**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ**

**ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І**

**СИСТЕМ**

**КУРСОВА РОБОТА**

**з дисципліни**

**«Об’єктно-орієнтоване програмування»**

**спеціальність 122 «Комп’ютерні науки »**

**Студента групи 202-ТН Потягова Дмитра Миколайовича**

Керівник роботи

Демиденко М.І.

Полтава – 2025

Оглавление

[Вступ 3](#_Toc198558816)

[АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 5](#_Toc198558817)

[ПОСТАНОВКА ЗАДАЧІ 6](#_Toc198558818)

[ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 8](#_Toc198558819)

[ОСНОВНІ ФРАГМЕНТИ ПРОГРАМНОГО КОДУ 13](#_Toc198558820)

[ТЕСТУВАННЯ ПРОГРАМИ 16](#_Toc198558821)

[ВИСНОВКИ 18](#_Toc198558822)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 19](#_Toc198558823)

# Вступ

У сучасному світі інформаційних технологій обробка та зберігання даних займає одне з ключових місць у програмній інженерії. Створення зручних додатків для роботи з інформацією є актуальним завданням для фахівців у галузі програмування. Особливої популярності набули застосунки, що дозволяють керувати базами даних у зручній графічній формі — з інтуїтивним інтерфейсом, швидким пошуком і можливістю додавання нової інформації.

Мова програмування **C# (C-Sharp)** була обрана для реалізації цього проєкту завдяки своїй об’єктно-орієнтованій природі, простому синтаксису та тісній інтеграції з платформою **.NET Framework**. C# є сучасною мовою, що поєднує потужність C++ із простотою Java, та забезпечує високу продуктивність при розробці десктопних застосунків.

Для реалізації курсової роботи використано середовище розробки **Microsoft Visual Studio 2022** — один із найпотужніших інструментів для створення застосунків на базі Windows Forms. Visual Studio надає зручні засоби для візуального проєктування форм, налаштування елементів керування та написання коду з використанням інтерактивної перевірки помилок.

У межах курсового проєкту було створено програму під назвою **«Бібліотека»**, призначену для ведення простої бази даних книг. Програма реалізує такі функції, як:

* додавання нової книги;
* перегляд повного списку книг;
* пошук книги за назвою;
* пошук книги за автором.

Для реалізації логіки збереження даних використовувався окремий файл-клас **BookStorage.cs**, у якому міститься статичний список книг. Основні форми, з якими працює користувач — **Form1.cs (головне вікно)**, **SearchForm.cs (панель навігації)**, **AddBookForm.cs (додавання книг)**, **ViewBooksForm.cs (перегляд)**, **SearchByAuthorForm.cs (пошук за автором)** — створено із застосуванням Windows Forms із відповідною обробкою подій для кожної кнопки.

Уся структура програми побудована таким чином, щоб користувач міг легко керувати своєю «бібліотекою» без необхідності використання баз даних або зовнішніх джерел.

Метою цієї курсової роботи є закріплення навичок об’єктно-орієнтованого програмування, роботи з подієвим підходом, формами Windows Forms і побудова зручного графічного інтерфейсу для обліку та пошуку книг.

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

У сучасних умовах стрімкого розвитку інформаційних технологій дедалі більшої актуальності набуває автоматизація процесів зберігання та обробки даних у різних сферах діяльності. Однією з таких сфер є бібліотечна справа, де потреба в електронному обліку літератури, швидкому пошуку й упорядкуванні книжкових фондів стоїть особливо гостро.

Традиційне ведення бібліотечного каталогу у вигляді паперових карток або журналів є не лише застарілим, а й незручним у використанні, особливо при великих об’ємах інформації. Пошук певної книги, сортування за автором чи назвою, а також додавання нових позицій вимагає значних витрат часу та ресурсів. Саме тому актуальним є створення програмного забезпечення, яке забезпечить зручну взаємодію з книжковим фондом.

