

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра Системотехніки  
Дисципліна: «Програмування на платформі .NET»

ЗВІТ З ПРАКТИЧНОЇ РОБОТИ №5  
«Колекції. Використання LINQ. Серіалізація»

Виконав:  
Студент групи КНТ-21-1  
Глова Станіслав Олександрович

Перевірив  
асис. каф.  
Калайда Н. С.

Оцінка \_\_\_\_\_  
«\_\_\_\_» \_\_\_\_\_ 2023 р.

Харків 2023

## 5. КОЛЕКЦІЇ. ВИКОРИСТАННЯ LINQ.СЕРІАЛІЗАЦІЯ

### 5.1 Мета роботи

Навчитися створювати та працювати з колекціями, аналізувати колекції використовуючи LINQ та використовувати серіалізацію для зберігання даних.

### 5.2 Організація самостійної роботи студентів

Під час підготовки до виконання лабораторної роботи необхідно вивчити існуючі колекції, методи для роботи з колекціями. Навчитися обробляти інформацію використовуючи мову запитів LINQ та зберігати дані за допомогою серіалізації об'єктів.

### 5.3 Індивідуальні завдання

#### Завдання №1

Створити клас Student з полями (властивостями) ім'я – Name (String), вік – Age (Int), колекція предмет-оцінка – Marks (Dictionary) та конструктором для створення об'єктів. Вважати, що у кожного студента три предмети: «Mathematics», «Philosophy», «English», а оцінка варіюється від 1 до 100. В головній функції програми створити колекцію об'єктів Student (не менше 10).

#### Завдання №2

Для створеної колекції об'єктів із завдання 1 провести наступні операції використовуючи LINQ:

- вибрати з колекції всі елементи, для яких довжина ім'я більше ніж 4 символи та вік менше 20. Результат відсортувати за ім'ям в зворотному напрямі;
- відсортувати студентів за оцінкою по математиці;
- отримати кількість студентів, які не склали хоча б один іспит;
- розрахувати та вивести середній бал по кожному з предметів;
- згрупувати студентів за віком та відобразити результат з вказанням кількості елементів в кожній групі;

#### Завдання №3

Серіалізувати колекцію об'єктів створену в завд. 1 в json файл. Врахувати, що поле Marks не серіалізується, а для поля Name замістити оригінальну назву на FirstName. Провести десеріалізацію отриманого файлу та вивести результат на екран.

### 5.4 Хід виконання

Створив новий консольний проект в середовищі JetBrains Rider 2022.3.2 та почав писати реалізацію завдань. Труднощів не виникало. Виконав усі

завдання та продемонстрував створення колекцій, маніпулювання даними в колекціях, використання LINQ та серіалізація/десеріалізацію в/з JSON формату.

## 5.5 Результат виконання програми

Task №1

```
Student{ name = Name 1, age = 23, marks = {Mathematics = 15,2, Philosophy = 97,7, English = 69,1} }
Student{ name = Name 2, age = 18, marks = {Mathematics = 77,4, Philosophy = 20,5, English = 86,6} }
Student{ name = Name 3, age = 19, marks = {Mathematics = 16,9, Philosophy = 52,6, English = 73,2} }
Student{ name = Name 4, age = 19, marks = {Mathematics = 74,9, Philosophy = 95,1, English = 78} }
Student{ name = Name 5, age = 18, marks = {Mathematics = 96,6, Philosophy = 88,7, English = 32,3} }
Student{ name = Name 6, age = 18, marks = {Mathematics = 5,9, Philosophy = 47,3, English = 17,6} }
Student{ name = Name 7, age = 22, marks = {Mathematics = 41,5, Philosophy = 81,6, English = 80,2} }
Student{ name = Name 8, age = 20, marks = {Mathematics = 62,3, Philosophy = 21,3, English = 43,1} }
Student{ name = Name 9, age = 17, marks = {Mathematics = 55,5, Philosophy = 62,2, English = 86,5} }
Student{ name = Name 10, age = 21, marks = {Mathematics = 49,3, Philosophy = 99,6, English = 5,6} }
Student{ name = Name 11, age = 21, marks = {Mathematics = 96,8, Philosophy = 1,2, English = 49,9} }
Student{ name = Name 12, age = 19, marks = {Mathematics = 12,9, Philosophy = 28,6, English = 87,8} }
Student{ name = Name 13, age = 23, marks = {Mathematics = 21,8, Philosophy = 56,1, English = 20} }
Student{ name = Name 14, age = 22, marks = {Mathematics = 10,7, Philosophy = 5,4, English = 19} }
Student{ name = Name 15, age = 19, marks = {Mathematics = 71,9, Philosophy = 75,7, English = 25,5} }
```

