press();

return;

}

}

private void UpDate()

{

database.openConnection();

for (int index = 0; index < dataGridView1.Rows.Count; index++)

{

var rowState = (RowState)dataGridView1.Rows[index].Cells[10].Value;

if (rowState == RowState.Existed)

continue;

if (rowState == RowState.Deleted)

{

var id = Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value);

var deleteQuery = $"delete from `points` where `id` = {id}";

var command = new MySqlCommand(deleteQuery, database.getConnection());

command.ExecuteNonQuery();

}

if (rowState == RowState.Modified)

{

var id = dataGridView1.Rows[index].Cells[0].Value.ToString(); ;

var name = dataGridView1.Rows[index].Cells[1].Value.ToString();

var engl = dataGridView1.Rows[index].Cells[2].Value.ToString() ;

var prgrm = dataGridView1.Rows[index].Cells[3].Value.ToString();

var foot = dataGridView1.Rows[index].Cells[4].Value.ToString();

var phsc = dataGridView1.Rows[index].Cells[5].Value.ToString();

var ukr = dataGridView1.Rows[index].Cells[6].Value.ToString();

var VT = dataGridView1.Rows[index].Cells[7].Value.ToString();

var iLoveMeth = dataGridView1.Rows[index].Cells[8].Value.ToString();

var AK = dataGridView1.Rows[index].Cells[9].Value.ToString();

var avg = textBoxAvg.Text;

var changeQuery = $"UPDATE `points` SET `PIB` = '{name}', `English` = '{engl}', `Programing` = '{prgrm}', `PhysicalCulture` = '{foot}', `Physic` = '{phsc}', `Ukrainian` = '{ukr}', `WEB` = '{VT}', `Math` = '{iLoveMeth}', `Computer` = '{AK}' WHERE `id` = '{id}'";

var changeQuery1 = $"UPDATE `abiturients` SET `AvgPoint` = '{avg}' WHERE `PIB` = '{name}'";

var command = new MySqlCommand(changeQuery, database.getConnection());

var command1 = new MySqlCommand(changeQuery1, database.getConnection());

command.ExecuteNonQuery();

command1.ExecuteNonQuery();

}

}

database.closeConnection();

}

private void pictureBoxRemoveUser\_Click(object sender, EventArgs e)

{

reads();

sounds.Press();

deleteRow();

ClearFields();

}

private void deleteRow()

{

int index = dataGridView1.CurrentCell.RowIndex;

dataGridView1.Rows[index].Visible = false;

if (dataGridView1.Rows[index].Cells[0].Value.ToString() == string.Empty)

{

dataGridView1.Rows[index].Cells[10].Value = RowState.Deleted;

return;

}

dataGridView1.Rows[index].Cells[10].Value = RowState.Deleted;

}

private void pictureBoxAddUser\_Click(object sender, EventArgs e)

{

reads();

sounds.Press();

this.Hide();

addPointForm add = new addPointForm();

add.Show();

}

private void Comparisonbutton\_Click(object sender, EventArgs e)

{

int[] array = new int[100];

int j = 0;

for (int i = 0; i < dataGridView1.RowCount; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[10].Value) == true)

{

array[j] = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);

j++;

}

}

if (j > 0)

{

this.Hide();

ComparForm f = new ComparForm();

f.arrayadd(array, j);

f.Show();

}

}

private void pictureBoxSave\_Click(object sender, EventArgs e)

{

reads();

sounds.Press();

UpDate();

}

private void Change()

{

if (EnglishUpDown.Text == "" || CultureUpDown.Text == "" || PhysicUpDown.Text == "" || MathUpDown.Text == "" || ProgrammingUpDown.Text == "" || UkrainianUpDown.Text == "" || WEBUpDown.Text == "" || ComputerUpDown.Text == "")

{

MessageBox.Show("Перевірте корректність введення даних!");

return;

}

if (isUserExists())

return;

var selectedRowIndex = dataGridView1.CurrentCell.RowIndex;

var id = textBoxid.Text;

var name = comboBoxPIB.Text;

var engl = EnglishUpDown.Text; //Залік

var foot = CultureUpDown.Text; //Залік

var phsc = PhysicUpDown.Text; //Залік

var iLoveMeth = MathUpDown.Text; //Залік

var ukr = UkrainianUpDown.Text; //Екзамен

var VT = ComputerUpDown.Text; //Екзамен

var prgrm = ProgrammingUpDown.Text; //Екзамен

var AK = ComputerUpDown.Text;//Екзамен

if (dataGridView1.Rows[selectedRowIndex].Cells[0].Value.ToString() != string.Empty)

