

Лабораторна робота № 3

Тема: LINQ – мова інтегрованих запитів

Мета: отримати практичні навички з використання LINQ

Практична частина

1. Створіть колекцію об'єктів класу Person, яка містить такі поля, як ім'я, прізвище, вік, місто.
2. Напишіть запит, який вибере всіх людей з міста «Київ» та відсортує їх за прізвищем.
3. Напишіть запит, який знайде всіх людей, які є старше 30 років та проживають у місті «Львів», відсортовані по імені.
4. Напишіть запит, який знайде всі унікальні міста, де проживають люди в колекції.
5. Напишіть запит, який знайде найбільш поширене ім'я серед людей у колекції.
6. Напишіть запит, який знайде кількість людей у колекції, які проживають у місті «Київ».
7. Напишіть запит, який вибере перші 5 людей, які проживають у місті «Львів».
8. Напишіть запит, який вибере останні 3 людей, які проживають у місті «Одеса».
9. Напишіть запит, який вибере всіх людей, чий вік ділиться на 3 без залишку та відсортує їх за спаданням віку.
10. Напишіть запит, який вибере всіх людей з міста «Київ» та порахує загальний вік цих людей.

Додатково: У вас є дві колекції: «студенти» та «оцінки». Колекція «студенти» містить ім'я та ідентифікатор кожного студента. Колекція «оцінки» містить ідентифікатор студента та його оцінку за певний предмет. Вам потрібно використовуючи LINQ, знайти всіх студентів, які мають оцінки вище 90 балів.

Завдання:

```
class Person
{
    public string? FirstName { get; set; }
    public string? LastName { get; set; }
    public int Age { get; set; }

    public string? City { get; set; }
}

Student
{
    public int Id { get; set; }
    public string? Name { get; set; }
}

class Grade
{
    public int StudentId { get; set; }
    public int Score { get; set; }
}

class Program
{
    static void Main()
    {
        Console.OutputEncoding = System.Text.Encoding.Default;

        List<Person> people =
        [
            new Person { FirstName = "Іван", LastName = "Петренко", Age = 25,
                City = "Київ" },
            new Person { FirstName = "Олена", LastName = "Іванова", Age = 32,
                City = "Львів" },
            new Person { FirstName = "Андрій", LastName = "Сидоренко", Age = 29,
                City = "Одеса" },
            new Person { FirstName = "Марія", LastName = "Коваль", Age = 41, City
                = "Київ" },
            new Person { FirstName = "Петро", LastName = "Бондар", Age = 34, City
                = "Львів" },
            new Person { FirstName = "Світлана", LastName = "Гончар", Age = 23,
                City = "Київ" },
            new Person { FirstName = "Ірина", LastName = "Мельник", Age = 22,
                City = "Львів" },
            new Person { FirstName = "Олег", LastName = "Шевченко", Age = 45,
                City = "Одеса" },
            new Person { FirstName = "Іван", LastName = "Захарченко", Age = 27,
                City = "Харків" },
            new Person { FirstName = "Наталія", LastName = "Литвин", Age = 36,
                City = "Київ" }
        ];

        List<Student> students =
        [
            new Student { Id = 1, Name = "Іван" },
            new Student { Id = 2, Name = "Марія" },
            new Student { Id = 3, Name = "Олег" },
            new Student { Id = 4, Name = "Ірина" }
        ];

        List<Grade> grades =
        [
            new Grade { StudentId = 1, Score = 95 },
            new Grade { StudentId = 2, Score = 87 },
            new Grade { StudentId = 3, Score = 91 },
        ]
    }
}
```

```

        new Grade { StudentId = 4, Score = 78 }
    ];

while (true)
{
    Console.WriteLine("\n\n=== Меню запитів ===");
    Console.WriteLine("1 - Люди з Києва (сортування за прізвищем)");
    Console.WriteLine("2 - Старші за 30 у Львові (сортування за ім'ям)");
    Console.WriteLine("3 - Унікальні міста");
    Console.WriteLine("4 - Найбільш поширене ім'я");
    Console.WriteLine("5 - Кількість людей у Києві");
    Console.WriteLine("6 - Перші 5 людей зі Львова");
    Console.WriteLine("7 - Останні 3 людини з Одеси");
    Console.WriteLine("8 - Люди з віком, кратним 3 (за спаданням)");
    Console.WriteLine("9 - Загальний вік людей з Києва");
    Console.WriteLine("a - Додаткове: Студенти з оцінками > 90");
    Console.WriteLine("n - Вихід із програми");
    Console.Write("\nВаш вибір: ");

    string? choice = Console.ReadLine();
    Console.WriteLine();

    switch (choice)
    {
        case "1":
            people
                .Where(p => p.City == "Київ")
                .OrderBy(p => p.LastName)
                .ToList()
                .ForEach(p => Console.WriteLine($"{p.FirstName}
{p.LastName} - {p.City}"));
            break;

        case "2":
            people
                .Where(p => p.City == "Львів" && p.Age > 30)
                .OrderBy(p => p.FirstName)
                .ToList()
                .ForEach(p => Console.WriteLine($"{p.FirstName}
{p.LastName} - {p.Age}"));
            break;

        case "3":
            people
                .Select(p => p.City)
                .Distinct()
                .ToList()
                .ForEach(city => Console.WriteLine(city));
            break;

        case "4":
            var commonName = people
                .GroupBy(p => p.FirstName)
                .OrderByDescending(g => g.Count())
                .First().Key;
            Console.WriteLine($"Найбільш поширене ім'я: {commonName}");
            break;

        case "5":
            var count = people.Count(p => p.City == "Київ");
            Console.WriteLine($"Кількість людей у Києві: {count}");
            break;

        case "6":
            people

