**УДУ імені М.П. Драгоманова**

**Факультет математики, інформатики та фізики**

***Кафедра комп’ютерної та програмної інженерії***

**ЛАБОРАТОРНА РОБОТА №5**

**ТЕМА: Використання LINQ та PLINQ**

з курсу

***«Робота з даними на платформі .NET»***

***Виконала:***

***Студентка:***Джола К.О.

***Група:*** 42ІПЗ

***Перевірив:***

***Викладач:*** Січкаренко В.О.

**Київ 2024**

***Мета***: Ознайомитись з синтаксисом LINQ та PLINQ. Застосування LINQ та PLINQ в процесі розробки міжплатформного програмного забезпечення на платформі .NET.

***Завдання:***

1. Створити новий консольний проект на мові програмування C#;
2. Визначити та створити необхідні колекції, які будуть використовуватись в обраній темі;
3. Використати мінімум 3 методи розширення вибірки даних (на власний розсуд);
4. Використати мінімум 3 методи розширення зміни порядку даних (на власний розсуд);
5. Використати мінімум 2 методи розширення вибірки даних (на власний розсуд);
6. Використати мінімум 1 метод розширення управління запитами (на власний розсуд);
7. Продемонструвати роботу програми;

**Хід роботи:**

1. **Створити новий консольний проект на мові програмування C#**

Створимо консольний проект у Visual Studio 2022, для цього необхідно натиснути кнопку “Create New Project”, та в пошуку написати “Console App” після чого обрати тип проекту Console App (рис. 1.1).

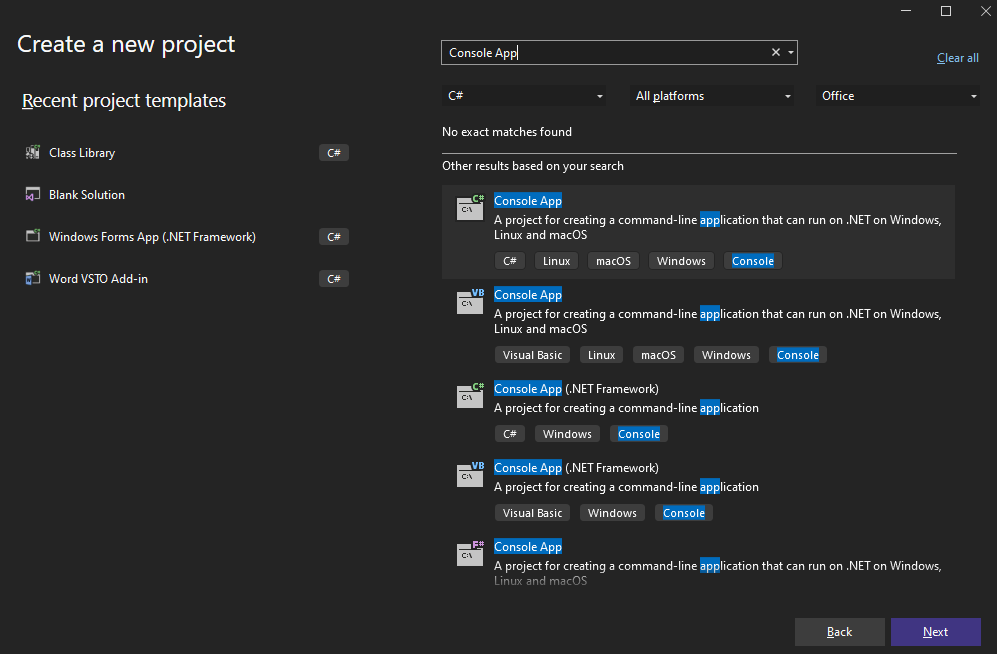


Рис. 1.1 – Створення консольного проекту

На наступному кроці вказуємо назву проекту та обираємо директорію, в якій буде зберігатися проект. Після чого оберемо цільовий фреймворк, для якого буде розроблятись бібліотека класів. Так як .NET Core 3.1 більше не підтримується, обираємо .NET 8.0 та натискаємо кнопку “Create” (рис. 1.2).

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Рис. 1.2 – Вибір платформи

Після успішного створення проекту буде відкрито файл за замовчуванням з назвою Program.cs. Даний файл можна перейменувати як-завгодно. В “Solution Explorer” можна додати будь-яку кількість файлів організованих в будь-яку структуру.

1. **Визначити та створити необхідні колекції, які будуть використовуватись в обраній темі**

В якості теми було обрано Створення системи бібліографічного обліку. Створені колекції показані на рис. 2.1.