{

dataGridView1.Rows[selectedRowIndex].SetValues(id, name, engl, prgrm, foot, phsc, ukr, VT, iLoveMeth, AK);

dataGridView1.Rows[selectedRowIndex].Cells[10].Value = RowState.Modified;

}

}

private void pictureBoxChange\_Click(object sender, EventArgs e)

{

sounds.Press();

Change();

ClearFields();

reads();

}

public void raiting()

{

CheckToEmpty();

int eng = Convert.ToInt32(EnglishUpDown.Text);//Залік

//

int cltr = Convert.ToInt32(CultureUpDown.Text);//Залік

int phsc = Convert.ToInt32(PhysicUpDown.Text);//Залік

int mat = Convert.ToInt32(MathUpDown.Text);//Залік

int prog = Convert.ToInt32(ProgrammingUpDown.Text);//Екзамен

int uk = Convert.ToInt32(UkrainianUpDown.Text); //Екзамен

int wt = Convert.ToInt32(WEBUpDown.Text); //Екзамен

int ak = Convert.ToInt32(ComputerUpDown.Text);//Екзамен

int exam = 0, zalik = 0;

if (EnglishUpDown.Text != "")

zalik++;

if (CultureUpDown.Text != "")

zalik++;

if ( PhysicUpDown.Text != "")

zalik++;

if (MathUpDown.Text != "")

zalik++;

if (ProgrammingUpDown.Text != "")

exam++;

if (ComputerUpDown.Text != "")

exam++;

if (WEBUpDown.Text != "")

exam++;

if (UkrainianUpDown.Text != "")

exam++;

if (exam == 0 && zalik == 0)

{

textBoxAvg.Text = "0";

return;

}

double summ = (2 \* (prog + wt + uk + ak) + (eng + cltr + phsc + mat)) / (2 \* exam + zalik);

if (summ > 100)

summ = 100;

//float summ = (eng + prog + cltr + phsc + uk + wt + mat + ak) / check;

textBoxAvg.Text = Convert.ToString(summ);

} //для підрахунку середнього значення

public void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

reads();

selectedRow = e.RowIndex;

if (e.RowIndex >= 0)

{

DataGridViewRow row = dataGridView1.Rows[selectedRow];

textBoxid.Text = row.Cells[0].Value.ToString();

comboBoxPIB.Text = row.Cells[1].Value.ToString();

EnglishUpDown.Text = row.Cells[2].Value.ToString();

ProgrammingUpDown.Text = row.Cells[3].Value.ToString();

CultureUpDown.Text = row.Cells[4].Value.ToString();

PhysicUpDown.Text = row.Cells[5].Value.ToString();

UkrainianUpDown.Text = row.Cells[6].Value.ToString();

WEBUpDown.Text = row.Cells[7].Value.ToString();

MathUpDown.Text = row.Cells[8].Value.ToString();

ComputerUpDown.Text = row.Cells[9].Value.ToString();

reads();

raiting();

}

}

private void Search(DataGridView dgw)

{

dgw.Rows.Clear();

string searchString = $"select \* from `points` where concat (`id`, `PIB`, `English`, `Programing`, `PhysicalCulture`, `Physic`, `Ukrainian`, `WEB`, `Math`, `Computer`) like '%" + textBoxSearch.Text + "%'";

MySqlCommand com = new MySqlCommand(searchString, database.getConnection());

database.openConnection();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

ReadSingleRow(dgw, read);

read.Close();

} //Пошук

private void Comparison\_Click(object sender, EventArgs e)

{

if (dataGridView1.Columns["check"].Visible == false)

{

Comparisonbutton.Visible = true;

textBoxSearch.ReadOnly = true;

dataGridView1.Columns["check"].Visible = true;

}

else

{

Comparisonbutton.Visible = false;

textBoxSearch.ReadOnly = false;

dataGridView1.Columns["check"].Visible = false;

}

}

public Boolean isUserExists() //Boolean(Булев, Логічний тип даних) - примітивний тип даних, які можуть приймати два можливі значення: true та false.

