

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: сучасні моделі відеокарт»

студента І курсу групи КБ-21-2
спеціальності 125 «Кібербезпека»
Новіцького Віктора Сергійовича
(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

_____	<u>В.Л. Левківський</u>
(підпис)	(прізвище та ініціали)
_____	<u>Г.В. Марчук</u>
(підпис)	(прізвище та ініціали)
_____	<u>О.В. Чижмотря</u>
(підпис)	(прізвище та ініціали)

Житомир – 2022

[illegible]

--	--	--	--

7. Дата видачі завдання “ 15 ” квітня _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
1	Постановка задачі	17.04.2022	
2	Пошук, огляд та аналіз аналогічних розробок	20.04.2022	
3	Формулювання технічного завдання	02.05.2022	
4	Опрацювання літературних джерел	03.06.2022	
5	Проектування структури	15.06.2022	
6	Написання програмного коду	30.06.2022	
7	Відлагодження	05.07.2022	
8	Написання пояснювальної записки	10.07.2022	
9	Захист		

Студент

(підпис)

Новіцький В.С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Левківський В.Л.

(прізвище та ініціали)

Реферат

Завданням на курсову роботу було створення інформаційно-пошукової системи для пошуку сучасних моделей відеокарт.

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

Текстова частина викладена на 34 сторінках друкованого тексту.

Пояснювальна записка має 24 сторінок додатків. Список використаних джерел містить 12 найменувань і займає 2 сторінки. В роботі наведено 30 рисунків. Загальний обсяг роботи – 58 сторінок.

Ключові слова: C#, SQL, WINDOWS FORMS, ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА, ВІДЕОКАРТА, БАЗА ДАНИХ.

					«Житомирська політехніка». 22.125.25.000 - ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Новіцький В.С.			Інформаційно-пошукова система: сучасні моделі ві- деокарт	Літ.	Арк.	Аркушів	
Перевір.		Левківський В.Л.					4	58	
Керівник						ФІКТ Гр. КБ 21-2[1]			
Н. контр.									
Зав. каф.									

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	12
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
2.1 ПРОЕКТУВАННЯ ЗАГАЛЬНОГО АЛГОРИТМУ РОБОТИ ПРОГРАМИ.....	13
2.2 РОЗРОБКА ФУНКЦІОНАЛЬНИХ АЛГОРИТМІВ РОБОТИ ПРОГРАМИ	14
2.3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	19
Висновки до другого розділу	20
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ	23
3.1 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ.....	23
3.2 ТЕСТУВАННЯ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	29
Висновки до третього розділу	31
ВИСНОВКИ	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33
ДОДАТКИ.....	35

ВСТУП

Актуальність теми: пошук інформації є однією з основних складових людської діяльності, з ним ми зіштовхуємося щодня: вивчаючи театральну афішу, щоб вибрати цікавий спектакль, підбираючи в розкладі поїздів зручну електричку, перегортаючи телефонну книгу... Людині, яка в силу своєї професії чи захоплення часто зіштовхується з підбором і пошуком якої-небудь тематичної інформації, рано чи пізно (зі зростанням її обсягу) приходить 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 C# та створення програмного забезпечення на тему "Інформаційно-пошукова система: сучасні моделі відеокарт". Встановлена мета: повинна забезпечити, реалізувати наступні функції:

- 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.
- Фільтрація даних (за 3-а параметрами);
- Додавання нових записів.
- Порівняння даних. Перед порівнянням відмітити потрібні записи
- Програма має містити реєстрацію та авторизацію та здійснювати перехід між різними формами.

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

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

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

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ док.ум.	Підпис	Дата		7

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДИ, ТА ЗАСОБИ ВИРІШЕННЯ

1.1. Аналіз задачі, засобів та методів її вирішення

Поставлено завдання створити інформаційно-пошукову систему на тему “Сучасні модель відеокарт”. Для зберігання інформації будемо використовувати баз даних, а саме MySQL для створення бази відеокарт та користувачів. Для виконання цього завдання програма має містити та вміти виконувати такі функції:

- Зчитувати інформацію 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 Windows Forms DataGridView.

- Пошук даних;

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

- Фільтрація даних;

Для реалізації буде створено метод, який фільтруватиме таблицю за параметрами, які визначить користувач.

- Порівняння даних;

Для реалізації буде створено метод, який відбиратиме обрані користувачем записи і показуватиме тільки їх.

- 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 запису;

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

– Авторизування і реєстрація користувача;

Це потрібно для ідентифікації користувача і надання йому можливостей: звичайний гість переглядати записи і доступ до пошуку адмін матиме змогу редагувати, видаляти та додавати записи, а зареєстрований юзер фільтрувати, бачити більш детальну інформацію та порівнювати записи.

– 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 інтернет магазини. Такі інформаційно-пошукові системи зазвичай використовуються на сайтах і переглядаються користувачем з браузера.

1) Один із найвідоміших прикладів є система пошуку маг інтернет-маркету з продажу відеокарт Rozetka, автором якої є Владислав Чечьоткін (<https://rozetka.com.ua/ua/>).

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

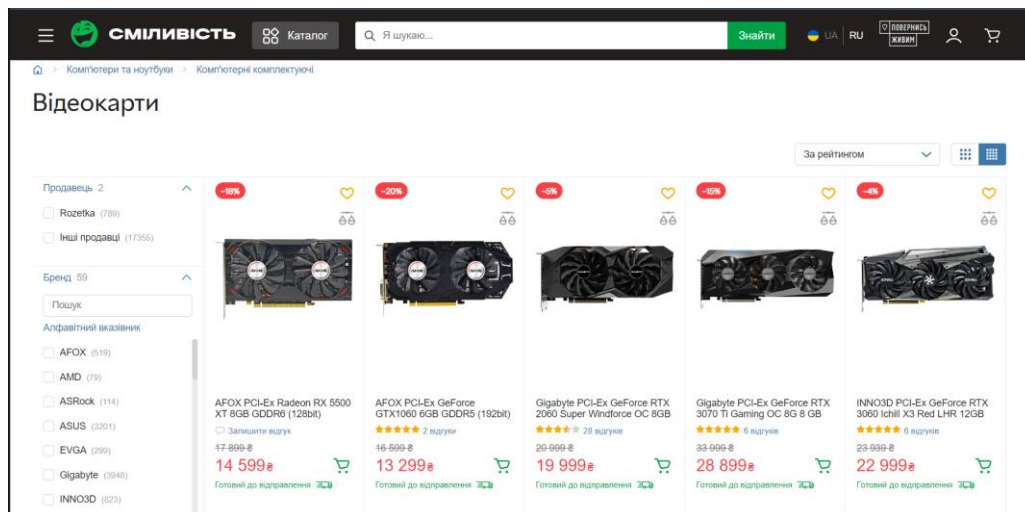


Рис. 1.1. Інтерфейс системи пошуку – “Rozetka”

Вона написана за допомогою Owox, яку створила українська компанія. Ця система має доволі широкий спектр можливостей: фільтрування, пошук, показ більш детальної інформації про конкретний продукт, сортування і занесення продукту до кошика.

Перевагою цієї системи є кількість функціоналу, а мінусом робота через браузер.

2) Іншим аналогом можна вважати інтернет систему пошуку для купівлі відеокарт e.katalog, автором якого є Руслан Новіков (<https://ek.ua/ua/>).

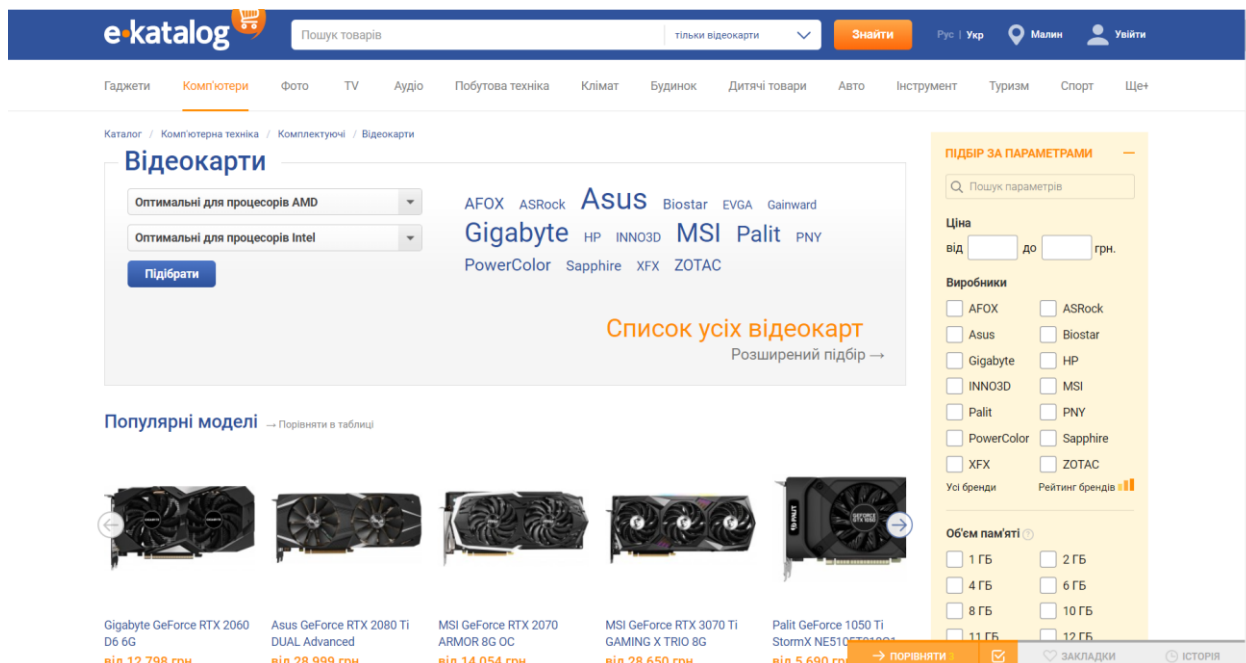


Рис. 1.2. Інтерфейс інформаційно-пошукової системи – “e.katalog”

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Ця система пошуку містить фільтрацію, сортування, порівняння, пошук, реєстрацію та авторизацію, порівняння цін в інтернет магазинах, перегляд детальної інформації про конкретний товар.

Основною перевагою цієї програми є зв'язок з іншими інформаційно-пошуковими системами, що дозволяє в одному середовищі переглядати дані відразу з деяких. Недоліком цієї системи пошуку є робота в браузері та залежність від інтернету.

3) Третім аналогом візьмемо закордону систему пошуку GPUTracker, автором якої є Чарльз Кьюффер (<https://www.gputracker.eu/en/terms>).

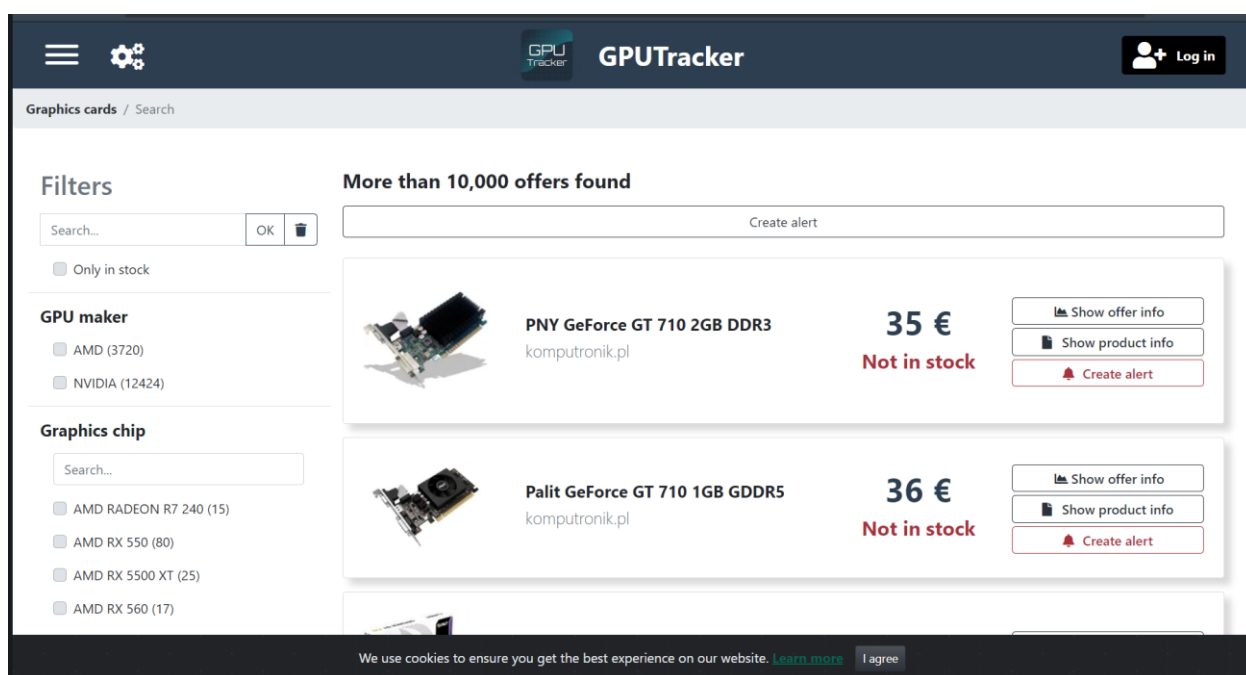


Рис. 1.3. Інтерфейс системи пошуку – “GPUTracker”

Система створена для знаходження найменшої ціни на відеокарту, тому містить відповідні функції й алгоритми: пошук, фільтрація, створення сповіщення про зниження ціни за відеокарту, реєстрація й авторизація та перегляд ціни в інтернет магазинах.