Зображення, що містить знімок екрана, текст

Автоматично згенерований опис

Рис. 2.1 – Створені колекції

1. **Використати мінімум 3 методи розширення вибірки даних (на власний розсуд)**

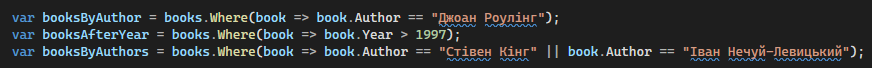


Рис. 3.1 – Методи розширення вибірки даних

books.Where(book => book.Author == "Джоан Роулінг"); – вибирає усі книги, написані автором Джоан Роулінг.

books.Where(book => book.Year > 1997); – вибирає усі книги, видані після 1997 року.

books.Where(book => book.Author == "Стівен Кінг" || book.Author == "Іван Нечуй-Левицький"); – вибирає усі книги, написані Стівеном Кінгом або Іваном Нечуй-Левицьким.

1. **Використати мінімум 3 методи розширення зміни порядку даних (на власний розсуд)**

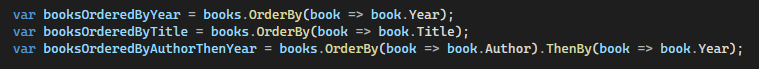


Рис. 4.1 – Методи розширення зміни порядку даних

books.OrderBy(book => book.Year); – сортує книги за роком видання.

books.OrderBy(book => book.Title); – сортує книги за назвою.

books.OrderBy(book => book.Author).ThenBy(book => book.Year); – сортує книги спочатку за автором, а потім за роком видання

1. **Використати мінімум 2 методи розширення вибірки даних (на власний розсуд)**



Рис. 5.1 – Методи розширення вибірки даних

books.Count(book => book.Author == "Стівен Кінг"); – кількість книг, написаних Стівеном Кінгом.

books.Any(book => book.Year > 2005); – чи є книги, видані після 2005 року.

1. **Використати мінімум 1 метод розширення управління запитами (на власний розсуд)**



Рис. 6.1 – Метод розширення управління запитом

books.First(book => book.Author == "Джоан Роулінг"); - показує першу книгу від Джоан Роулінг.

1. **Продемонструвати роботу програми**

Зображення, що містить текст, знімок екрана, меню, Шрифт

Автоматично згенерований опис

Рис. 7.1 – Демонстрація роботи програми

***Відповіді на контрольні запитання:***

1. **Основні операції опрацювання даних.**

Під час виконання програми можуть використовуватись різні дані, які потрібно опрацьовувати певним чином. До опрацювання можна віднести операції:

1. Керування даними (додавання, видалення, сортування, групування, переміщення);
2. Пошук даних;
3. Модифікація даних (оновлення, зміна типу даних або вмісту, форматування);
4. Отримання метаданих (кількість даних, максимальне/мінімальне значення, перший/останній елементи і т.д.);
5. **Що таке мова запитів? Які існують мови запитів?**

Мова запитів (query language) – система символів (алфавіт) та команд, призначених для опрацювання даних з різних джерел. До мов запитів відносяться:

1. ***SQL (Structured Query Language)*** – мова запитів до реляційних баз даних. Складається з DDL (Data Definition Language – створення даних), DML (Data Modification Language – модифікація даних), DQL (Data Query Language – отримання даних);
2. ***LINQ/PLINQ (Language Integrated Query/Parallel Language Integrated Query)*** – сукупність механізмів платформи .NET для маніпулювання даними з різних джерел (колекції та масиви (LINQ to Objects), бази даних (LINQ to DataSet, LINQ to SQL, LINQ to Entities), файли (LINQ to XML)). Реалізований у вигляді методів розширення (extension method). Використання LINQ в кожній окремій .NET-сумісній мові відрізняєтсья. PLINQ – версія LINQ для опрацювання даних в паралельному режимі;
3. ***XQuery, XPath*** – мови запитів для управління даними з різних джерел (веб-служби, бази даних, файли), які використовують XML-модель (доступ до елементів, структура). XPath, на даний момент є частиною XQuery;
4. **Для яких джерел даних існують розширення LINQ/PLINQ?**

LINQ (Language Integrated Query) та PLINQ (Parallel Language Integrated Query) в C# можуть використовуватися з різними джерелами даних: вбудовані колекції .NET, XML-документи, SQL-бази даних, Entity Framework, ADO.NET DataSet, паралельні колекції тощо.