Програма «Бібліотека», реалізована в рамках даної курсової роботи, імітує роботу електронного каталогу книг. Користувач має змогу:

* додавати нові книги із зазначенням основних характеристик (назва, автор, жанр, кількість сторінок);
* здійснювати пошук книг за назвою або автором;
* переглядати список усіх наявних у базі книг.

Уся інформація зберігається у динамічному списку, що дозволяє швидко обробляти введені дані без складного налаштування баз даних або серверної частини.

Подібні додатки можуть застосовуватись не лише в бібліотеках, а й у книжкових магазинах, приватних архівах, освітніх закладах або будь-де, де є потреба в обліку літературних ресурсів. В умовах навчального проєкту така система є чудовим прикладом практичного застосування знань з об’єктно-орієнтованого програмування, роботи з формами, контролями та структурованими даними у вигляді об’єктів класу Book.

# ПОСТАНОВКА ЗАДАЧІ

Метою даної курсової роботи є розробка прикладного програмного забезпечення для моделювання роботи електронної бібліотеки, що забезпечує базовий функціонал обліку та пошуку книг. Основна увага приділяється розробці графічного інтерфейсу користувача, реалізації основних функцій керування даними, а також забезпеченню зручності та простоти використання програми.

**Основні завдання, які необхідно реалізувати в програмі:**

* **Розробити графічний інтерфейс користувача** з використанням Windows Forms, який дозволить взаємодіяти з програмою без потреби писати код або використовувати командний рядок.
* **Реалізувати збереження списку книг** у вигляді об'єктів класу Book, що містить поля:
  + назва книги;
  + автор;
  + жанр;
  + кількість сторінок.
* **Створити форму додавання нової книги**, яка дозволить користувачеві ввести всю необхідну інформацію та додати книгу до загального списку.
* **Реалізувати перегляд усіх книг**, що містяться в списку, у вигляді таблиці з повною інформацією.
* **Розробити функції пошуку книг за автором та за назвою**, що дозволяють знайти відповідні записи у списку згідно з критерієм пошуку.
* **Забезпечити обробку помилок**, зокрема:
  + перевірку правильності введених даних;
  + виведення повідомлень у разі відсутності результатів пошуку;
  + сповіщення користувача про успішне додавання книги.
* **Забезпечити навігацію між вікнами** програми через кнопки, що відкривають відповідні форми.

Програма повинна бути написана мовою програмування C# з використанням середовища розробки Visual Studio 2022. Усі елементи інтерфейсу повинні бути зручно розміщені, а функціональність — логічно структурованою та легкою у використанні.