Перевагою є зв'язок з інтернет магазинами, а недоліками є відсутність української локалізації, відсутність сортування, мало інформації про відеокарту, присутня тільки про ціну та назва, і робота в браузері.

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

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проектування загального алгоритму роботи програми

Відкривавши програму, користувача зустріне панель гостя з якої він зможе виконати потрібні йому дії. (Рис. 2.1)

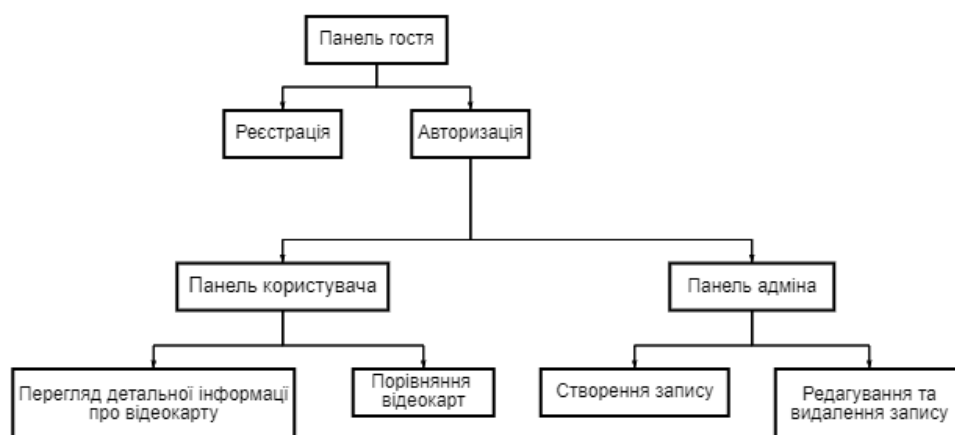


Рис. 2.1. Загальна схема варіантів використання програми

Для кращого розуміння переходу між функціоналом складемо детальну схему варіантів використання програми. (Рис. 2.2)

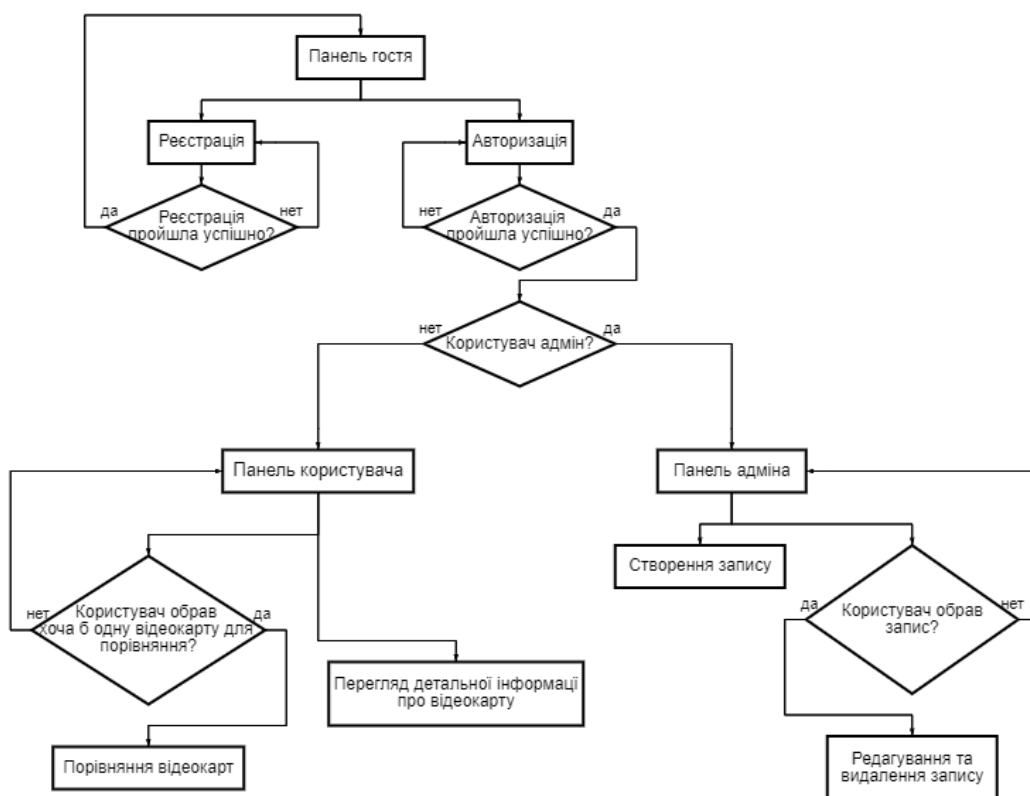


Рис. 2.2. Детальна схема варіантів використання програми

2.2 Розробка функціональних алгоритмів роботи програми

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

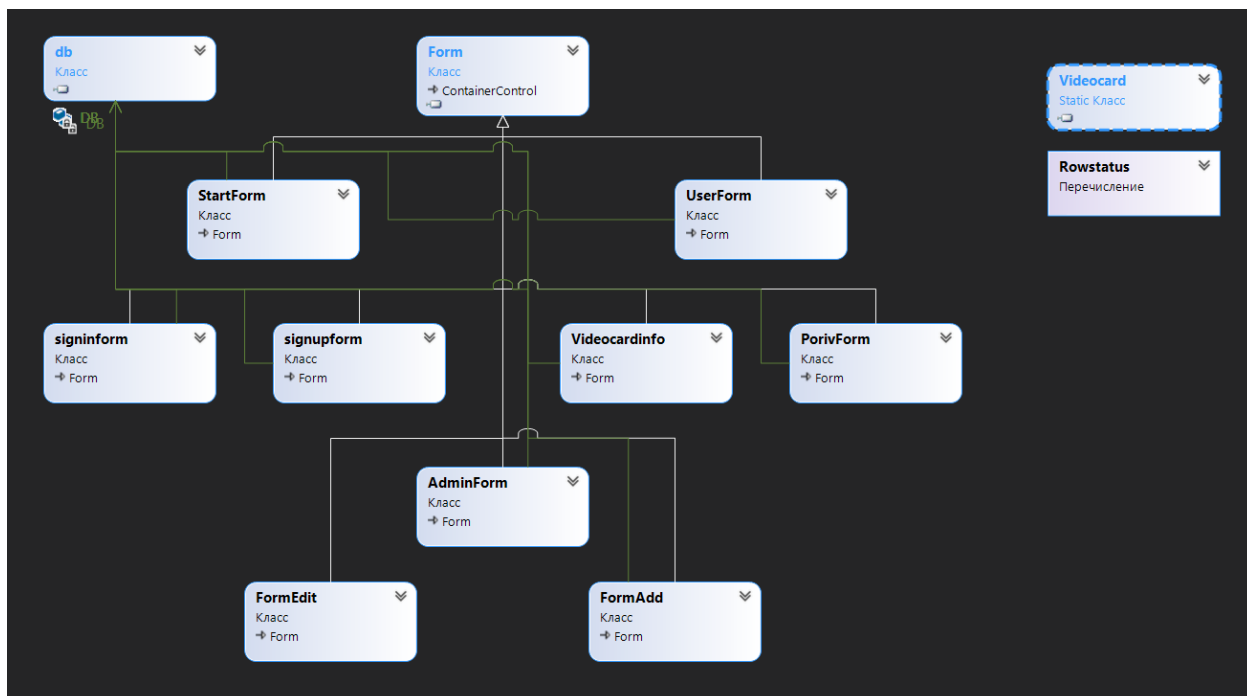


Рис. 2.3. Діаграма класів

Далі складемо блок-схеми методів програми:

1) Метод зчитування з бази даних (рис. 2.4):

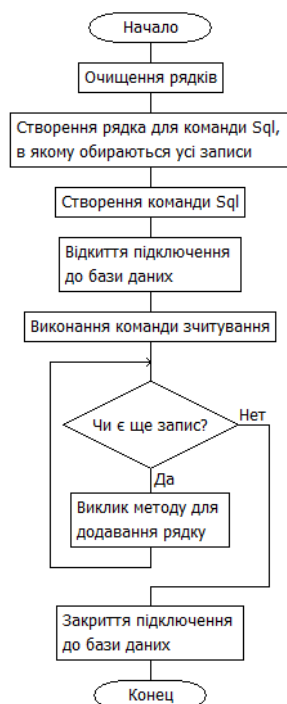


Рис. 2.4. Блок-схема для зчитування з бази даних

		Новіцький В.С.		
		Левківський В. Л.		
Змн.	Арк.	№ докум.	Підпис	Дата

2) Метод пошуку (рис. 2.5):

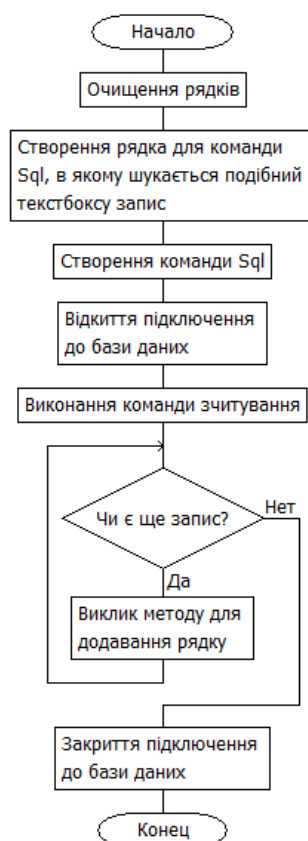


Рис. 2.5. Блок-схема методу пошуку

3) Метод оновлення DataGridView (рис. 2.7):

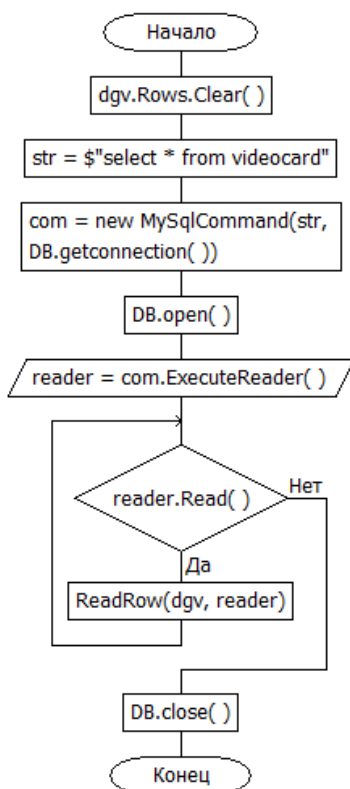


Рис. 2.7. Блок-схема методу оновлення DataGridView

4) Метод порівняння (рис. 2.6):

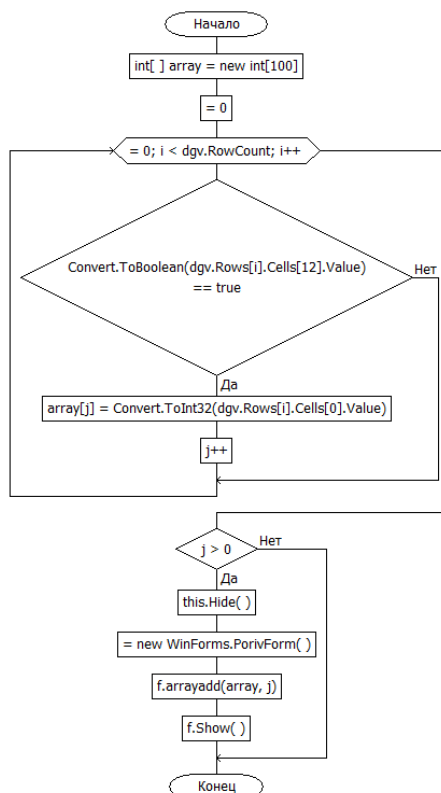


Рис. 2.6. Блок-схема методу порівняння

5) Метод видалення (рис. 2.8):

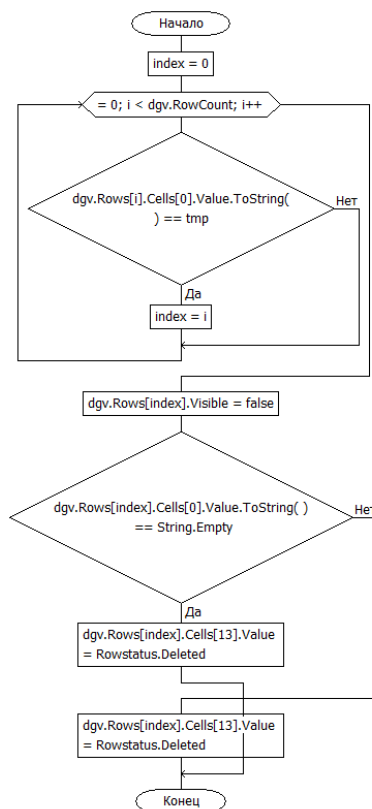


Рис. 2.8. Блок-схема методу видалення

6) Метод редагування (рис. 2.9):