Task №2

Всі студенти для яких довжина ім'я більше ніж 4 символи та вік менше 20 у відсортованому за ім'ям в зворотному порядку:

```
Student{ name = Name 9, age = 17, marks = {Mathematics = 55,5, Philosophy = 62,2, English = 86,5} }
Student{ name = Name 6, age = 18, marks = {Mathematics = 5,9, Philosophy = 47,3, English = 17,6} }
Student{ name = Name 5, age = 18, marks = {Mathematics = 96,6, Philosophy = 88,7, English = 32,3} }
Student{ name = Name 4, age = 19, marks = {Mathematics = 74,9, Philosophy = 95,1, English = 78} }
Student{ name = Name 3, age = 19, marks = {Mathematics = 16,9, Philosophy = 52,6, English = 73,2} }
Student{ name = Name 2, age = 18, marks = {Mathematics = 77,4, Philosophy = 20,5, English = 86,6} }
Student{ name = Name 15, age = 19, marks = {Mathematics = 71,9, Philosophy = 75,7, English = 25,5} }
Student{ name = Name 12, age = 19, marks = {Mathematics = 12,9, Philosophy = 28,6, English = 87,8} }
```

Всі студенти відсортовані за оцінкою по математиці:

```
Student{ name = Name 6, age = 18, marks = {Mathematics = 5,9, Philosophy = 47,3, English = 17,6} }
Student{ name = Name 14, age = 22, marks = {Mathematics = 10,7, Philosophy = 5,4, English = 19} }
Student{ name = Name 12, age = 19, marks = {Mathematics = 12,9, Philosophy = 28,6, English = 87,8} }
Student{ name = Name 1, age = 23, marks = {Mathematics = 15,2, Philosophy = 97,7, English = 69,1} }
Student{ name = Name 3, age = 19, marks = {Mathematics = 16,9, Philosophy = 52,6, English = 73,2} }
Student{ name = Name 13, age = 23, marks = {Mathematics = 21,8, Philosophy = 56,1, English = 20} }
Student{ name = Name 7, age = 22, marks = {Mathematics = 41,5, Philosophy = 81,6, English = 80,2} }
Student{ name = Name 10, age = 21, marks = {Mathematics = 49,3, Philosophy = 99,6, English = 5,6} }
Student{ name = Name 9, age = 17, marks = {Mathematics = 55,5, Philosophy = 62,2, English = 86,5} }
Student{ name = Name 8, age = 20, marks = {Mathematics = 62,3, Philosophy = 21,3, English = 43,1} }
Student{ name = Name 15, age = 19, marks = {Mathematics = 71,9, Philosophy = 75,7, English = 25,5} }
Student{ name = Name 4, age = 19, marks = {Mathematics = 74,9, Philosophy = 95,1, English = 78} }
Student{ name = Name 2, age = 18, marks = {Mathematics = 77,4, Philosophy = 20,5, English = 86,6} }
Student{ name = Name 5, age = 18, marks = {Mathematics = 96,6, Philosophy = 88,7, English = 32,3} }
Student{ name = Name 11, age = 21, marks = {Mathematics = 96,8, Philosophy = 1,2, English = 49,9} }
```

Кількість студентів, які не склали хоча б один іспит дорівнює 14

Середній бал по кожному з предметів наступний:

Mathematics: 47,306667

Philosophy: 55,573334

English: 51,626667

Є наступні вікові групи студентів та кількість студентів в них:

23: 2

18: 3

19: 4

22: 2

20: 1

17: 1

21: 2

```
Task №3
Student{ name = Name 1, age = 23, marks = {} }
Student{ name = Name 2, age = 18, marks = {} }
Student{ name = Name 3, age = 19, marks = {} }
Student{ name = Name 4, age = 19, marks = {} }
Student{ name = Name 5, age = 18, marks = {} }
Student{ name = Name 6, age = 18, marks = {} }
Student{ name = Name 7, age = 22, marks = {} }
Student{ name = Name 8, age = 20, marks = {} }
Student{ name = Name 9, age = 17, marks = {} }
Student{ name = Name 10, age = 21, marks = {} }
Student{ name = Name 11, age = 21, marks = {} }
Student{ name = Name 12, age = 19, marks = {} }
Student{ name = Name 13, age = 23, marks = {} }
Student{ name = Name 14, age = 22, marks = {} }
Student{ name = Name 15, age = 19, marks = {} }
```

Рисунок 5.1 – Результат виконання всіх 3 завдань

З повним кодом програми можна ознайомитися на [GitHub repository](#).

### **Висновок.**

Під час практичного заняття були набуті наступні вміння: використовування та створення колекцій, аналізування колекцій за допомогою LINQ, серіалізувати об'єкти для зберігання даних та десеріалізувати дані.

В цілому, базові практичні навички із взаємодії з колекціями, аналізу за допомогою LINQ та серіалізацією в C# були набуті.