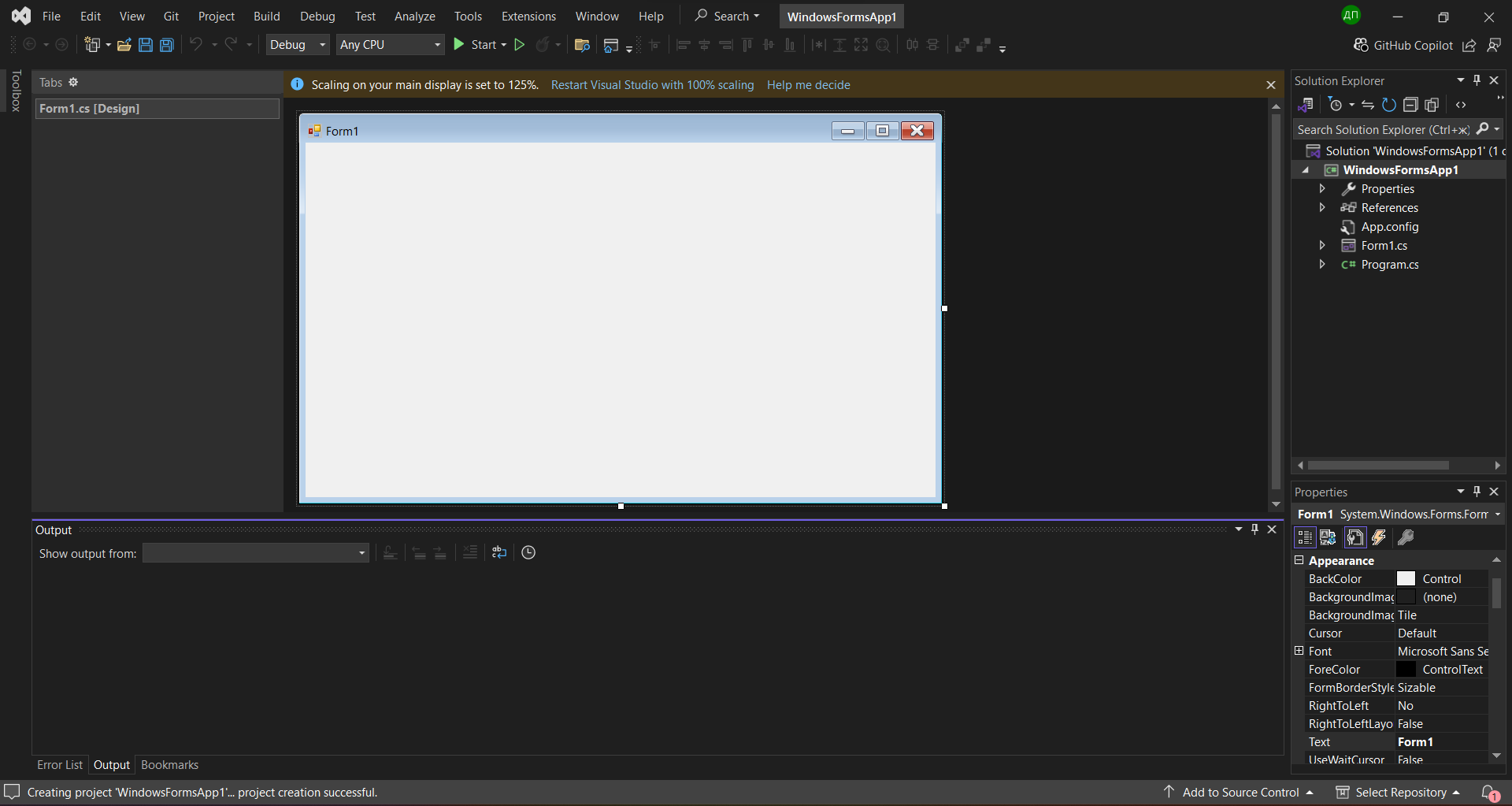
# ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

* + **Вибір середовища розробки**

Для реалізації курсової роботи було обрано сучасне та функціональне середовище розробки **Microsoft Visual Studio 2022**, яке підтримує написання програм мовою C# та має зручний візуальний редактор форм — **Windows Forms Designer**. Visual Studio дозволяє ефективно створювати графічні інтерфейси, пов’язувати елементи керування з кодом і реалізовувати подієво-орієнтовану логіку.

Крім цього, середовище підтримує автоматичну перевірку синтаксису, відлагодження, управління структурою проєкту, а також містить засоби керування ресурсами, шрифтами, кольорами, подіями тощо.

Вибір Visual Studio зумовлений його стабільною роботою, підтримкою багатьох типів проєктів і широким використанням у професійній розробці програмного забезпечення. *(рис. 1).*



*(рис. 1) – Головне вікно Visual Studio з відкритим проєктом*

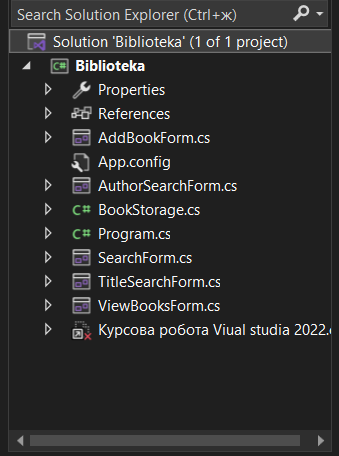
* + **Структура програми**

Програма реалізована у вигляді багатовіконного застосунку. Головним елементом структури є **класи форм**, кожен із яких відповідає за певну частину функціональності. Дані зберігаються у статичному списку, оголошеному в окремому класі **BookStorage**. Усі форми використовують доступ до цього списку для відображення, пошуку або додавання книг.