1. **Різниця між LINQ to SQL , LINQ to DataSet, LINQ to Entities.**

Основні відмінності між ними:

***LINQ to SQL:*** Це провайдер LINQ, який дозволяє виконувати запити до баз даних SQL Server. Він працює з базами даних на основі рядків і стовпців і може виконувати операції, такі як вставка, видалення, оновлення та вибірка. LINQ to SQL використовує об’єктно-реляційне відображення (ORM) для перетворення таблиць бази даних на класи C# і навпаки.

***LINQ to DataSet:*** Це провайдер LINQ, який дозволяє виконувати запити до ADO.NET DataSets та DataTables. DataSet - це в пам’яті представлення даних, яке може містити декілька таблиць даних. LINQ to DataSet дозволяє використовувати синтаксис LINQ для фільтрації, сортування, групування та інших операцій з даними в DataSet.

***LINQ to Entities:*** Це провайдер LINQ, який дозволяє виконувати запити до баз даних за допомогою Entity Framework. Entity Framework - це ORM-рішення від Microsoft, яке дозволяє працювати з даними на більш високому рівні абстракції, ніж LINQ to SQL. LINQ to Entities підтримує більше видів баз даних, ніж LINQ to SQL, і має більш потужні можливості, такі як підтримка наслідування, багато до багатьох відносин та інше.

1. **Різниця між використанням методів розширення та синтаксису запитів LINQ/PLINQ.**

***Методи розширення (extension methods)*** – взаємодія з даними відбувається за допомогою послідовних викликів необхідних методів, в які передаються відповідні параметри (зазвичай, лямбда-вирази). Оскільки не всі .NET-сумісні мови підтримують методи розширення, то цей підхід може відрізнятись за синтаксисом (наприклад, в C++/CLI для цього використовуються прямий доступ до методу інтерфейсу IEnumerable, як приклад, Enumerable::FirstOrDefault).

***Синтаксис запитів (query)*** – взаємодія з даними відбувається за допомого послідовності ключових слів (keywords). Цей декларативний підхід побудований на методах розширення і використовується в мові програмування C#.

1. **Призначення ключового слова let під час використання синтаксису запитів LINQ/PLINQ?**

Ключове слово let – зміна діапазону, для збереження проміжних резульатів запитів. Може використовуватись тільки всередині запитів.

1. **Призначення методів розширення вибірки. Приклади таких методів.**

Методи розширення вибірки в LINQ призначені для фільтрації, вибору та перетворення даних у вихідні результати. Приклади:

* Where: Вибирає елементи, що відповідають певному умові.
* Select: Трансформує кожен елемент в нову форму за допомогою вказаного виразу.
* Take: Вибирає перші N елементів з вихідної послідовності.
* Skip: Пропускає перші N елементів у вихідній послідовності.
* Distinct: Видаляє дублікати з послідовності.

1. **Призначення методів розширення зміни порядку. Приклади таких методів.**

Методи розширення для зміни порядку в LINQ і PLINQ призначені для сортування, фільтрації та реорганізації даних у послідовностях. Приклади методів:

* OrderBy: Сортує елементи послідовності за зростанням.
* OrderByDescending: Сортує елементи послідовності за спаданням.
* ThenBy: Додає додаткове сортування за вказаним критерієм (при використанні після OrderBy або OrderByDescending).
* ThenByDescending: Додає додаткове сортування за вказаним критерієм у спадаючому порядку (при використанні після OrderBy або OrderByDescending).
* Reverse: Змінює порядок елементів у послідовності на протилежний.

1. **Призначення методів агрегації. Приклади таких методів.**

Методи агрегації використовуються для обчислення загальних значень або результатів операцій над елементами послідовності. Приклади методів:

* Count: Підраховує кількість елементів у послідовності.
* Sum: Обчислює суму значень числових елементів.
* Average: Обчислює середнє значення числових елементів.
* Min: Знаходить мінімальне значення у послідовності.
* Max: Знаходить максимальне значення у послідовності.

1. **Призначення методів управління запитами. Приклади таких методів.**

Методи управління запитами в LINQ дозволяють виконувати операції, що впливають на виконання або результати запиту. Приклади:

* OrderBy: Визначає порядок сортування результатів запиту.
* Skip: Пропускає задану кількість елементів у результаті запиту.
* Take: Вибирає задану кількість елементів з результату запиту.
* Distinct: Видаляє дублікати з результату запиту.
* Where: Фільтрує результати запиту за заданою умовою.