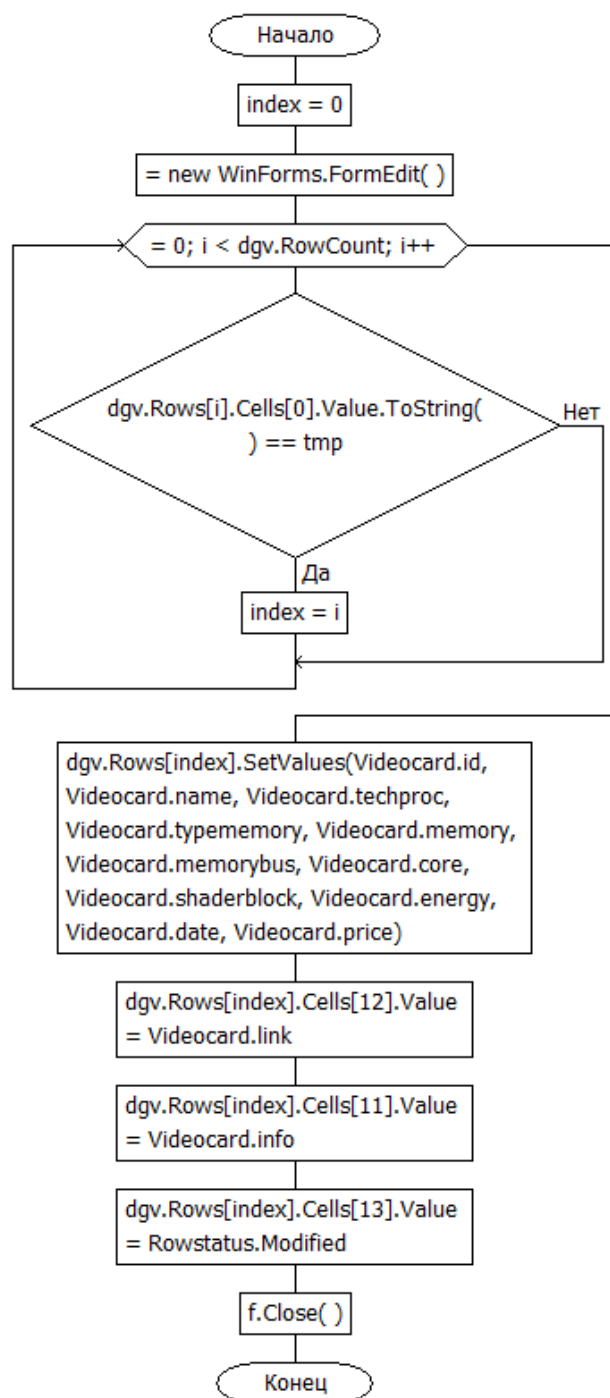


Рис. 2.9. Блок-схема методу редагування

7) Метод оновлення бази даних (рис. 2.10):

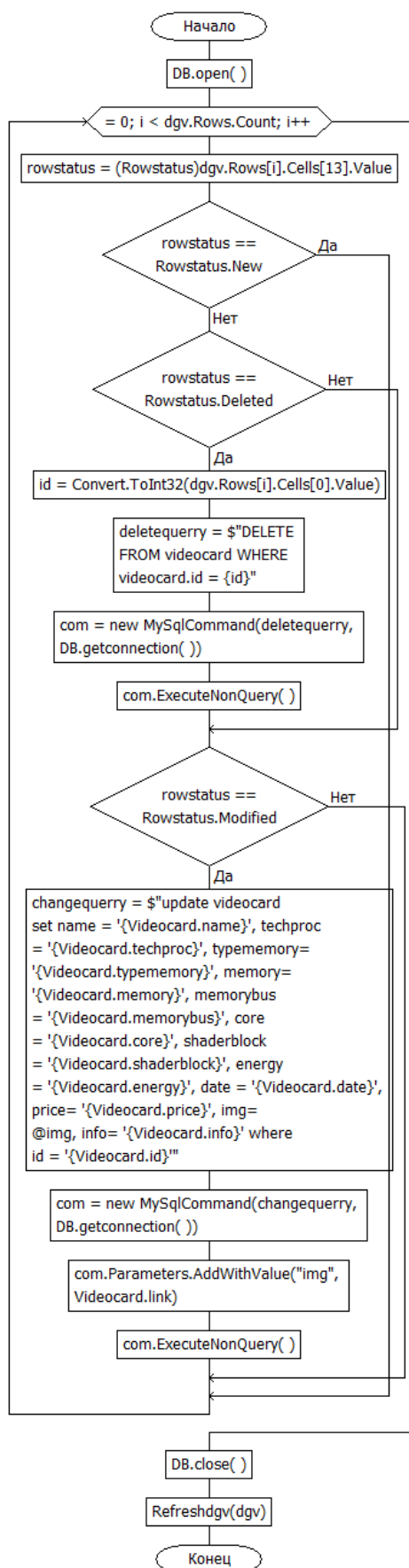


Рис. 2.10. Блок-схема оновлення бази даних

		Новіцький В.С.		
		Левківський В. Л.		
Змн.	Арк.	№ докум.	Підпис	Дата

2.3 Розробка програмного забезпечення

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

Лістинг:

```
public class db
{
    MySqlConnection connection = new MySqlConnection("server = localhost; port=3306;
    username=root; password = root;database =database");
    public void open()
    {
        if (connection.State == System.Data.ConnectionState.Closed)
            connection.Open();
    }
    public void close()
    {
        if (connection.State == System.Data.ConnectionState.Open)
            connection.Close();
    }
    public MySqlConnection getconnection()
    {
        return connection;
    }
}
```

Далі реалізуємо вивід даних у DataGridView.

1) Створимо стовпці:

Лістинг:

```
private void CreateColumns()
{
    dgv.Columns.Add("id", "№");
    dgv.Columns.Add("name", "Назва");
    dgv.Columns.Add("techproc", "Техпроцес, нм");
    dgv.Columns.Add("typememory", "Тип пам'яті");
    dgv.Columns.Add("memory", "Об'єм пам'яті");
    dgv.Columns.Add("memorybus", "Шина пам'яті");
    dgv.Columns.Add("core", "Частота роботи ядра, МГц");
    dgv.Columns.Add("shaderblock", "Шейдерних блоків");
    dgv.Columns.Add("energy", "Споживання енергії, Вт");
    dgv.Columns.Add("date", "Дата випуску");
    dgv.Columns.Add("price", "Вартість, $");
    dgv.Columns.Add("info", "info");
}
```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

```

dgv.Columns.Add("img", "img");
dgv.Columns.Add("status", String.Empty);
}

```

2) Заповнимо їх:

Лістинг:

```

private void ReadRow(DataGridView dgv, IDataRecord record)
{
    dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), record.GetString(11), record.GetString(12), Rowstatus.New); ;
}

private void Refreshdgv(DataGridView dgv)
{
    dgv.Rows.Clear();
    string str = $"select * from videocard";
    MySqlCommand com = new MySqlCommand(str, DB.getconnection());
    DB.open();
    MySqlDataReader reader = com.ExecuteReader();
    while (reader.Read())
        ReadRow(dgv, reader);
    DB.close();
}

```

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

Лістинг:

```

private void search(DataGridView dgv)
{
    dgv.Rows.Clear();
    string searchquerry = $"select * from videocard where concat (id, name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%'+ textBoxsearch.Text +'%";
    MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());
    DB.open();
    MySqlDataReader read = com.ExecuteReader();
    while (read.Read())
    {
        ReadRow(dgv, read);
    }
    read.Close();
}

```

Дані для редагування та видалення будуть передаватися в форму FormAdd.cs за допомогою статичного класа Videocard.cs. Детальніше методи для реалізації цього будуть розписані в додатку А.

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

При виведенні детальної інформації про відеокарту, користувач буде бачити фотографію, яка зберігається локально на жорсткому диску, а для редагування та додавання шляху написаний метод:

Лістинг:

```
private void buttonimg_Click(object sender, EventArgs e)
{
    ofd.Filter = "Image Files(*.JPG;*.PNG)|*.JPG;*.PNG|All files(*)|*.*";
    if(ofd.ShowDialog() == DialogResult.OK)
    {
        try
        {
            Videocard.link = ofd.FileName;
        }
        catch
        {
            MessageBox.Show("Неправильний формат картинки!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Для обирання користувачем відеокарти, інформацію про яку потрібно показати детальніше, створенно подію CellContentDoubleClick.

Лістинг:

```
private void dgv_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)
{
    selectedrow = e.RowIndex;
    int selectedcolumn = e.ColumnIndex;
    if (selectedcolumn == 1)
    {
        if (e.RowIndex >= 0)
        {
            if (dgv.Rows[selectedrow].Cells[1].Value.ToString() != String.Empty)
            {
                Videocard.name = dgv.Rows[selectedrow].Cells[1].Value.ToString();
                string querryimg = "select img from videocard where name like '%" + Videocard.name + "%'";
                string querryinfo = "select info from videocard where name like '%" + Videocard.name + "%'";
                DB.open();
                MySqlCommand com = new MySqlCommand(querryimg, DB.getconnection());
                Videocard.link = (string)com.ExecuteScalar();
                com = new MySqlCommand(querryinfo, DB.getconnection());
                Videocard.info = (string)com.ExecuteScalar();
                DB.close();
                this.Hide();
                WinForms.Videocardinfo f = new WinForms.Videocardinfo();
                f.Show();
            }
        }
    }
}
```

Висновки до другого розділу

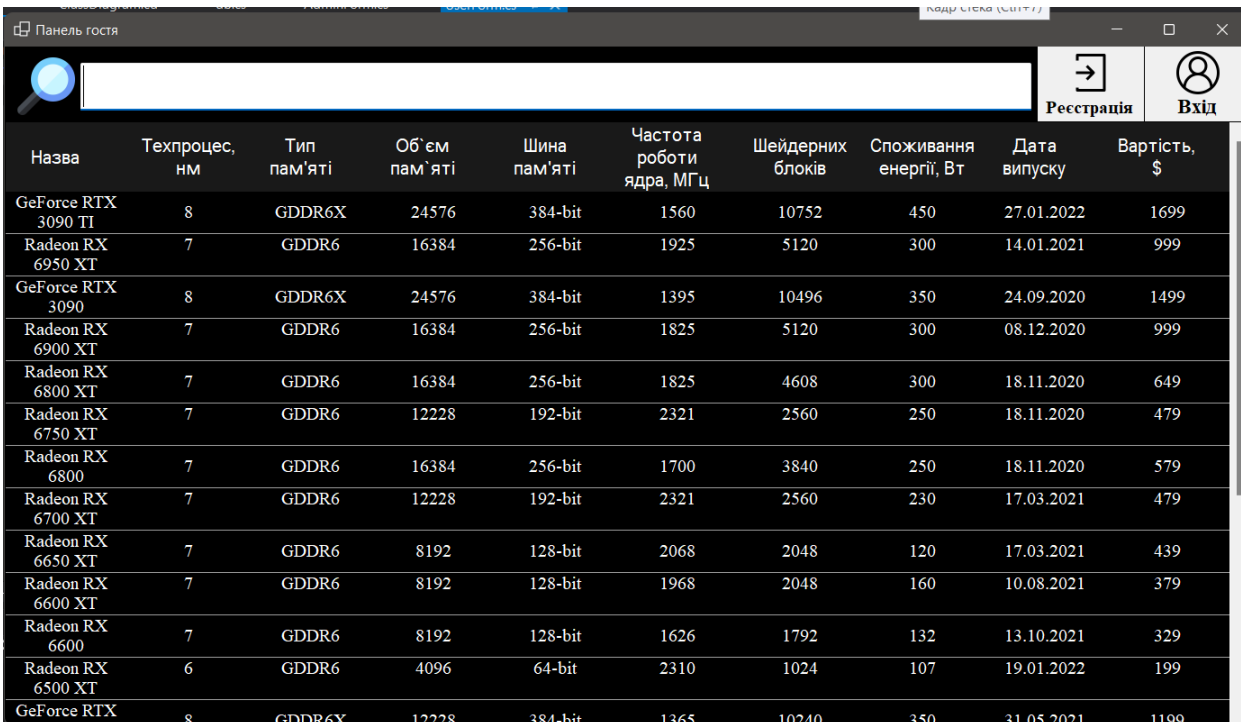
Отже, створено загальний алгоритм роботи програми. Описано алгоритм роботи основних методів. Розроблено та наведено діаграму класів програми. Представлено та описано основні реалізовані методи.

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Опис роботи з програмним додатком

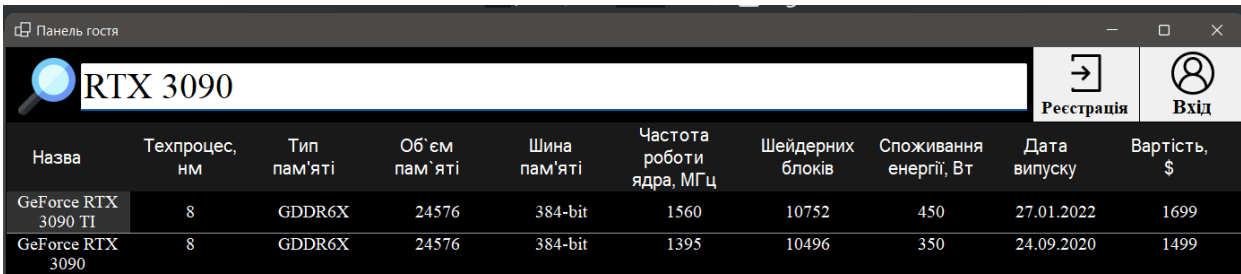
Після запуску користувача зустрічатиме вікно “Панель гостя” на якій він бачить список записів, може знайти якийсь з них, та може зареєструватися або авторизуватися. (Рис. 3.1)



Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$
GeForce RTX 3090 TI	8	GDDR6X	24576	384-bit	1560	10752	450	27.01.2022	1699
Radeon RX 6950 XT	7	GDDR6	16384	256-bit	1925	5120	300	14.01.2021	999
GeForce RTX 3090	8	GDDR6X	24576	384-bit	1395	10496	350	24.09.2020	1499
Radeon RX 6900 XT	7	GDDR6	16384	256-bit	1825	5120	300	08.12.2020	999
Radeon RX 6800 XT	7	GDDR6	16384	256-bit	1825	4608	300	18.11.2020	649
Radeon RX 6750 XT	7	GDDR6	12228	192-bit	2321	2560	250	18.11.2020	479
Radeon RX 6800	7	GDDR6	16384	256-bit	1700	3840	250	18.11.2020	579
Radeon RX 6700 XT	7	GDDR6	12228	192-bit	2321	2560	230	17.03.2021	479
Radeon RX 6650 XT	7	GDDR6	8192	128-bit	2068	2048	120	17.03.2021	439
Radeon RX 6600 XT	7	GDDR6	8192	128-bit	1968	2048	160	10.08.2021	379
Radeon RX 6600	7	GDDR6	8192	128-bit	1626	1792	132	13.10.2021	329
Radeon RX 6500 XT	6	GDDR6	4096	64-bit	2310	1024	107	19.01.2022	199
GeForce RTX	8	GDDR6X	12228	384-bit	1365	10240	350	31.05.2021	1199

Рис. 3.1. Стартове меню програми

Якщо користувач вирішить знайти запис, то він може просто вписати потрібну йому інформацію у текстбокс, пошук відбувається по всім полям таблиці. (Рис. 3.2)



Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$
GeForce RTX 3090 TI	8	GDDR6X	24576	384-bit	1560	10752	450	27.01.2022	1699
GeForce RTX 3090	8	GDDR6X	24576	384-bit	1395	10496	350	24.09.2020	1499

Рис. 3.2. Пошук запису в програмі

Для відкриття більшого функціоналу користувачу потрібно зареєструватися рисунок 3.3, або авторизуватися рисунок 3.4.

Реєстрація

Логін: Введіть логін

Пароль:

Зареєструватися

Рис. 3.3. Вікно реєстрації програми

Авторизація

Логін:

Пароль:

Авторизуватися

Рис. 3.4. Вікно авторизації програми

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Після успішної авторизації звичайний користувач побачить “Панель користувача”, яку ми можемо побачити на рисунку 3.5.

Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$
GeForce RTX 3090 TI	8	GDDR6X	24576	384-bit	1560	10752	450	27.01.2022	1699
Radeon RX 6950 XT	7	GDDR6	16384	256-bit	1925	5120	300	14.01.2021	999
GeForce RTX 3090	8	GDDR6X	24576	384-bit	1395	10496	350	24.09.2020	1499
Radeon RX 6900 XT	7	GDDR6	16384	256-bit	1825	5120	300	08.12.2020	999
Radeon RX 6800 XT	7	GDDR6	16384	256-bit	1825	4608	300	18.11.2020	649
Radeon RX 6750 XT	7	GDDR6	12228	192-bit	2321	2560	250	18.11.2020	479
Radeon RX 6800	7	GDDR6	16384	256-bit	1700	3840	250	18.11.2020	579
Radeon RX 6700 XT	7	GDDR6	12228	192-bit	2321	2560	230	17.03.2021	479
Radeon RX 6650 XT	7	GDDR6	8192	128-bit	2068	2048	120	17.03.2021	439
Radeon RX 6600 XT	7	GDDR6	8192	128-bit	1968	2048	160	10.08.2021	379
Radeon RX 6600	7	GDDR6	8192	128-bit	1626	1792	132	13.10.2021	329
Radeon RX 6500 XT	6	GDDR6	4096	64-bit	2310	1024	107	19.01.2022	199

Рис. 3.5. Панель користувача програми

У авторизованого користувача відкривається додатковий функціонал: фільтрування, порівняння та виведення детальної інформації.

Почнемо з фільтрування, для відкриття меню користувач має натиснути лівою клавішою миші на слово “Фільтри:”. Після цього відкриється додаткове меню, з яким уже можна фільтрувати записи. (Рис. 3.6)

Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$
GeForce RTX 3070	8	GDDR6	8192	192-bit	1500	3584	220	15.10.2020	499
GeForce RTX 3060	8	GDDR6	12228	192-bit	1322	3584	170	25.02.2021	329
GeForce RTX 2080 TI	12	GDDR6	12228	352-bit	1350	4352	250	23.02.2021	499
GeForce RTX 2070	12	GDDR6	8192	256-bit	1410	2304	220	23.02.2021	399

Рис. 3.7. Приклад фільтрації записів у програмі

Далі перейдемо до порівняння, для відкриття меню користувач має натиснути лівою кавішою миші на слово “Порівняння:”. Після цього можна буде обрати конкретні записи поставивши в колонці “Порівняння” галочку та вивести їх у окремому вікні, натиснувши кнопку “Порівняти”.

Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$	Порівняння
GeForce RTX 3090 TI	8	GDDR6X	24576	384-bit	1560	10752	450	27.01.2022	1699	<input type="checkbox"/>
Radeon RX 6950 XT	7	GDDR6	16384	256-bit	1925	5120	300	14.01.2021	999	<input type="checkbox"/>
GeForce RTX 3090	8	GDDR6X	24576	384-bit	1395	10496	350	24.09.2020	1499	<input checked="" type="checkbox"/>
Radeon RX 6900 XT	7	GDDR6	16384	256-bit	1825	5120	300	08.12.2020	999	<input type="checkbox"/>
Radeon RX 6800 XT	7	GDDR6	16384	256-bit	1825	4608	300	18.11.2020	649	<input checked="" type="checkbox"/>
Radeon RX 6750 XT	7	GDDR6	12228	192-bit	2321	2560	250	18.11.2020	479	<input type="checkbox"/>
Radeon RX 6800	7	GDDR6	16384	256-bit	1700	3840	250	18.11.2020	579	<input checked="" type="checkbox"/>
Radeon RX 6700 XT	7	GDDR6	12228	192-bit	2321	2560	230	17.03.2021	479	<input type="checkbox"/>
Radeon RX 6650 XT	7	GDDR6	8192	128-bit	2068	2048	120	17.03.2021	439	<input type="checkbox"/>
Radeon RX 6600 XT	7	GDDR6	8192	128-bit	1968	2048	160	10.08.2021	379	<input checked="" type="checkbox"/>
Radeon RX 6600	7	GDDR6	8192	128-bit	1626	1792	132	13.10.2021	329	<input type="checkbox"/>
Radeon RX 6500 XT	6	GDDR6	4096	64-bit	2310	1024	107	19.01.2022	199	<input type="checkbox"/>

Рис. 3.8. Меню порівняння в програмі

Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$
