

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»

«Програмна технологія DOTNET»

Звіт з лабораторної роботи №5

Тема: «Серіалізація в C#. Делегати»

Виконав:
ст. гр. КІТ-118в
Соколенко Д.Г.

Перевірив:
Бартош М.В.

Харків – 2020

Мета: отримати практичний досвід у розробці програм на С# і консольних додатків через додавання у попередню роботу серіалізацію і роботу з делегатами.

Загальне завдання:

- Забезпечити відображення у виді таблиці даних особистих справ усіх студентів вибраної групи, спеціальності, факультету, вузу.
- Реалізувати групове видалення особистих справ студентів (по групі, спеціальності, факультету, вузу або власному критерію).
- Продемонструвати ефективне використання делегатів і забезпечити:
 - розрахунок середнього віку усіх студентів обраної групи, спеціальності, факультету, вузу;
 - розрахунок середньої успішності усіх студентів обраної групи, спеціальності, факультету, вузу;
- Забезпечити автоматичний режим тестування
- Для збереження/відновлення колекції використовувати серіалізацію/десеріалізацію.

Хід роботи

1. Спочатку була створена можливість відображення у виді таблиці даних особистих справ усіх студентів вибраної групи, спеціальності, факультету чи повний список студентів (рис.1-4).

```
Make a choice: 1
Choose:
0 - By group
1 - By faculty
2 - By speciality
3 - All students
Make a choice: 3
```

#	Surname	Name	Patronymic	Date of birth	Enter date	Group index	Faculty	Specialty	Academic performance
0	Sokolenko	Dmytro	Alexandrovich	03.04.2001	01.09.2019	V	CIT	123Sys	100
1	T	T	T	02.02.2003	03.03.2019	C	KN	125Mach	100
2	T	T	T	02.02.2001	03.03.2018	C	CIT	123Sys	80

Рисунок 1 – повний список студентів

```
Make a choice: 1
Choose:
0 - By group
1 - By faculty
2 - By speciality
3 - All students
Make a choice: 0
Enter criteria:
C
```

#	Surname	Name	Patronymic	Date of birth	Enter date	Group index	Faculty	Specialty	Academic performance
0	T	T	T	02.02.2003	03.03.2019	C	KN	125Mach	100
1	T	T	T	02.02.2001	03.03.2018	C	CIT	123Sys	80

Рисунок 2 – список студентів обраної групи

```
Make a choice: 1
Choose:
0 - By group
1 - By faculty
2 - By speciality
3 - All students
Make a choice: 1
Enter criteria:
CIT
```

#	Surname	Name	Patronymic	Date of birth	Enter date	Group index	Faculty	Specialty	Academic performance
0	Sokolenko	Dmytro	Alexandrovich	03.04.2001	01.09.2019	V	CIT	123Sys	100
1	T	T	T	02.02.2001	03.03.2018	C	CIT	123Sys	80

Рисунок 3 – список студентів обраного факультету

```
Make a choice: 1
Choose:
0 - By group
1 - By faculty
2 - By speciality
3 - All students
Make a choice: 2
Enter criteria:
125Mach
```

#	Surname	Name	Patronymic	Date of birth	Enter date	Group index	Faculty	Specialty	Academic performance
0	T	T	T	02.02.2003	03.03.2019	C	KN	125Mach	100

Рисунок 4 – список студентів обраної спеціальності

- Далі було реалізовано групове видалення особистих справ студентів (по групі, спеціальності, факультету, вузу або власному критерію) (рис.5-7).

```
case '6':
    PrintOptions();
    Console.WriteLine("Make a choice: ");
    intChoice = Io.InputInt();

    Console.WriteLine("Enter criteria: ");
    strChoice = Console.ReadLine();
    pigsty.DeleteStudents(StudentContainer.Search(strChoice, pigsty, intChoice));
    break;
```

Рисунок 5 – виклик функції видалення студентів з параметрами: видалення з контейнера його частини.