* + **Основні елементи структури:**
* Book.cs — модель книги;
* BookStorage.cs — сховище всіх книг;
* Form1.cs — стартове вікно програми;
* SearchForm.cs — головна панель керування функціоналом;
* AddBookForm.cs — форма додавання книги;
* ViewBooksForm.cs — перегляд усіх книг;
* SearchByAuthorForm.cs — пошук за автором;
* SearchByTitleForm.cs — пошук за назвою.

Усі дані існують у пам’яті під час роботи програми (без використання зовнішньої бази даних).

*(рис. 2)*



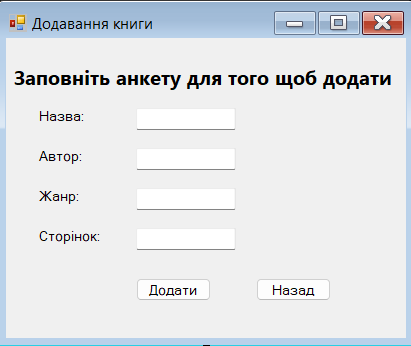
*(рис. 2)– Структура файлів у Solution Explore*

* + **.Інтерфейс користувача**

Інтерфейс програми реалізований засобами Windows Forms. Усі елементи — кнопки, текстові поля, мітки (Label), таблиці (DataGridView) — були додані через візуальний редактор. Стиль витриманий у єдиному оформленні для кожного вікна, що забезпечує зручність та інтуїтивне розуміння взаємодії.

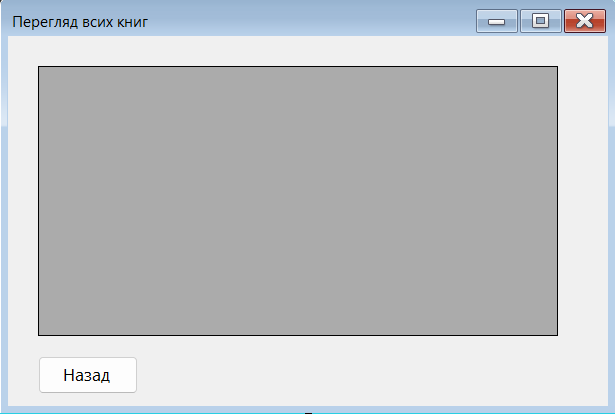
На формі Form1 є лише назва програми — "Бібліотека" — та кнопка "Розпочати роботу". Після цього відкривається SearchForm, де користувач може обрати бажану дію (додати книгу, переглянути всі, пошук за автором чи назвою).

У вікні додавання книги присутні поля для введення назви, автора, жанру та кількості сторінок, а також кнопки "Додати" й "Назад"*(рис. 3)*.

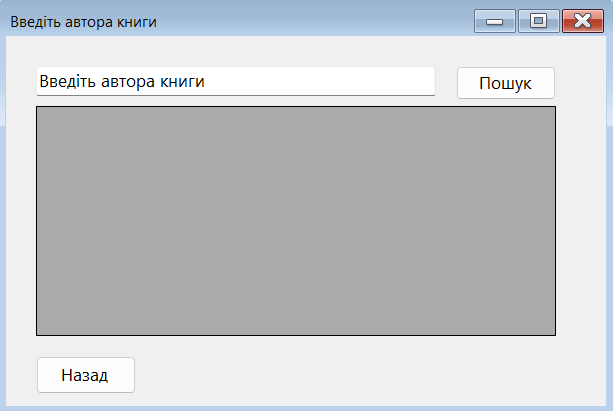


*(рис. 3) – Вікно додавання книг)*

Вікна перегляду та пошуку реалізовані з використанням DataGridView, що дозволяє зручно відображати результати.



