

Автор: Абдуллін Олексій

Група: КІТ-119а

Дата: 17.12.2021

Лабораторна робота № 1

Тема: Розробка програм на C#. Консольні програми

Задачі:

1. Введення та збереження анкетних даних особистої справи студента вузу: П.І.Б., дата народження, дата надходження, індекс академ. групи (а, б...), фт, спеціальність, успішність (у%). Забезпечити валідацію даних, що вводяться.
2. Доступ до особистих полів реалізувати за допомогою властивостей класу, забезпечивши необхідні обчислення та перевірку коректності встановлюваних значень. Для спрощеного доступу використовувати властивості, що автоматично реалізуються.
3. Відображення введених даних.
4. Демонстрація роботи з масивами об'єктів.

Опис класів

Student – клас, який відображує студента;

Menu – клас для роботи меню;

IPrinter – інтерфейс для виводу у консоль;

Текст програми

IPrinter

```
using System;

namespace Abdullin01
{
    public interface IPrinter
    {
        void Print(string str);
    }
    public class ConsolePrinter : IPrinter
    {
        public void Print(string str)
        {
            Console.WriteLine(str);
        }
    }
}
```

Student

```
using System;

namespace Abdullin01
{
    class Student
    {
        public string Firstname { get; set; }
        public string Surname { get; set; }
        public string GroupIndex { get; set; }
        public string Faculty { get; set; }
        public int Specialization { get; set; }
        public int AcademicPerformance { get; set; }
        public DateTime DateOfBirth { get; set; }
        public DateTime DateOfEnter { get; set; }
        public IPrinter Printer;

        public override string ToString()
        {
            return "Firstname: " + Firstname + "\nSurname" + Surname +
                "\nDate of birth: " + DateOfBirth.Day + "." + DateOfBirth.Month + "." +
                DateOfBirth.Year +
                "\nDate of enter: " + DateOfEnter.Day + "." + DateOfEnter.Month + "." +
                DateOfEnter.Year +
                "\nIndex of group: " + GroupIndex + "\nFaculty: " + Faculty +
                "\nSpecialization: " + Specialization + "\nAcademic Performance: " +
                AcademicPerformance + "\n";
        }

        public void Print()
        {
            Printer.Print(ToString());
        }

        public Student() : this("Oleksii", "Abdullin", "a", "CIT", 123, 86, new
            DateTime(2002, 5, 31), new DateTime(2019, 8, 12))
        {
        }
    }
}
```

```

        public Student(string firstname, string surname, string groupIndex, string
faculty,
        int specialization, int academicPerformance, DateTime dateOfBirth, DateTime
dateOfEnter)
        {
            this.Firstname = firstname;
            this.Surname = surname;
            this.GroupIndex = groupIndex;
            this.Faculty = faculty;
            this.Specialization = specialization;
            this.AcademicPerformance = academicPerformance;
            this.DateOfBirth = dateOfBirth;
            this.DateOfEnter = dateOfEnter;
            this.Printer = new ConsolePrinter();
        }
    }
}

```

Menu

```

using System;
using System.Collections.Generic;
using System.Text.RegularExpressions;

namespace Abdullin01
{
    public class Menu
    {
        public void MenuStudents()
        {
            var students = new List<Student>();
            students.Add(new Student());
            int option;
            bool inMenu = true;
            while (inMenu)
            {
                Console.WriteLine("Menu options:");
                Console.WriteLine("1. Add");
                Console.WriteLine("2. Remove");
                Console.WriteLine("3. Show all students");
                Console.WriteLine("0. Exit");
                Console.Write("Enter your option: ");

                if (!int.TryParse(Console.ReadLine(), out option))
                {
                    Console.WriteLine("\nError! Invalid datatype.\n");
                    option = -1;
                }

                switch (option)
                {
                    case 1:
                        Regex regex_string = new Regex(@"^[a-z]+$",
RegexOptions.IgnoreCase);

                        string firstname;
                        string surname;
                        string groupIndex;
                        string faculty;
                        int specialization;
                        int academicPerformance;
                        DateTime dateOfBirth;
                        DateTime dateOfEnter;