ссылка: 1

```
public void DeleteStudents(StudentContainer removing)
{
    for (int i = 0; i < removing.Students.Length; i++)
    {
        for (int j = 0; j < Students.Length; j++)
        {
            if (this.Students[j].Equals(removing.Students[i]))
            {
                DeleteStudent(j);
            }
        }
    }
}
```

Рисунок 6 – функція видалення студентів

```
Ссылка: 3
public static StudentContainer Search(String criteria, StudentContainer studentArray, int category)
{
    var searched = new StudentContainer();
    string groupIndex;
    if (studentArray.Students != null)
    {
        switch (category)
        {
            case 0:
            {
                foreach (Student student in studentArray.Students)
                {
                    if (student.GroupIndex.ToString().ToLower().Contains(criteria.ToLower()))
                    {
                        searched.AddStudent(student);
                    }
                }
                break;
            }
            case 1:
            {
                foreach (Student student in studentArray.Students)
                {
                    if (student.Faculty.ToLower().Contains(criteria.ToLower()))
                    {
                        searched.AddStudent(student);
                    }
                }
                break;
            }
            case 2:
            {
                foreach (Student student in studentArray.Students)
                {
                    if (student.Specialization.ToLower().Contains(criteria.ToLower()))
                    {
                        searched.AddStudent(student);
                    }
                }
                break;
            }
        }
        return searched;
    }
    return new StudentContainer();
}
```

Рисунок 7 – функція знаходження підчастини контейнеру студентів за ключем категорії та критерієм

3. Наступним кроком продемонстрували ефективне використання делегатів і забезпечили розрахунок середнього віку і середньої успішності усіх студентів обраної групи, спеціальності, факультету, вузу (рис.8-9).

```

case '7':
    PrintMathematicOptions();
    Console.Write("Make a choice: ");
    intChoice = Io.InputInt();

    StudentContainerProcessing.Del del = null;
    switch (intChoice)
    {
        case 0:
        {
            del = StudentContainerProcessing.CalculateAverageAge;
            break;
        }
        case 1:
        {
            del = StudentContainerProcessing.CalculateAverageAcademicPerformance;
            break;
        }
    }

    PrintOptions();

    Console.Write("Make a choice: ");
    intChoice = Io.InputInt();

    Console.WriteLine("Enter criteria: ");
    strChoice = Console.ReadLine();

    Console.WriteLine(del?.Invoke(StudentContainer.Search(strChoice, pigsty, intChoice)));
    break;

```

Рисунок 8 – робота с делегатами

```

ссылка: 1
public static double CalculateAverageAge(StudentContainer studentArray)
{
    double sum = 0;

    foreach (var student in studentArray.Students)
    {
        sum += student.Age;
    }

    return sum / studentArray.Students.Length;
}

ссылка: 1
public static double CalculateAverageAcademicPerformance(StudentContainer studentArray)
{
    double sum = 0;

    foreach (var student in studentArray.Students)
    {
        sum += student.Performance;
    }

    return sum / studentArray.Students.Length;
}

```

Рисунок 9 – математичні функції

4. Далі була реалізована можливість збереження/відновлення колекції з використанням серіалізації/десеріалізації (рис.10).

```
<?xml version="1.0" encoding="UTF-8"?>
- <ArrayOfStudent xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Student>
    <LastName>Sokolenko</LastName>
    <FirstName>Dmytro</FirstName>
    <Patronymic>Alexandrovich</Patronymic>
    <BirthDate>2001-04-03T00:00:00</BirthDate>
    <EnterDate>2019-09-01T00:00:00</EnterDate>
    <GroupIndex>86</GroupIndex>
    <Faculty>CIT</Faculty>
    <Specialization>123Sys</Specialization>
    <Performance>100</Performance>
  </Student>
  - <Student>
    <LastName>T</LastName>
    <FirstName>T</FirstName>
    <Patronymic>T</Patronymic>
    <BirthDate>2001-02-02T00:00:00</BirthDate>
    <EnterDate>2018-03-03T00:00:00</EnterDate>
    <GroupIndex>67</GroupIndex>
    <Faculty>CIT</Faculty>
    <Specialization>123Sys</Specialization>
    <Performance>80</Performance>
  </Student>
</ArrayOfStudent>
```

Рисунок 10 – серіалізований контейнер

5. Також показаний зв'язок класів у програмі (рис.11).