*(рис. 4) – Вікно перегляду всіх книг*

  
*(рис. 5) – Вікно пошуку за автором*

* + **Основні форми та їх функції**

**Form1.cs** — головне стартове вікно. Містить тільки кнопку «Розпочати роботу», яка відкриває основне меню.  
**SearchForm.cs** — вікно з чотирма кнопками: додати книгу, переглянути книги, пошук за автором, пошук за назвою.  
**AddBookForm.cs** — дозволяє ввести інформацію про нову книгу. Перевіряє коректність введених даних та додає книгу до списку.  
**ViewBooksForm.cs** — відображає всі книги у вигляді таблиці з назвами, авторами, жанрами та кількістю сторінок.  
**SearchByAuthorForm.cs** — пошук книг за автором. Відображає результат у таблиці або показує повідомлення, якщо книг не знайдено.  
**SearchByTitleForm.cs** — аналогічне попередньому, але пошук відбувається за назвою книги.

Усі форми пов’язані між собою через кнопки "Назад" або "Перехід", реалізовані методом this.Hide(); new Form().Show();.

# ОСНОВНІ ФРАГМЕНТИ ПРОГРАМНОГО КОДУ

* + **Додавання книги**

Для додавання нової книги створена форма AddBookForm, у якій реалізовані текстові поля введення назви, автора, жанру та кількості сторінок. Кнопка «Додати» перевіряє правильність даних та додає об'єкт Book до загального списку.

using System.Windows.Forms;

using System;

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

{

string title = txtTitle.Text.Trim();

string author = txtAuthor.Text.Trim();

string genre = txtGenre.Text.Trim();

int pages;

if (string.IsNullOrWhiteSpace(title) ||

string.IsNullOrWhiteSpace(author) ||

string.IsNullOrWhiteSpace(genre) ||

!int.TryParse(txtPages.Text, out pages) || pages <= 0)

{

MessageBox.Show("Будь ласка, заповніть всі поля правильно!",

"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

Book newBook = new Book

{

Title = title,

Author = author,

Genre = genre,

Pages = pages

};

BookStorage.Books.Add(newBook);

MessageBox.Show("Книгу успішно додано!",

"Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTitle.Clear();

txtAuthor.Clear();

txtGenre.Clear();

txtPages.Clear();

}

* + **Перегляд усіх книг**

У формі ViewBooksForm використовується елемент DataGridView, до якого прив’язано таблицю, створену з полів кожної книги в списку.

using System.Data;

private void LoadBooks()

{

DataTable table = new DataTable();

table.Columns.Add("Назва");

table.Columns.Add("Автор");

table.Columns.Add("Жанр");

table.Columns.Add("Кількість сторінок");

foreach (var book in BookStorage.Books)

{

table.Rows.Add(book.Title, book.Author, book.Genre, book.Pages);

}

dataGridViewBooks.DataSource = table;

}

Функція викликається автоматично при завантаженні форми. Користувач бачить таблицю з усіма книгами, що зберігаються в системі.

* + **Пошук за назвою книги**

Форма SearchByTitleForm дозволяє здійснювати пошук книг за ключовими словами в назві. Результат виводиться в таблицю або повідомленням, якщо книга не знайдена.

using System.Data;

using System.Windows.Forms;

using System;

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

{

string query = txtTitle.Text.Trim().ToLower();

var results = BookStorage.Books

.Where(b => b.Title.ToLower().Contains(query))

.ToList();

if (results.Count == 0)

{

MessageBox.Show("Книгу не знайдено!",

"Результат", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

DataTable table = new DataTable();

table.Columns.Add("Назва");

table.Columns.Add("Автор");

table.Columns.Add("Жанр");

table.Columns.Add("Кількість сторінок");

foreach (var book in results)

{

table.Rows.Add(book.Title, book.Author, book.Genre, book.Pages);

}

dataGridViewResults.DataSource = table;

}

* + **Пошук за автором**

Аналогічно реалізовано пошук за автором. Збіги виводяться у вигляді таблиці.

using System.Data;

using System.Windows.Forms;

using System;

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

{

string query = txtAuthor.Text.Trim().ToLower();

var results = BookStorage.Books

.Where(b => b.Author.ToLower().Contains(query))