```

```

        .Where(p => p.City == "Львів")
        .Take(5)
        .ToList()
        .ForEach(p => Console.WriteLine($"{p.FirstName}
{p.LastName}"));
        break;

        case "7":
            people
                .Where(p => p.City == "Одеса")
                .Reverse()
                .Take(3)
                .ToList()
                .ForEach(p => Console.WriteLine($"{p.FirstName}
{p.LastName}"));
            break;

        case "8":
            people
                .Where(p => p.Age % 3 == 0)
                .OrderByDescending(p => p.Age)
                .ToList()
                .ForEach(p => Console.WriteLine($"{p.FirstName}
{p.LastName} - {p.Age}"));
            break;

        case "9":
            int totalAge = people
                .Where(p => p.City == "Київ")
                .Sum(p => p.Age);
            Console.WriteLine($"Загальний вік людей з Києва:
{totalAge}");
            break;

        case "a":
            var topStudents = students
                .Join(grades,
                    s => s.Id,
                    g => g.StudentId,
                    (s, g) => new { s.Name, g.Score })
                .Where(x => x.Score > 90)
                .ToList();
            Console.WriteLine("Студенти з оцінками > 90:");
            topStudents.ForEach(s => Console.WriteLine($"{s.Name} -
{s.Score}"));
            break;

        case "n":
            Console.WriteLine("Програма завершена.");
            return;

        default:
            Console.WriteLine("Невірний вибір. Спробуйте ще раз.");
            break;
    }
}
}
}

```

На рисунку 3.1 зображений результат виконання завдання 1.

```
=== Меню запитів ===
1 - Люди з Києва (сортування за прізвищем)
2 - Старші за 30 у Львові (сортування за ім'ям)
3 - Унікальні міста
4 - Найбільш поширене ім'я
5 - Кількість людей у Києві
6 - Перші 5 людей зі Львова
7 - Останні 3 людини з Одеси
8 - Люди з віком, кратним 3 (за спаданням)
9 - Загальний вік людей з Києва
a - Додаткове: Студенти з оцінками > 90
n - Вихід із програми

Ваш вибір: 1

Світлана Гончар - Київ
Марія Коваль - Київ
Наталія Литвин - Київ
Іван Петренко - Київ
```

Рисунок 3.1 – Люди з Києва (сортування за прізвищем)

На рисунку 3.2 зображений результат виконання завдання 2.

```
Ваш вибір: 2

Олена Іванова - 32
Петро Бондар - 34
```

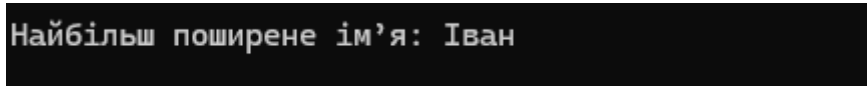
Рисунок 3.2 – Старші за 30 у Львові (сортування за ім'ям)

На рисунку 3.3 зображений результат виконання завдання 3.

```
Київ
Львів
Одеса
Харків
```