GeForce RTX 3090	8	GDDR6X	24576	384-bit	1395	10496	350	24.09.2020	1499
Radeon RX 6800 XT	7	GDDR6	16384	256-bit	1825	4608	300	18.11.2020	649
Radeon RX 6800	7	GDDR6	16384	256-bit	1700	3840	250	18.11.2020	579
Radeon RX 6600 XT	7	GDDR6	8192	128-bit	1968	2048	160	10.08.2021	379

Рис. 3.9. Приклад вікна порівняння

Щоб побачити більш детальну інформацію про відеокарту, користувачу потрібно двічі натиснути на назву відеокарти в таблиці та йому відкриється вікно. (Рис. 3.10)

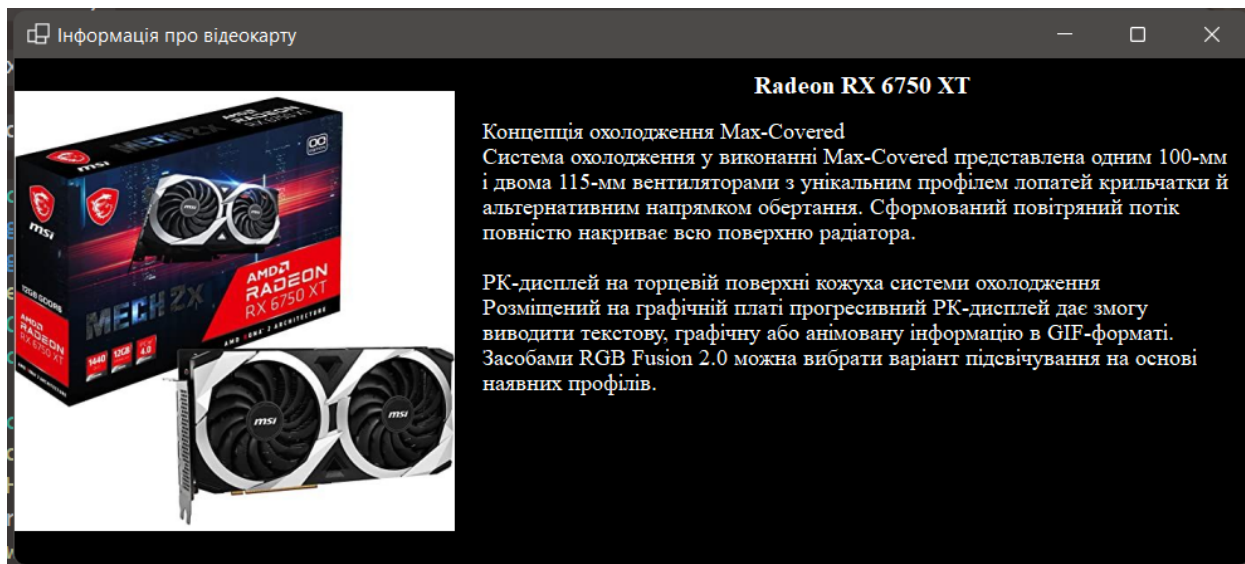


Рис. 3.10. Вікно з детальною інформацією про відеокарту

Якщо користувач є адміном, то він побачить вікно “Панель адміна” з іншим функціоналом. (Рис. 3.11.)

Панель адміна										
№	Назва	Техпроцес, нм	Тип пам'яті	Об'єм пам'яті	Шина пам'яті	Частота роботи ядра, МГц	Шейдерних блоків	Споживання енергії, Вт	Дата випуску	Вартість, \$
1	GeForce RTX 3090 TI	8	GDDR6X	24576	384-bit	1560	10752	450	27.01.2022	1699
2	Radeon RX 6950 XT	7	GDDR6	16384	256-bit	1925	5120	300	14.01.2021	999
3	GeForce RTX 3090	8	GDDR6X	24576	384-bit	1395	10496	350	24.09.2020	1499
19	Radeon RX 6900 XT	7	GDDR6	16384	256-bit	1825	5120	300	08.12.2020	999
20	Radeon RX 6800 XT	7	GDDR6	16384	256-bit	1825	4608	300	18.11.2020	649
21	Radeon RX 6750 XT	7	GDDR6	12228	192-bit	2321	2560	250	18.11.2020	479
22	Radeon RX 6800	7	GDDR6	16384	256-bit	1700	3840	250	18.11.2020	579
23	Radeon RX 6700 XT	7	GDDR6	12228	192-bit	2321	2560	230	17.03.2021	479
24	Radeon RX 6650 XT	7	GDDR6	8192	128-bit	2068	2048	120	17.03.2021	439
25	Radeon RX 6600 XT	7	GDDR6	8192	128-bit	1968	2048	160	10.08.2021	379
Додати запис						Зберегти				

Рис. 3.11. Панель адміна в програмі

Ми бачимо додаткові кнопки “Додати запис” та “Зберегти”. Натиснувши кнопку “Додати запис”, користувач побачить вікно “Додавання запису”. (Рис. 3.12)

Додавання запису

NEW

Створення запису

Назва:

Картинка:

Техпроцес:

Інформація про відеокарту:

Тип пам'яті:

Частота пам'яті:

Шина пам'яті:

Частота ядра:

Шейдерних блоків:

Енергоспоживання:

Дата випуску:
17 липня 2022 р.

Вартість:

Додати

Рис. 3.12. Вікно додавання запису

Для редагування або видалення користувачу потрібно двічі клікнути на комірку таблиці з потрібним записом. Він побачить вікно, де зможе відредагувати або видалити запис. (Рис. 3.13)

Редагування запису

Редагування запису 24

Назва:

Картинка:

Техпроцес:

Інформація про відеокарту:

Концепція охолодження Max-Covered
Система охолодження у виконанні Max-Covered представлена одним 100-мм і двома 115-мм вентиляторами з унікальним профілем лопатей крильчатки й альтернативним напрямком обертання. Сформований повітряний потік повністю накриває всю поверхню радіатора.

ПК-дисплей на торцевій поверхні кожуха системи охолодження
Розміщений на графічній платі прогресивний ПК-дисплей дає змогу виводити текстову, графічну або анімовану інформацію в GIF-форматі. Засобами RGB Fusion 2.0 можна вибрати варіант підсвічування на основі наявних профілів.

Тип пам'яті:

Частота пам'яті:

Шина пам'яті:

Частота ядра:

Шейдерних блоків:

Енергоспоживання:

Дата випуску:
17 березня 2021 р.

Вартість:

Редагувати

Видалити

Рис. 3.13. Вікно редагування запису

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Тестування роботи програмного забезпечення

Програма передбачає помилки при введенні користувачем даних для запобіження помилок. Наприклад не можна при реєстрації вказати логін, який уже є в базі даних. (Рис. 3.14)

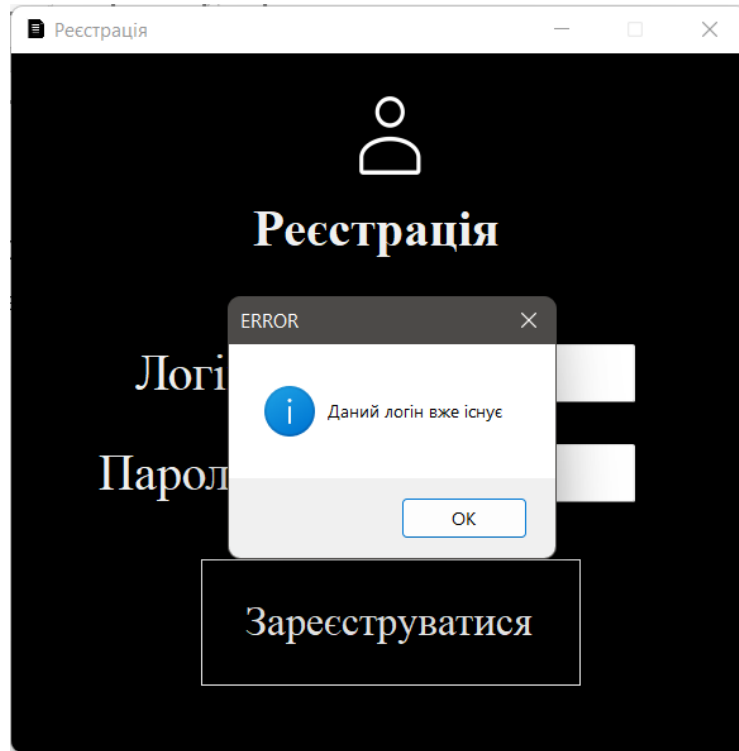


Рис. 3.14 Помилка “Даний логін вже існує”

Якщо користувач вирішить при редагуванні або додаванні запису вписати текст у поле для числа, він також зустрине помилку рисунок 3.15.

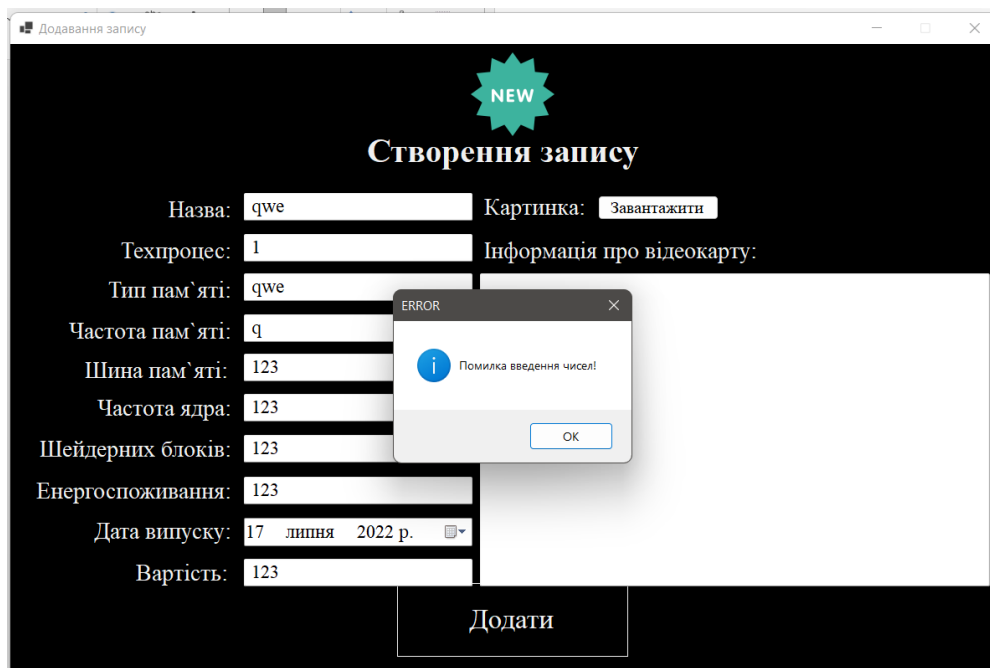


Рис. 3.15. Помилка “Помилка введення чисел!”

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

В програмі передбачено неможливість під час фільтрації вписати вартість “від” більше “до”.

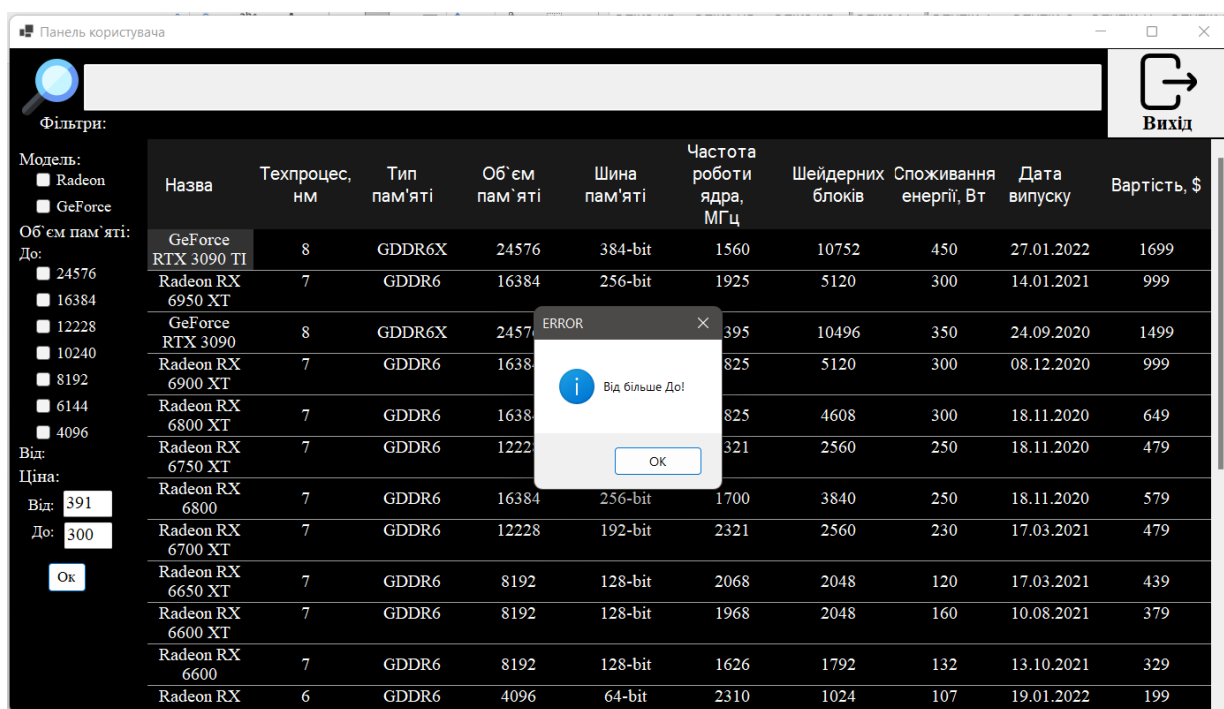


Рис. 3.16. Помилка “Від більше До!”

При введенні неіснуючого логіну або неправильного паролю під час авторизації, користувача про це сповістить програма. (Рис. 3.17)

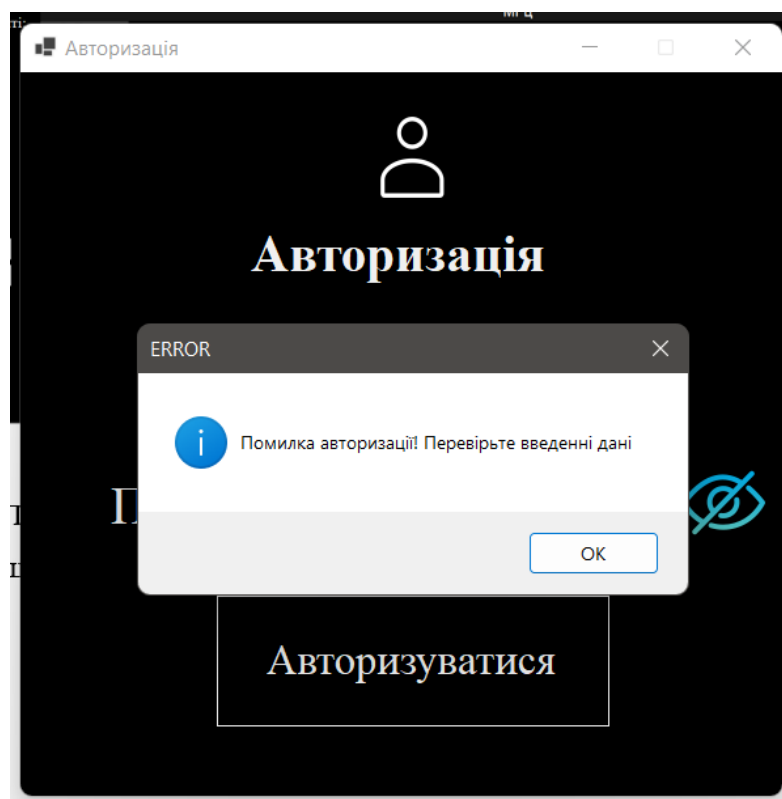


Рис. 3.17. Помилка “Помилка авторизації! Перевірьте введенні дані”

Висновки до третього розділу

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

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Отже, під час виконання курсової роботи було вдосконалено навички програмування мовою C#. 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: фільтрування, пошук, додавання записів, редагування записів, видалення записів, порівняння записів, реєстрація та авторизація.

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В.Л.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Эндрю Т. Язык Программирования С# 5.0 и платформа .NET 4.5 / Троелсен Эндрю., 2013. – 1168 с. – (6-е издание).
2. admin. MS Access и С# [Электронный ресурс] / admin. – 2018. – Режим доступа до ресурсу: <https://vscode.ru/prog-lessons/ms-access-i-c-sharp-rabotaem-sbd.html>.
3. admin. Добавление форм. Взаимодействие между формами [Электронный ресурс] / admin. – 2015. – Режим доступа до ресурсу: <https://metanit.com/sharp/windowsforms/2.3.php>.
4. admin. Создание меню MenuStrip [Электронный ресурс] / admin. – 2015. – Режим доступа до ресурсу: <https://metanit.com/sharp/windowsforms/5.2.php>.
5. admin. Контекстное меню ContextMenuStrip [Электронный ресурс] / admin. – 2015. – Режим доступа до ресурсу: <https://metanit.com/sharp/windowsforms/5.4.php>.
6. admin. Все операции с БД в графическом приложении [Электронный ресурс] / admin. – 2015. – Режим доступа до ресурсу: <https://metanit.com/sharp/adonet/3.5.php>.
7. admin. Операции со строками [Электронный ресурс] / admin. – 2015. – Режим доступа до ресурсу: <https://metanit.com/sharp/tutorial/7.2.php>.
8. 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).
9. Шилдт Герберт C# 4.0: полное руководство. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.
10. Фленов М. Библия C# / Михаил Фленов. – Питер: БХВ-Петербург, 2011. – 541 с. – (2-е изд).
11. Warren G. Подключение к данным в базе данных Access (Windows Forms) [Электронный ресурс] / Genevieve Warren. – 2019. – Режим доступа до

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В.Л.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

ресурсу: <https://docs.microsoft.com/ru-ru/visualstudio/data-tools/connect-to-data-in-an-access-database-windows-forms?view=vs-2019>.

12. admin. Отношения между классами и объектами [Электронный ресурс] / admin. – 2015. – Режим доступа до ресурсу: <https://metanit.com/sharp/patterns/1.2.php>.

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В.Л.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Бібліотека класів lib

Лістинг db.cs:

```

using MySql.Data.MySqlClient;
using System;

namespace lib
{
    public class db
    {
        MySqlConnection connection = new MySqlConnection("server = localhost; port=3306;
username=root; password = root;database =database");
        public void open()
        {
            if (connection.State == System.Data.ConnectionState.Closed)
                connection.Open();
        }
        public void close()
        {
            if (connection.State == System.Data.ConnectionState.Open)
                connection.Close();
        }
        public MySqlConnection getconnection()
        {
            return connection;
        }
    }
}

```

Лістинг Videocard.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lib
{
    public static class Videocard
    {
        public static int id;
        public static string name;
        public static int techproc;
        public static string typememory;
        public static int memory;
        public static string memorybus;
        public static int core;
        public static int shaderblock;
        public static int energy;
        public static string date;
        public static int price;
        public static string info;
        public static string link;
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

Проект WinForms

Лістинг AdminForm.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace MainForm
{
    enum Rowstatus{
        Modified,
        New,
        Deleted
    }

    public partial class AdminForm : Form
    {
        db DB = new db();
        int selectedRow;
        public string tmp;
        public AdminForm()
        {
            InitializeComponent();
        }
        private void CreateColumns()
        {
            dgv.Columns.Add("id", "№");
            dgv.Columns.Add("name", "Назва");
            dgv.Columns.Add("techproc", "Техпроцес, нм");
            dgv.Columns.Add("typememory", "Тип пам'яті");
            dgv.Columns.Add("memory", "Об'єм пам'яті");
            dgv.Columns.Add("memorybus", "Шина пам'яті");
            dgv.Columns.Add("core", "Частота роботи ядра, МГц");
            dgv.Columns.Add("shaderblock", "Шейдерних блоків");
            dgv.Columns.Add("energy", "Споживання енергії, Вт");
            dgv.Columns.Add("date", "Дата випуску");
            dgv.Columns.Add("price", "Вартість, $");
            dgv.Columns.Add("info", "info");
            dgv.Columns.Add("img", "img");
            dgv.Columns.Add("status", String.Empty);
        }

        private void ReadRow(DataGridView dgv, IDataRecord record)
        {
            dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), record.GetString(11), record.GetString(12), Rowstatus.New); ;
        }
        private void Refreshdgv(DataGridView dgv)
        {
            dgv.Rows.Clear();
            string str = $"select * from videocard";
            MySqlCommand com = new MySqlCommand(str, DB.getconnection());

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        DB.open();
        MySqlDataReader reader = com.ExecuteReader();
        while (reader.Read())
            ReadRow(dgv, reader);
        DB.close();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        CreateColumns();
        Refreshdgv(dgv);
        styledatagridview();
        dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
        dgv.AutoResizeRows(DataGridViewAutoSizeColumnsMode.AllCells);
        dgv.Columns["status"].Visible = false;
        dgv.Columns["img"].Visible = false;
        dgv.Columns["info"].Visible = false;
    }
    private void styledatagridview()
    {
        dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;
        dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;
        dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);
        dgv.DefaultCellStyle.SelectionForeColor = Color.White;
        dgv.BackgroundColor = Color.Black;
        dgv.EnableHeadersVisualStyles = false;
        dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;
        dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);
        dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);
        dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;
    }
    public void change()
    {
        int index = 0;
        WinForms.FormEdit f = new WinForms.FormEdit();
        for (int i = 0; i < dgv.RowCount; i++)
        {
            if (dgv.Rows[i].Cells[0].Value.ToString() == tmp)
            {
                index = i;
            }
        }
        dgv.Rows[index].SetValues(Videocard.id, Videocard.name, Videocard.techproc,
        Videocard.typememory, Videocard.memory, Videocard.memorybus, Videocard.core, Video-
        card.shaderblock, Videocard.energy, Videocard.date, Videocard.price);
        dgv.Rows[index].Cells[12].Value = Videocard.link;
        dgv.Rows[index].Cells[11].Value = Videocard.info;
        dgv.Rows[index].Cells[13].Value = Rowstatus.Modified;
        f.Close();
    }
    private void button_add_Click(object sender, EventArgs e)
    {
        this.Hide();
        WinForms.FormAdd f = new WinForms.FormAdd();
        f.StartPosition = FormStartPosition.CenterScreen;
        f.Show();
    }

    private void textBoxsearch_TextChanged(object sender, EventArgs e)
    {
        search(dgv);
    }
    private void search(DataGridView dgv)
    {

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В.Л.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        dgv.Rows.Clear();
        string searchquerry = $"select * from videocard where concat (id, name,
techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like
'%" + textBoxsearch.Text + "%'";
        MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());
        DB.open();
        MySqlDataReader read = com.ExecuteReader();
        while (read.Read())
        {
            ReadRow(dgv, read);
        }
        read.Close();
    }

    public void delete()
    {
        int index = 0;
        for (int i = 0; i < dgv.RowCount; i++)
        {
            if (dgv.Rows[i].Cells[0].Value.ToString() == tmp)
            {
                index = i;
            }
        }
        dgv.Rows[index].Visible = false;
        if (dgv.Rows[index].Cells[0].Value.ToString() == String.Empty)
        {
            dgv.Rows[index].Cells[13].Value = Rowstatus.Deleted;
            return;
        }
        dgv.Rows[index].Cells[13].Value = Rowstatus.Deleted;
    }

    private void update()
    {
        DB.open();
        for (int i = 0; i < dgv.Rows.Count; i++)
        {
            var rowstatus = (Rowstatus)dgv.Rows[i].Cells[13].Value;
            if (rowstatus == Rowstatus.New)
                continue;
            if (rowstatus == Rowstatus.Deleted)
            {
                var id = Convert.ToInt32(dgv.Rows[i].Cells[0].Value);
                var deletequerry = $"DELETE FROM videocard WHERE videocard.id =
{id}";
                MySqlCommand com = new MySqlCommand(deletequerry, DB.getconnec-
tion());
                com.ExecuteNonQuery();
            }
            if (rowstatus == Rowstatus.Modified)
            {
                string changequerry = $"update videocard set name = '{Video-
card.name}', techproc = '{Videocard.techproc}', typememory= '{Videocard.typememory}',
memory= '{Videocard.memory}', memorybus = '{Videocard.memorybus}', core = '{Video-
card.core}', shaderblock = '{Videocard.shaderblock}', energy = '{Videocard.energy}', date
= '{Videocard.date}', price= '{Videocard.price}', img= @img, info= '{Videocard.info}'
where id = '{Videocard.id}'";
                MySqlCommand com = new MySqlCommand(changequerry, DB.getconnec-
tion());
                com.Parameters.AddWithValue("img", Videocard.link);
                com.ExecuteNonQuery();
            }
        }
        DB.close();
        Refreshdgv(dgv);
    }

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    private void button_save_Click(object sender, EventArgs e)
    {
        update();
    }

    private void dgv_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
    {
        selectedRow = e.RowIndex;
        if (e.RowIndex >= 0)
        {
            DataGridViewRow row = dgv.Rows[selectedRow];
            Videocard.name = row.Cells[1].Value.ToString();
            Videocard.techproc = Convert.ToInt32(row.Cells[2].Value);
            Videocard.typememory = row.Cells[3].Value.ToString();
            Videocard.memory = Convert.ToInt32(row.Cells[4].Value);
            Videocard.memorybus = row.Cells[5].Value.ToString();
            Videocard.core = Convert.ToInt32(row.Cells[6].Value);
            Videocard.shaderblock = Convert.ToInt32(row.Cells[7].Value);
            Videocard.energy = Convert.ToInt32(row.Cells[8].Value);
            Videocard.date = row.Cells[9].Value.ToString();
            Videocard.price = Convert.ToInt32(row.Cells[10].Value);
            Videocard.id = Convert.ToInt32(row.Cells[0].Value);
            Videocard.link = row.Cells[12].Value.ToString();
            Videocard.info = row.Cells[11].Value.ToString();
            this.Hide();
            WinForms.FormEdit f = new WinForms.FormEdit();
            f.StartPosition = FormStartPosition.CenterScreen;
            f.Show();
        }
    }

    private void pictureBoxrefresh_Click(object sender, EventArgs e)
    {
        Refreshdgv(dgv);
    }

    private void buttonexit_Click(object sender, EventArgs e)
    {
        DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?",
        "Вихід", MessageBoxButtons.YesNo);
        if (dialogResult1 == DialogResult.Yes)
        {
            DialogResult dialogResult2 = MessageBox.Show("Ви точно бажаєте покинути
        ваш чудовий акаунт?", "Вихід", MessageBoxButtons.YesNo);
            if (dialogResult2 == DialogResult.Yes)
            {
                MainForm.StartForm f = new MainForm.StartForm();
                this.Hide();
                f.Show();
            }
        }
    }

    private void AdminForm_FormClosing(object sender, FormClosingEventArgs e)
    {
        DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?",
        "Вихід", MessageBoxButtons.YesNo);
        if (dialogResult1 == DialogResult.Yes)
        {
            Application.Exit();
        }
        else
        {
            e.Cancel = true;
        }
    }

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				40
Змн.	Арк.	№ докум.	Підпис	Дата		


```

    }
}
}

```

Лістинг FormAdd.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace WinForms
{
    public partial class FormAdd : Form
    {
        db DB = new db();
        public FormAdd()
        {
            InitializeComponent();
        }
        OpenFileDialog ofd = new OpenFileDialog();
        private void buttonedit_Click(object sender, EventArgs e)
        {
            DB.open();
            Videocard.name = textBoxname.Text;
            int techproc = Videocard.techproc;
            Videocard.typememory = textBoxmemorytype.Text;
            int memory = Videocard.memory;
            Videocard.memorybus = textBoxmemorybus.Text;
            int core = Videocard.core;
            int shaderblock = Videocard.shaderblock;
            int energy = Videocard.energy;
            Videocard.date = dateTimePicker1.Value.ToString("s");
            int price = Videocard.price;
            Videocard.info = textBoxinfo.Text;
            Videocard.link = ofd.FileName;
            if (int.TryParse(textBoxtechproc.Text, out techproc) && int.TryParse(textBox-
memory.Text, out memory) && int.TryParse(textBoxcore.Text, out core) && int.Try-
Parse(textBoxshaderblock.Text, out shaderblock) && int.TryParse(textBoxenergy.Text, out
energy) && int.TryParse(textBoxprice.Text, out price))
            {
                Videocard.techproc = techproc;
                Videocard.memory = memory;
                Videocard.core = core;
                Videocard.shaderblock = shaderblock;
                Videocard.energy = energy;
                Videocard.price = price;
                var addQuery = $"insert into videocard ( , name, techproc, typememory,
memory, memorybus, core, shaderblock, energy, date, price, info, img) values (NULL, '{Vid-
eocard.name}', '{Videocard.techproc}', '{Videocard.typememory}', '{Videocard.memory}', '{Vid-
eocard.memorybus}', '{Videocard.core}', '{Videocard.shaderblock}', '{Videocard.en-
ergy}', '{Videocard.date}', '{Videocard.price}', '{Videocard.info}', @img)";
                MySqlCommand com = new MySqlCommand(addQuery, DB.getconnection());
                com.Parameters.AddWithValue("img", Videocard.link);
                com.ExecuteNonQuery();
                MessageBox.Show("Запис успішно додано!", "Уху!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                MainForm.AdminForm form = new MainForm.AdminForm();
                this.Hide();
            }
        }
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        form.Show();
    }
    else
    {
        MessageBox.Show("Помилка введення чисел!", "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    DB.close();
}

private void FormAdd_FormClosed(object sender, FormClosedEventArgs e)
{
    MainForm.AdminForm form = new MainForm.AdminForm();
    form.Show();
}

private void buttonimg_Click(object sender, EventArgs e)
{
    ofd.Filter = "Image Files(*.JPG;*.PNG)|*.JPG;*.PNG|All files(*.*)|*.*";
    if (ofd.ShowDialog() == DialogResult.OK)
    {
        try
        {
            Videocard.link = ofd.FileName;
        }
        catch
        {
            MessageBox.Show("Неправильний формат картинки!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}
}