.ToList();

if (results.Count == 0)

{

MessageBox.Show("Автор не знайдений!",

"Результат", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

DataTable table = new DataTable();

table.Columns.Add("Назва");

table.Columns.Add("Автор");

table.Columns.Add("Жанр");

table.Columns.Add("Кількість сторінок");

foreach (var book in results)

{

table.Rows.Add(book.Title, book.Author, book.Genre, book.Pages);

}

dataGridViewResults.DataSource = table;

}

# ТЕСТУВАННЯ ПРОГРАМИ

Після реалізації основного функціоналу програма «Бібліотека» була протестована з метою перевірки її працездатності та коректності виконання передбачених дій. Основним завданням тестування є виявлення помилок на етапі виконання, а також перевірка відповідності роботи інтерфейсу очікуваному функціоналу.

У ході тестування були перевірені такі аспекти:

* робота кнопок навігації між формами;
* додавання книг з валідними і невалідними даними;
* пошук за автором та назвою (наявність/відсутність результатів);
* коректне відображення таблиць у DataGridView;
* очищення полів після додавання запису;
* повернення з підформ у головне меню.

Було проведено тестування з різними сценаріями введення даних:

1. **Додавання книги з правильними даними** — програма успішно додавала запис, з’являлось повідомлення про успіх.
2. **Спроба додати книгу з порожніми полями** — з’являлося повідомлення про помилку, книга не додавалася.
3. **Пошук існуючого автора** — таблиця результатів виводила коректну інформацію.
4. **Пошук неіснуючого автора** — з’являлося повідомлення «Автор не знайдений».
5. **Пошук за частковим збігом у назві книги** — спрацьовувало коректно.
6. **Перегляд усіх книг** — у таблиці відображалися всі книги, включаючи додані під час роботи.

Тестування показало, що програма стабільно функціонує у межах поставленого завдання та відповідає очікуваному результату. Всі перевірені сценарії були реалізовані з урахуванням обробки помилок, що підвищує надійність системи.

# ВИСНОВКИ

У результаті виконання курсової роботи було розроблено прикладний програмний продукт — програму «Бібліотека» мовою програмування C# з використанням технології Windows Forms у середовищі розробки Visual Studio 2022.

Під час реалізації було досягнуто наступних результатів:

* створено зручний та зрозумілий графічний інтерфейс користувача;
* реалізовано повний цикл роботи з даними: додавання, збереження, пошук, перегляд;
* забезпечено валідацію введених даних для уникнення помилок введення;
* організовано обробку виняткових ситуацій (порожні поля, неправильні значення тощо);
* протестовано програму на відповідність усім функціональним вимогам.

Програма є повністю функціональною, працює стабільно, має інтуїтивно зрозумілу навігацію та забезпечує основні потреби користувача у веденні невеликої бази книг.

Курсова робота дозволила поглибити знання з об’єктно-орієнтованого програмування, освоїти роботу з подієво-орієнтованими формами, списками об’єктів, взаємодією між формами, а також основами побудови простих інформаційних систем.

Таким чином, поставлену мету курсової роботи досягнуто, усі функціональні вимоги виконано, а здобуті навички можуть бути застосовані у подальшій практичній діяльності програміста.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Яворський В. Д. Об’єктно-орієнтоване програмування: навч. посіб. — К.: Кондор, 2021. — 328 с.
2. Шарп Р. Програмування на C#. Повний курс. — СПб.: Пітер, 2020. — 864 с.
3. MSDN Documentation – <https://learn.microsoft.com/>
4. Albahari J. C# 10 in a Nutshell. — O’Reilly Media, 2022. — 1056 p.
5. Методичні вказівки до оформлення письмових робіт для студентів спеціальностей 122, 123, 125. — Полтава: НУПП, 2024. — 22 с.
6. Соловйов В. І. Програмування мовою C# у середовищі Visual Studio. — Харків: ХНУРЕ, 2022. — 256 с.
7. Документація Microsoft Visual Studio 2022 – <https://visualstudio.microsoft.com/>