Рисунок 3.3 – Унікальні міста

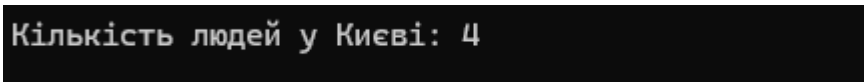
На рисунку 3.4 зображений результат виконання завдання 4.



```
Найбільш поширене ім'я: Іван
```

Рисунок 3.4 – Найбільш поширене ім'я


На рисунку 3.5 зображений результат виконання завдання 5.



```
Кількість людей у Києві: 4
```

Рисунок 3.5 – Кількість людей у Києві


На рисунку 3.6 зображений результат виконання завдання 6.



```
Олена Іванова  
Петро Бондар  
Ірина Мельник
```

Рисунок 3.6 – Перші 5 людей зі Львова

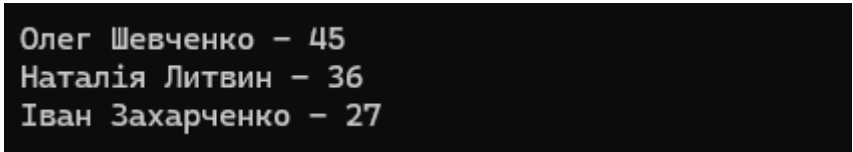
На рисунку 3.7 зображений результат виконання завдання 7.



```
Олег Шевченко  
Андрій Сидоренко
```

Рисунок 3.7 – Останні 3 людини з Одеси


На рисунку 3.8 зображений результат виконання завдання 8.



```
Олег Шевченко - 45  
Наталія Литвин - 36  
Іван Захарченко - 27
```

Рисунок 3.8 – Люди з віком, кратним 3 (за спаданням)

На рисунку 3.9 зображений результат виконання завдання 9.



```
Загальний вік людей з Києва: 125
```

Рисунок 3.9 – Загальний вік людей з Києва

На рисунку 3.10 зображений результат виконання додаткового завдання.

```
Студенти з оцінками > 90:  
Іван - 95  
Олег - 91
```

Рисунок 3.10 – Студенти з оцінками більше 90

На рисунку 3.11 зображені дані таблиць застосунку.

```
Дані таблиці People:  
Ім'я: Іван Петренко, Вік: 25, Місто: Київ  
Ім'я: Олена Іванова, Вік: 32, Місто: Львів  
Ім'я: Андрій Сидоренко, Вік: 29, Місто: Одеса  
Ім'я: Марія Коваль, Вік: 41, Місто: Київ  
Ім'я: Петро Бондар, Вік: 34, Місто: Львів  
Ім'я: Світлана Гончар, Вік: 23, Місто: Київ  
Ім'я: Ірина Мельник, Вік: 22, Місто: Львів  
Ім'я: Олег Шевченко, Вік: 45, Місто: Одеса  
Ім'я: Іван Захарченко, Вік: 27, Місто: Харків  
Ім'я: Наталія Литвин, Вік: 36, Місто: Київ  
  
Дані таблиці Students:  
ID: 1, Ім'я: Іван  
ID: 2, Ім'я: Марія  
ID: 3, Ім'я: Олег  
ID: 4, Ім'я: Ірина  
  
Дані таблиці Grades:  
ID Студента: 1, Оцінка: 95  
ID Студента: 2, Оцінка: 87  
ID Студента: 3, Оцінка: 91  
ID Студента: 4, Оцінка: 78
```

Рисунок 3.11 – Дані таблиць

Контрольні питання:

1. Дайте визначення поняттю LINQ

LINQ (Language Integrated Query) — це технологія в .NET, яка дозволяє писати запити до колекцій даних (масивів, списків, баз даних, XML тощо) безпосередньо в коді C#, використовуючи уніфікований синтаксис.

2. Основне призначення LINQ

Основне призначення LINQ — забезпечити єдиний, зручний і типобезпечний спосіб доступу та обробки даних з різних джерел (об'єкти в пам'яті, SQL-бази даних, XML-документи тощо).

3. В чому полягає відмінність LINQ від PLINQ?

- LINQ (стандартний) виконує запити послідовно, один елемент за іншим.
- PLINQ (Parallel LINQ) виконує запити паралельно, використовуючи кілька ядер процесора для прискорення обробки великих колекцій. PLINQ реалізується за допомогою простору імен `System.Linq.Parallel`.

4. Що представляє собою оператор запиту LINQ?

Оператор запиту LINQ — це вираз (часто з ключовими словами `from`, `where`, `select` тощо), який визначає умови вибірки, фільтрації, сортування та перетворення елементів з колекції або джерела даних у результат.