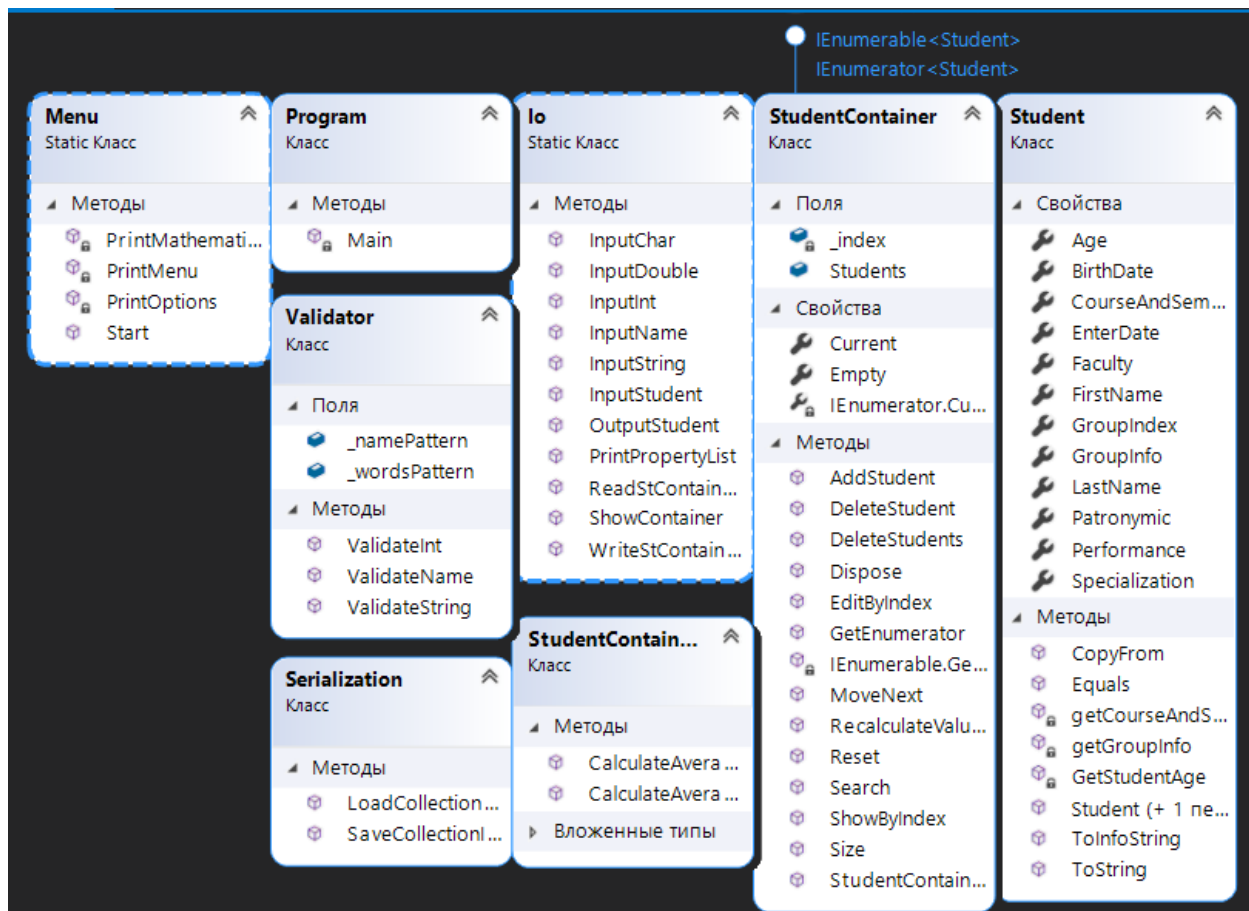


Рисунок 11 – Зв'язок класів у програмі

Висновки: отримали практичний досвід у розробці програм на C# і консольних додатків через додавання у попередню роботу серіалізацію і роботу з делегатами.