```

Лістинг FormEdit.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;

namespace WinForms
{
    public partial class FormEdit : Form
    {
        public int tmp;
        public FormEdit()
        {
            InitializeComponent();
        }
        OpenFileDialog ofd = new OpenFileDialog();
        private void FormAdd_Load(object sender, EventArgs e)
        {
            textBoxname.Text = Videocard.name;
            textBoxtechproc.Text = Videocard.techproc.ToString();
            textBoxmemorytype.Text = Videocard.typememory;
            textBoxmemory.Text = Videocard.memory.ToString();
            textBoxmemorybus.Text = Videocard.memorybus;
        }
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        textBoxcore.Text = Videocard.core.ToString();
        textBoxshaderblock.Text = Videocard.shaderblock.ToString();
        textBoxenergy.Text = Videocard.energy.ToString();
        dateTimePicker1.Text = Videocard.date;
        textBoxprice.Text = Videocard.price.ToString();
        labelid.Text = Videocard.id.ToString();
        textBoxinfo.Text = Videocard.info;
        ofd.FileName = Videocard.link;
    }

    private void buttonedit_Click(object sender, EventArgs e)
    {
        MainForm.AdminForm f = new MainForm.AdminForm();
        f.tmp = labelid.Text;
        Videocard.id = Convert.ToInt32(labelid.Text);
        Videocard.name = textBoxname.Text;
        int techproc = Videocard.techproc;
        Videocard.typememory = textBoxmemorytype.Text;
        int memory = Videocard.memory;
        Videocard.memorybus = textBoxmemorybus.Text;
        int core = Videocard.core;
        int shaderblock = Videocard.shaderblock;
        int energy = Videocard.energy;
        Videocard.date = dateTimePicker1.Value.ToString("s");
        int price = Videocard.price;
        Videocard.info = textBoxinfo.Text;
        Videocard.link = ofd.FileName;
        if (int.TryParse(textBoxtechproc.Text, out techproc) && int.TryParse(textBox-
memory.Text, out memory) && int.TryParse(textBoxcore.Text, out core) && int.Try-
Parse(textBoxshaderblock.Text, out shaderblock) && int.TryParse(textBoxenergy.Text, out
energy) && int.TryParse(textBoxprice.Text, out price))
        {
            Videocard.techproc = techproc;
            Videocard.memory = memory;
            Videocard.core = core;
            Videocard.shaderblock = shaderblock;
            Videocard.energy = energy;
            Videocard.price = price;
            this.Hide();
            f.Show();
            f.change();
        }
        else
        {
            MessageBox.Show("Помилка введення чисел!", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
    }

    private void buttondel_Click(object sender, EventArgs e)
    {
        MainForm.AdminForm f = new MainForm.AdminForm();
        f.tmp = labelid.Text;
        this.Hide();
        f.Show();
        f.delete();
    }

    private void FormEdit_FormClosed(object sender, FormClosedEventArgs e)
    {
        MainForm.AdminForm f = new MainForm.AdminForm();
        f.Show();
    }

    private void buttonimg_Click(object sender, EventArgs e)
    {

```

```

ofd.Filter = "Image Files(*.JPG;*.PNG)|*.JPG;*.PNG|All files(*.*)|*.*";
if(ofd.ShowDialog() == DialogResult.OK)
{
    try
    {
        Videocard.link = ofd.FileName;
    }
    catch
    {
        MessageBox.Show("Неправильний формат картинки!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}
}
}

```

Лістинг PorivForm.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace WinForms
{
    public partial class PorivForm : Form
    {
        db DB = new db();
        int[] array = new int[100];
        int j;
        int i = 0;
        public PorivForm()
        {
            InitializeComponent();
        }
        public void arrayadd(int[] a, int i)
        {
            array = a;
            j = i;
        }
        public void CreateColumns()
        {
            dgv.Columns.Add("id", "№");
            dgv.Columns.Add("name", "Назва");
            dgv.Columns.Add("techproc", "Техпроцес, нм");
            dgv.Columns.Add("typememory", "Тип пам'яті");
            dgv.Columns.Add("memory", "Об'єм пам'яті");
            dgv.Columns.Add("memorybus", "Шина пам'яті");
            dgv.Columns.Add("core", "Частота роботи ядра, МГц");
            dgv.Columns.Add("shaderblock", "Шейдерних блоків");
            dgv.Columns.Add("energy", "Споживання енергії, Вт");
            dgv.Columns.Add("date", "Дата випуску");
            dgv.Columns.Add("price", "Вартість, $");
        }
        public void ReadRow(DataGridView dgv, IDataRecord record)
        {
            dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6),

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10)); ;
    }
    private void Refreshdgv(DataGridView dgv)
    {
        string str = $"select * from videocard where id = '{array[i]}'";
        MySqlCommand com = new MySqlCommand(str, DB.getconnection());
        DB.open();
        MySqlDataReader reader = com.ExecuteReader();
        while (reader.Read())
            ReadRow(dgv, reader);
        DB.close();
    }
    private void PorivForm_Load(object sender, EventArgs e)
    {
        CreateColumns();
        while (i <= j)
        {
            Refreshdgv(dgv);
            i++;
        }
        styledatagridview();
        dgv.Columns["id"].Visible = false;
        dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
        dgv.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);
    }
    private void styledatagridview()
    {
        dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;
        dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;
        dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);
        dgv.DefaultCellStyle.SelectionForeColor = Color.White;
        dgv.BackgroundColor = Color.Black;
        dgv.EnableHeadersVisualStyles = false;
        dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;
        dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);
        dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);
        dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;
    }

    private void PorivForm_FormClosed(object sender, FormClosedEventArgs e)
    {
        MainForm.UserForm form = new MainForm.UserForm();
        form.Show();
    }
}

```

Лістинг signinform.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

namespace signinform
{
    public partial class signinform : Form
    {
        db DB = new db();
        public signinform()
        {
            InitializeComponent();
            this.textBoxpass.AutoSize = false;
            this.textBoxpass.Size = new Size(this.textBoxpass.Size.Width, 46);
        }
        public static Regex regex;

        private void button1_Click(object sender, EventArgs e)
        {
            string login = textBoxlogin.Text;
            string pass = textBoxpass.Text;
            DataTable table = new DataTable();
            MySqlDataAdapter adapter = new MySqlDataAdapter();
            MySqlCommand com = new MySqlCommand("SELECT * FROM user WHERE login=@uL AND
password=@uP", DB.getconnection());
            com.Parameters.Add("@uL", MySqlDbType.VarChar).Value = login;
            com.Parameters.Add("@uP", MySqlDbType.VarChar).Value = pass;
            adapter.SelectCommand = com;
            adapter.Fill(table);
            if (table.Rows.Count > 0)
            {
                MessageBox.Show("Авторизовано", "Успіх!", MessageBoxButtons.OK, Message-
BoxIcon.Information);
                if (login == "admin" && pass == "admin")
                {
                    this.Hide();
                    MainForm.AdminForm f = new MainForm.AdminForm();
                    f.Show();
                }
                else
                {
                    this.Hide();
                    MainForm.UserForm f = new MainForm.UserForm();
                    f.Show();
                }
            }
            else
            {
                MessageBox.Show("Помилка авторизації! Перевірте введенні дані", "ERROR",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            }

            private void signinform_FormClosed(object sender, FormClosedEventArgs e)
            {
                MainForm.StartForm form = new MainForm.StartForm();
                form.Show();
            }

            private void pictureBoxvisible_Click(object sender, EventArgs e)
            {
                if (textBoxpass.UseSystemPasswordChar == true)
                {
                    pictureBoxvisible.Image = Image.FromFile("C:\\Users\\Victor\\source\\re-
pos\\kursova\\MainForm\\images\\view.png");
                    textBoxpass.UseSystemPasswordChar = false;
                }
                else
                {
                    pictureBoxvisible.Image = Image.FromFile("C:\\Users\\Victor\\source\\re-
pos\\kursova\\MainForm\\images\\hide.png");
                }
            }
        }
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        textBoxpass.UseSystemPasswordChar = true;
    }
}
}

```

Лістинг signupform.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace signupform
{
    public partial class signupform : Form
    {
        public signupform()
        {
            InitializeComponent();
            this.textBoxpass.AutoSize = false;
            this.textBoxpass.Size = new Size(this.textBoxpass.Size.Width, 46);
        }
        public static Regex regex;
        db DB = new db();

        private void button1_Click(object sender, EventArgs e)
        {
            int tmp = 0;
            regex = new Regex(@"^[a-zA-Z][a-zA-Z0-9-_.]{1,15}$");
            if (!(regex.IsMatch(textBoxlogin.Text)))
            {
                tmp++;
                MessageBox.Show("Помилка введення логіну", "ERROR", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            }
            regex = new Regex(@"([a-z]+[A-Z]+[0-9]+|[a-z]+[0-9]+[A-Z]+|[A-Z]+[a-z]+[0-9]+|[A-Z]+[0-9]+[a-z]+|[0-9]+[a-z]+[A-Z]+|[0-9]+[A-Z]+[a-z]+)");
            if (!(regex.IsMatch(textBoxpass.Text)))
            {
                tmp++;
                MessageBox.Show("Помилка введення паролю\nМає бути щонайменше один символ верхнього регістру, один нижнього і цифра", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            DB.open();
            MySqlCommand command = new MySqlCommand("select id from user where login=@uL", DB.getconnection());
            command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = textBoxlogin.Text;
            int id = Convert.ToInt32(command.ExecuteScalar());
            if (id == 0)
            {
                MySqlCommand com = new MySqlCommand("INSERT INTO user (id, login, password) VALUES (NULL, @login, @password);", DB.getconnection());
                com.Parameters.AddWithValue("login", textBoxlogin.Text);
                com.Parameters.AddWithValue("password", textBoxpass.Text);
                if (tmp == 0)
            }
        }
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (Convert.ToInt32(com.ExecuteNonQuery()) != 0)
        {
            MessageBox.Show("Реєстрація пройшла успішно!", "Успіх!", Message-
BoxButtons.OK, MessageBoxIcon.Information);
            DB.close();
            this.Hide();
            MainForm.StartForm f = new MainForm.StartForm();
            f.Show();
        }
    }
    else
    {
        MessageBox.Show("Даний логін вже існує", "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}

private void signupform_FormClosed(object sender, FormClosedEventArgs e)
{
    MainForm.StartForm form = new MainForm.StartForm();
    form.Show();
}
}
}

```

Лістинг StartForm.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace MainForm
{
    public partial class StartForm : Form
    {
        db DB = new db();
        public StartForm()
        {
            InitializeComponent();
        }
        public void CreateColumns()
        {
            dgv.Columns.Add("id", "№");
            dgv.Columns.Add("name", "Назва");
            dgv.Columns.Add("techproc", "Техпроцес, нм");
            dgv.Columns.Add("typememory", "Тип пам'яті");
            dgv.Columns.Add("memory", "Об'єм пам'яті");
            dgv.Columns.Add("memorybus", "Шина пам'яті");
            dgv.Columns.Add("core", "Частота роботи ядра, МГц");
            dgv.Columns.Add("shaderblock", "Шейдерних блоків");
            dgv.Columns.Add("energy", "Споживання енергії, Вт");
            dgv.Columns.Add("date", "Дата випуску");
            dgv.Columns.Add("price", "Вартість, $");
            dgv.Columns.Add("status", String.Empty);
        }
        public void ReadRow(DataGridView dgv, IDataRecord record)
        {

```



```

        dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), Rowstatus.New); ;
    }
    private void Refreshdgv(DataGridView dgv)
    {
        dgv.Rows.Clear();
        string str = $"select * from videocard";
        MySqlCommand com = new MySqlCommand(str, DB.getconnection());
        DB.open();
        MySqlDataReader reader = com.ExecuteReader();
        while (reader.Read())
            ReadRow(dgv, reader);
        DB.close();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        CreateColumns();
        Refreshdgv(dgv);
        styledatagridview();
        dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
        dgv.AutoResizeRows(DataGridViewAutoSizeColumnsMode.AllCells);
        dgv.Columns["status"].Visible = false;
        dgv.Columns["id"].Visible = false;
    }
    private void styledatagridview()
    {
        dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;
        dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;
        dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);
        dgv.DefaultCellStyle.SelectionForeColor = Color.White;
        dgv.BackgroundColor = Color.Black;
        dgv.EnableHeadersVisualStyles = false;
        dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;
        dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);
        dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);
        dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;
    }
    private void buttonsignin_Click(object sender, EventArgs e)
    {
        signinform.signinform signin = new signinform.signinform();
        this.Hide();
        signin.StartPosition = FormStartPosition.CenterScreen;
        signin.Show();
    }
    private void buttonsignup_Click(object sender, EventArgs e)
    {
        signupform.signupform signup = new signupform.signupform();
        this.Hide();
        signup.StartPosition = FormStartPosition.CenterScreen;
        signup.Show();
    }
    private void textBoxsearch_TextChanged(object sender, EventArgs e)
    {
        search(dgv);
    }
    private void search(DataGridView dgv)
    {
        dgv.Rows.Clear();
        string searchquery = $"select * from videocard where concat (id, name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%" + textBoxsearch.Text + "%'";
    }

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());
        DB.open();
        MySqlDataReader read = com.ExecuteReader();
        while (read.Read())
        {
            ReadRow(dgv, read);
        }
        read.Close();
    }

    private void StartForm_FormClosing(object sender, FormClosingEventArgs e)
    {
        DialogResult dialogResult = MessageBox.Show("Ви впевнені що хочете вийти?",
"Вихід", MessageBoxButtons.YesNo);
        if (dialogResult == DialogResult.Yes)
        {
            Application.Exit();
        }
        else if (dialogResult == DialogResult.No)
        {
            e.Cancel = true;
        }
    }
}

```

Лістинг UserForm.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace MainForm
{
    public partial class UserForm : Form
    {
        public string tmp;
        int selectedrow;
        db DB = new db();
        public UserForm()
        {
            InitializeComponent();
        }
        public void CreateColumns()
        {
            dgv.Columns.Add("id", "№");
            dgv.Columns.Add("name", "Назва");
            dgv.Columns.Add("techproc", "Техпроцес, нм");
            dgv.Columns.Add("typememory", "Тип пам'яті");
            dgv.Columns.Add("memory", "Об'єм пам'яті");
            dgv.Columns.Add("memorybus", "Шина пам'яті");
            dgv.Columns.Add("core", "Частота роботи ядра, МГц");
            dgv.Columns.Add("shaderblock", "Шейдерних блоків");
            dgv.Columns.Add("energy", "Споживання енергії, Вт");
            dgv.Columns.Add("date", "Дата випуску");
            dgv.Columns.Add("price", "Вартість, $");
            dgv.Columns.Add("status", String.Empty);
        }
    }
}