{

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter(); //Сторюємо об'єкт, з яким будемо працювати в майбутньому

MySqlCommand command = new MySqlCommand("SELECT \* FROM `points` WHERE `PIB` = @uL", database.getConnection());//Вказали деяку команду, яка повинна виконатись по відношенню до бази даних. Вказали до якої бази даних м підключаємось - db.getConnection()

//@uL та @uP це деякі 'Заглушки', які збільшують рівень захисту бази данних

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = comboBoxPIB.Text; //Замість 'заглушки @uL' вказали конктрентну змінну loginField

adapter.SelectCommand = command; //Обрали потрібну комманду та виконали її

adapter.Fill(table); //Всі ті данні які ми отримали, ми помістили всередину об'єкта table, який просто являється табличкою, в якій ми можемо зручно працювати з кожним іх об'єктів

if (table.Rows.Count > 0) //Якщо кількість рядків у об'єкта table більше 0, то

{

sounds.Invalid();

DialogResult result = MessageBox.Show("Оцінки цьому абітурєнту вже виставлені, впевнені що хочете продовжити?", "Питання", MessageBoxButtons.YesNo, MessageBoxIcon.Question); // Якщо умова істинна, то можемо стверджувати що користувач інсує

if (result == DialogResult.No)

return true;

else

return false;

}

else

return false;

}

private void textBoxSearch\_TextChanged(object sender, EventArgs e)

{

reads();

Search(dataGridView1);

}

void Fillcombo()

{

string comstring = "datasource=localhost;port=3306;username=root;password=root";

string Query = "select \* from termpaper.abiturients ;";

MySqlConnection conDataBase = new MySqlConnection(comstring);

MySqlCommand cmd = new MySqlCommand(Query, conDataBase);

MySqlDataReader myReader;

try

{

conDataBase.Open();

myReader = cmd.ExecuteReader();

while (myReader.Read())

{

string sname = myReader.GetString("PIB");

comboBoxPIB.Items.Add(sname);

}

}

catch

{

}

}

private void comboBoxPIB\_KeyPress(object sender, KeyPressEventArgs e)

{

this.comboBoxPIB.DropDownStyle = ComboBoxStyle.DropDownList;

}

//Виклик messagebox при закривання форми

private void PointForm\_FormClosed(object sender, FormClosedEventArgs e)

{

sounds.Close();

DialogResult result = MessageBox.Show("Ви впевненні?", "Вихід", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.No)

{

PointForm point = new PointForm();

point.Show();

}

else

Application.Exit();//Якщо вікно закривається, то відбувається завершення програми

}

//При виході із певних текстбоксів викликається фу-я raiting()

private void textBoxEnglish\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxPrograming\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxPhysicalCulture\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxPhysic\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxUkrainian\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxWEB\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxMath\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxComputer\_Leave(object sender, EventArgs e)

{

raiting();

}

private void comboBoxPIB\_Leave(object sender, EventArgs e)

{

raiting();

}

private void textBoxid\_Leave(object sender, EventArgs e)

{

raiting();

}

//Відтворення звукових еффектів нижче

private void pictureBoxRemoveUser\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxAddUser\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxSave\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxChange\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void buttonUsers\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void buttonAbiturients\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxRefresh\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBoxExit\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

sounds.Invalid();

}

private void pictureBox5\_MouseHover(object sender, EventArgs e)

{

sounds.Hovering();

}

//Введення тільки чисел в textbox

private void EnglishUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void ProgrammingUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void CultureUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void PhysicUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void UkrainianUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void WEBUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void MathUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void ComputerUpDown\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void textBoxAvg\_KeyPress(object sender, KeyPressEventArgs e)

{

raiting();

}

private void textBoxAvg\_Enter(object sender, EventArgs e)

{

raiting();

}

private void textBoxAvg\_MouseEnter(object sender, EventArgs e)

{

raiting();

}

private void EnglishUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void ProgrammingUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void CultureUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void PhysicUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void UkrainianUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void WEBUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void MathUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void ComputerUpDown\_Leave(object sender, EventArgs e)

{

raiting();

}

private void comboBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

this.comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;

}

private void Comparisonbutton\_MouseEnter(object sender, EventArgs e)

{

sounds.Hovering();

}

private void EnglishUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void ProgrammingUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void CultureUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void PhysicUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void UkrainianUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void WEBUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void MathUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

private void ComputerUpDown\_ValueChanged(object sender, EventArgs e)

{

reads();

}

}

}