```

```

Console.Write("Enter firstname of student: ");
firstname = Console.ReadLine();
if (!regex_string.IsMatch(firstname))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

Console.Write("Enter surname of student: ");
surname = Console.ReadLine();
if (!regex_string.IsMatch(surname))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

Console.Write("Enter index of group: ");
groupIndex = Console.ReadLine();
if (!regex_string.IsMatch(groupIndex))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

Console.Write("Enter faculty of student: ");
faculty = Console.ReadLine();
if (!regex_string.IsMatch(faculty))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

Console.Write("Enter specialization of student: ");
if (!int.TryParse(Console.ReadLine(), out specialization))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

Console.Write("Enter academic performance of student: ");
if (!int.TryParse(Console.ReadLine(), out academicPerformance))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

if (academicPerformance > 100 || academicPerformance < 0)
{
    Console.WriteLine("\nError! Invalid value\n");
    break;
}

Console.Write("Enter date of birth of student (e.g. 01/01/2001 or
1.1.2001): ");
if (!DateTime.TryParse(Console.ReadLine(), out dateOfBirth))
{
    Console.WriteLine("\nError! Invalid datatype.\n");
    break;
}

Console.Write("Enter date of enter to university (e.g. 01/01/2001
or 1.1.2001): ");
if (!DateTime.TryParse(Console.ReadLine(), out dateOfEnter))
{

```

specialization,

```
        Console.WriteLine("\nError! Invalid datatype.\n");
        break;
    }

    Student s = new Student(firstname, surname, groupIndex, faculty,
        academicPerformance, dateOfBirth, dateOfEnter);
    students.Add(s);
    break;
case 2:
    int id;
    Console.Write("\nEnter student id: ");
    if (!int.TryParse(Console.ReadLine(), out id))
    {
        Console.Write("\nError! Invalid datatype. \n");
        break;
    }

    bool result = id <= students.Count - 1;
    if (result && id >= 0)
    {
        students.RemoveAt(id);
        Console.Write("\nStudent was deleted successfully.\n");
    }
    else
    {
        Console.Write("\nError! Invalid student id. \n");
    }
    break;
case 3:
    if (students.Count > 0)
    {
        int i = 0;
        foreach (Student stud in students)
        {
            Console.WriteLine("\nStudent ID: " + i);
            stud.Print();
            i++;
        }
    }
    else
    {
        Console.WriteLine("\nList is empty!\n");
    }
    break;
case 0:
    inMenu = false;
    break;
default:
    if (option == -1)
    {
        break;
    }
    Console.WriteLine("\nIncorrect option. Try again.\n");
    break;
}
}
}
}
```

Program

```
namespace Abdullin01
{
    class Program
    {
        static void Main(string[] args)
        {
            var menu = new Menu();
            menu.MenuStudents();
        }
    }
}
```

Результат роботи програми

```
Menu options:
1. Add
2. Remove
3. Show all students
0. Exit
Enter your option: 1
Enter firstname of student: Vologymir
Enter surname of student: Georgovich
Enter index of group: b
Enter faculty of student: CS
Enter specialization of student: 121
Enter academic performance of student: 68
Enter date of birth of student (e.g. 01/01/2001 or 1.1.2001): 12.12.2002
Enter date of enter to university (e.g. 01/01/2001 or 1.1.2001): 02.08.2019
```

```
Enter your option: 3

Student ID: 0
Fristname: Oleksii
SurnameAbdullin
Date of birth: 31.5.2002
Date of enter: 12.8.2019
Index of group: a
Faculty: CIT
Specialization: 123
Academic Performance: 86

Student ID: 1
Fristname: Vologymir
SurnameGeorgovich
Date of birth: 12.12.2002
Date of enter: 2.8.2019
Index of group: b
Faculty: CS
Specialization: 121
Academic Performance: 68
```

```
Enter your option: 2  
Enter student id: 0  
Student was deleted successfully.
```

```
Enter your option: 3  
  
Student ID: 0  
Fristname: Vologymir  
SurnameGeorgovich  
Date of birth: 12.12.2002  
Date of enter: 2.8.2019  
Index of group: b  
Faculty: CS  
Specialization: 121  
Academic Performance: 68
```

Рисунок 1 – Результати роботи програми

Висновок: У результаті виконання лабораторної роботи було створено клас Student, який відображує модель студента. Було створено масив таких об'єктів та виведено його у консоль.