```

```

        DataGridViewCheckBoxColumn dgvcheckbox = new DataGridViewCheckBoxColumn();
        dgvcheckbox.ValueType = typeof(bool);
        dgvcheckbox.Name = "check";
        dgvcheckbox.HeaderText = "Порівняння";
        dgvcheckbox.ReadOnly = false;
        dgv.Columns.Add(dgvcheckbox);
    }
    public void ReadRow(DataGridView dgv, IDataRecord record)
    {
        dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), Rowstatus.New);
    }
    private void Refreshdgv(DataGridView dgv)
    {
        dgv.Rows.Clear();
        string str = $"select * from videocard";
        MySqlCommand com = new MySqlCommand(str, DB.getconnection());
        DB.open();
        MySqlDataReader reader = com.ExecuteReader();
        while (reader.Read())
        {
            ReadRow(dgv, reader);
        }
        DB.close();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        panelform.Width = 1;
        button_poriv.Visible = false;
        buttonok.Visible = false;
        label1.Visible = false;
        label7.Visible = false;
        textBoxot.Visible = false;
        textBoxdo.Visible = false;
        checkBox_4096.Visible = false;
        checkBox_6144.Visible = false;
        checkBox_8192.Visible = false;
        checkBox_10240.Visible = false;
        checkBox_12228.Visible = false;
        checkBox_16384.Visible = false;
        checkBox_24576.Visible = false;
        checkBox_amd.Visible = false;
        checkBox_nvidia.Visible = false;
        label2.Visible = false;
        label3.Visible = false;
        label4.Visible = false;
        label5.Visible = false;
        label6.Visible = false;
        CreateColumns();
        Refreshdgv(dgv);
        styledatagridview();
        dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;
        dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
        dgv.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);
        dgv.Columns["id"].ReadOnly = true;
        dgv.Columns["name"].ReadOnly = true;
        dgv.Columns["techproc"].ReadOnly = true;
        dgv.Columns["typememory"].ReadOnly = true;
        dgv.Columns["memory"].ReadOnly = true;
        dgv.Columns["memorybus"].ReadOnly = true;
        dgv.Columns["core"].ReadOnly = true;
        dgv.Columns["shaderblock"].ReadOnly = true;
        dgv.Columns["energy"].ReadOnly = true;
    }

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        dgv.Columns["date"].ReadOnly = true;
        dgv.Columns["price"].ReadOnly = true;
        dgv.Columns["status"].Visible = false;
        dgv.Columns["id"].Visible = false;
        dgv.Columns["check"].Visible = false;
    }
    private void styledatagridview()
    {
        dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;
        dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;
        dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);
        dgv.DefaultCellStyle.SelectionForeColor = Color.White;
        dgv.BackgroundColor = Color.Black;
        dgv.EnableHeadersVisualStyles = false;
        dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;
        dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);
        dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);
        dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;
    }
    private void buttonexit_Click(object sender, EventArgs e)
    {
        DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?",
        "Вихід", MessageBoxButtons.YesNo);
        if (dialogResult1 == DialogResult.Yes)
        {
            DialogResult dialogResult2 = MessageBox.Show("Ви точно бажаєте покинути
            ваш чудовий акаунт?", "Вихід", MessageBoxButtons.YesNo);
            if (dialogResult2 == DialogResult.Yes)
            {
                MainForm.StartForm f = new MainForm.StartForm();
                this.Hide();
                f.Show();
            }
        }
    }
    private void textBoxsearch_TextChanged(object sender, EventArgs e)
    {
        search(dgv);
    }
    private void search(DataGridView dgv)
    {
        dgv.Rows.Clear();
        string searchquerry = $"select * from videocard where concat (id, name,
        techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%"
        + textBoxsearch.Text + "%'";
        MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());
        DB.open();
        MySqlDataReader read = com.ExecuteReader();
        while (read.Read())
        {
            ReadRow(dgv, read);
        }
        read.Close();
    }
    private void labelfilt_Click(object sender, EventArgs e)
    {
        if (label2.Visible == false)
        {
            labelporiv.Visible = false;
            textBoxsearch.ReadOnly = true;
            panelform.Width = 155;
            buttonok.Visible = true;
            checkBox_4096.Visible = true;
            checkBox_6144.Visible = true;
            checkBox_8192.Visible = true;
            checkBox_10240.Visible = true;
        }
    }

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        checkBox_12228.Visible = true;
        checkBox_16384.Visible = true;
        checkBox_24576.Visible = true;
        checkBox_amd.Visible = true;
        checkBox_nvidia.Visible = true;
        label2.Visible = true;
        label3.Visible = true;
        label4.Visible = true;
        label5.Visible = true;
        label6.Visible = true;
        label1.Visible = true;
        label7.Visible = true;
        textBoxot.Visible = true;
        textBoxdo.Visible = true;
        Refreshdgv(dgv);
    }
    else
    {
        labelporiv.Visible = true;
        textBoxsearch.ReadOnly = false;
        panelform.Width = 1;
        buttonok.Visible = false;
        label1.Visible = false;
        label7.Visible = false;
        textBoxot.Visible = false;
        textBoxdo.Visible = false;
        checkBox_4096.Visible = false;
        checkBox_6144.Visible = false;
        checkBox_8192.Visible = false;
        checkBox_10240.Visible = false;
        checkBox_12228.Visible = false;
        checkBox_16384.Visible = false;
        checkBox_24576.Visible = false;
        checkBox_amd.Visible = false;
        checkBox_nvidia.Visible = false;
        label2.Visible = false;
        label3.Visible = false;
        label4.Visible = false;
        label5.Visible = false;
        label6.Visible = false;
        Refreshdgv(dgv);
    }
}

private void checkBox_amd_CheckedChanged(object sender, EventArgs e)
{
    check();
}

private void checkBox_nvidia_CheckedChanged(object sender, EventArgs e)
{
    check();
}

int maxmem = 0, minmem = 25000;
int minprice = 0, maxprice = 99999;
private int findmax()
{
    if (checkBox_24576.Checked == true && Convert.ToInt32(checkBox_24576.Text) >
minmem)
    {
        maxmem = Convert.ToInt32(checkBox_24576.Text);
        return maxmem;
    }
    else if (checkBox_16384.Checked == true && Convert.ToInt32(check-
Box_16384.Text) > minmem)
    {
        maxmem = Convert.ToInt32(checkBox_16384.Text);
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return maxmem;
    }
    if (checkBox_12228.Checked == true && Convert.ToInt32(checkBox_12228.Text) >
minmem)
    {
        maxmem = Convert.ToInt32(checkBox_12228.Text);
        return maxmem;
    }
    else if (checkBox_10240.Checked == true && Convert.ToInt32(check-
Box_10240.Text) > minmem)
    {
        maxmem = Convert.ToInt32(checkBox_10240.Text);
        return maxmem;
    }
    else if (checkBox_8192.Checked == true && Convert.ToInt32(checkBox_8192.Text)
> minmem)
    {
        maxmem = Convert.ToInt32(checkBox_8192.Text);
        return maxmem;
    }
    else if (checkBox_6144.Checked == true && Convert.ToInt32(checkBox_6144.Text)
> minmem)
    {
        maxmem = Convert.ToInt32(checkBox_6144.Text);
        return maxmem;
    }
    else if (checkBox_4096.Checked == true && Convert.ToInt32(checkBox_4096.Text)
> minmem)
    {
        maxmem = Convert.ToInt32(checkBox_4096.Text);
        return maxmem;
    }
    else return maxmem = 25000;
}
private int findmin()
{
    if (checkBox_4096.Checked == true && Convert.ToInt32(checkBox_4096.Text) <
minmem)
    {
        minmem = Convert.ToInt32(checkBox_4096.Text);
        return minmem;
    }
    else if (checkBox_6144.Checked == true && Convert.ToInt32(checkBox_6144.Text)
< minmem)
    {
        minmem = Convert.ToInt32(checkBox_6144.Text);
        return minmem;
    }
    else if (checkBox_8192.Checked == true && Convert.ToInt32(checkBox_8192.Text)
< minmem)
    {
        minmem = Convert.ToInt32(checkBox_8192.Text);
        return minmem;
    }
    else if (checkBox_10240.Checked == true && Convert.ToInt32(check-
Box_10240.Text) < minmem)
    {
        minmem = Convert.ToInt32(checkBox_10240.Text);
        return minmem;
    }
    else if (checkBox_12228.Checked == true && Convert.ToInt32(check-
Box_12228.Text) < minmem)
    {
        minmem = Convert.ToInt32(checkBox_12228.Text);
        return minmem;
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        else if (checkBox_16384.Checked == true && Convert.ToInt32(check-
Box_16384.Text) < minmem)
        {
            minmem = Convert.ToInt32(checkBox_16384.Text);
            return minmem;
        }
        else if (checkBox_24576.Checked == true && Convert.ToInt32(check-
Box_24576.Text) < minmem)
        {
            minmem = Convert.ToInt32(checkBox_24576.Text);
            return minmem;
        }
        else return minmem = 0;
    }

    private void check()
    {
        maxmem = 0;
        int max = findmax();
        minmem = 25000;
        int min = findmin();
        if (checkBox_nvidia.Checked == false && checkBox_amd.Checked == true)
        {
            dgv.Rows.Clear();
            string filterquery = $"select * from videocard where memory between
{minmem} and {maxmem} and price between {minprice} and {maxprice} and name like '%" +
checkBox_amd.Text + "%'";
            MySqlCommand com = new MySqlCommand(filterquery, DB.getconnection());
            DB.open();
            MySqlDataReader read = com.ExecuteReader();
            while (read.Read())
            {
                ReadRow(dgv, read);
            }
            read.Close();
        }
        else if (checkBox_nvidia.Checked == true && checkBox_amd.Checked == false)
        {
            dgv.Rows.Clear();
            string filterquery = $"select * from videocard where memory between
{minmem} and {maxmem} and price between {minprice} and {maxprice} and name like '%" +
checkBox_nvidia.Text + "%'";
            MySqlCommand com = new MySqlCommand(filterquery, DB.getconnection());
            DB.open();
            MySqlDataReader read = com.ExecuteReader();
            while (read.Read())
            {
                ReadRow(dgv, read);
            }
            read.Close();
        }
        else
        {
            dgv.Rows.Clear();
            string filterquery = $"select * from videocard where memory between
{minmem} and {maxmem} and price between {minprice} and {maxprice}";
            MySqlCommand com = new MySqlCommand(filterquery, DB.getconnection());
            DB.open();
            MySqlDataReader read = com.ExecuteReader();
            while (read.Read())
            {
                ReadRow(dgv, read);
            }
            read.Close();
        }
    }
}

```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В.Л.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private void checkBox_24576_CheckedChanged(object sender, EventArgs e)
{
    check();
}
private void checkBox_16384_CheckedChanged(object sender, EventArgs e)
{
    check();
}
private void checkBox_12228_CheckedChanged(object sender, EventArgs e)
{
    check();
}
private void checkBox_10240_CheckedChanged(object sender, EventArgs e)
{
    check();
}
private void checkBox_8192_CheckedChanged(object sender, EventArgs e)
{
    check();
}
private void button1_Click(object sender, EventArgs e)
{
    if (textBoxdo.Text != String.Empty && textBoxot.Text != String.Empty)
    {
        if (Convert.ToInt32(textBoxot.Text) >= Convert.ToInt32(textBoxdo.Text))
        {
            MessageBox.Show("Від більше До!", "ERROR", MessageBoxButtons.OK, Mes-
sageBoxIcon.Information);
        }
        else
        {
            minprice = int.Parse(textBoxot.Text);
            maxprice = int.Parse(textBoxdo.Text);
            check();
        }
    }
}
private void UserForm_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?",
"Вихід", MessageBoxButtons.YesNo);
    if (dialogResult1 == DialogResult.Yes)
    {
        Application.Exit();
    }
    else
    {
        e.Cancel = true;
    }
}
private void checkBox_6144_CheckedChanged(object sender, EventArgs e)
{
    check();
}
private void labelporiv_Click(object sender, EventArgs e)
{
    if (dgv.Columns["check"].Visible == false)
    {
        button_poriv.Visible = true;
        textBoxsearch.Visible = false;
        labelfilt.Visible = false;
        dgv.Columns["check"].Visible = true;
    }
    else
    {
        button_poriv.Visible = false;
    }
}

```


Лістинг Videocardinfo.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using lib;
using MySql.Data.MySqlClient;

namespace WinForms
{
    public partial class Videocardinfo : Form
    {
        db DB = new db();
        public Videocardinfo()
        {
            InitializeComponent();

            private void Videocard_Load(object sender, EventArgs e)
            {
                pictureBox1.Image = Image.FromFile(Videocard.link);
                labelname.Text = Videocard.name;
                labelinfo.Text = Videocard.info;
                pictureBox1.Focus();
            }

            private void Videocardinfo_FormClosed(object sender, FormClosedEventArgs e)
            {
                MainForm.UserForm form = new MainForm.UserForm();
                form.Show();
            }
        }
    }
}
```

		Новіцький В.С.			«Житомирська політехніка». 22.125.25.000 – ПЗ	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		58