Автор: Сиромятников Марк КІТ-119а

**Лабораторна робота 3**

**Тема**. Обробка колекцій об'єктів. Робота з файлами

Задачі:

1. Збереження введених даних у файлі для подальшого редагування.
2. Відновлення раніше збережених даних із файлу.
3. Пошук та редагування даних студента.
4. Видалення даних особистої справи студента.

Опис класів

Container – власний клас контейнера для реалізації колекції об'єктів;

ContainerEnumerator – клас, який реалізує інтерфейс IEnumerator;

Текст програми

Container.cs

using System;

using syromiatnikov01;

using syromiatnikov02;

namespace syromiatnikov03

{

/// <summary>

/// Class Container

/// class that implements class container

/// for collection of students

/// </summary>

public class ContainerLab03 : Container

{

/// <summary>

/// Конструктор класса наследника

/// </summary>

/// <param name="students"></param>

public ContainerLab03(Student[] students) : base(students)

{

}

/// <summary>

/// Method that allows to edit data of chosen student

/// </summary>

/// <param name="student"></param>

public void EditData(Student student)

{

var pos = -1;

for (var i = 0; i < \_students.Length; i++)

{

if (\_students[i].Equals(student))

{

pos = i;

break;

}

}

if (pos != -1)

{

Console.WriteLine("Enter what field you want to edit:\n1) First name\n2) Last name\n3) Patronymic\n4) Date of birth\n5) Date of admission\n" +

"6) Group\n7) Faculty\n8) Specialty\n9) Academic performance\n");

var option = Console.ReadLine();

try

{

switch (option)

{

case "Name":

\_students[pos].FirstName = Console.ReadLine();

break;

case "Surname":

\_students[pos].LastName = Console.ReadLine();

break;

case "Patronymic":

\_students[pos].Patronymic = Console.ReadLine();

break;

case "Date of birth":

\_students[pos].DateOfBirth = DateTime.Parse(Console.ReadLine());

break;

case "Date of admission":

\_students[pos].DateOfAdmission = DateTime.Parse(Console.ReadLine());

break;

case "Group":

\_students[pos].Group = Console.ReadLine();

break;

case "Faculty":

\_students[pos].Faculty = Console.ReadLine();

break;

case "Specialty":

\_students[pos].Specialty = Console.ReadLine();

break;

case "Academic performance":

\_students[pos].AcademicPerformance = int.Parse(Console.ReadLine());

break;

default:

Console.WriteLine("Invalid option\n");

break;

}

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

}

else

{

Console.WriteLine("There is no such student in collection\n");

}

}

}

}

ContainerEnumerator.cs

using syromiatnikov01;

using System;

using System.Collections;

namespace syromiatnikov03

{

/// <summary>

/// Class ContainerEnum

/// class that implements IEnumerator for student class

/// </summary>

public class ContainerEnumerator : IEnumerator

{

/// <summary>

/// Private fields of a class

/// </summary>

private Student[] \_students;

private int \_position = -1;

/// <summary>

/// Constructor with one parameter

/// </summary>

/// <param name="students"></param>

public ContainerEnumerator(Student[] students)

{

\_students = students;

}

/// <summary>

/// Implemented Current property

/// </summary>

public object Current

{

get

{

try

{

return \_students[\_position];

}

catch (IndexOutOfRangeException)

{

throw new InvalidOperationException();

}

}

}

/// <summary>

/// Implemented MoveNext method

/// </summary>

/// <returns></returns>

public bool MoveNext()

{

\_position++;

return \_position < \_students.Length;

}

/// <summary>

/// Implemented Reset method

/// </summary>

public void Reset()

{

\_position = -1;

}

}

}

Program.cs

using System;

using syromiatnikov01;

namespace syromiatnikov03

{

class Program

{

static void Main(string[] args)

{

var customStudent = new Student("Momot", "Roman", "Evegenievich", DateTime.Parse("10-8-2001"), DateTime.Parse("16-05-2019"), "119a", "CIT", "Computer engineering", 80);

var students = new Student[] { new Student("Bily", "Vadim", "Ivanovich", DateTime.Parse("12-6-2001"), DateTime.Parse("16-05-2019"), "119a", "CIT", "Computer engineering", 100),

new Student("Menshakov", "Dmytro", "Olegovich", DateTime.Parse("16-11-2000"), DateTime.Parse("23-8-2019"), "119a", "CIT", "Computer engineering", 90)};

var list = new ContainerLab03(students);

list.Add(customStudent);

list.EditData(customStudent);

foreach (var item in list)

{

Console.WriteLine(item.ToString());

}

list.Remove(new Student("Menshakov", "Dmytro", "Olegovich", DateTime.Parse("16-11-2000"), DateTime.Parse("23-8-2019"), "119a", "CIT", "Computer engineering", 90));

foreach (var item in list)

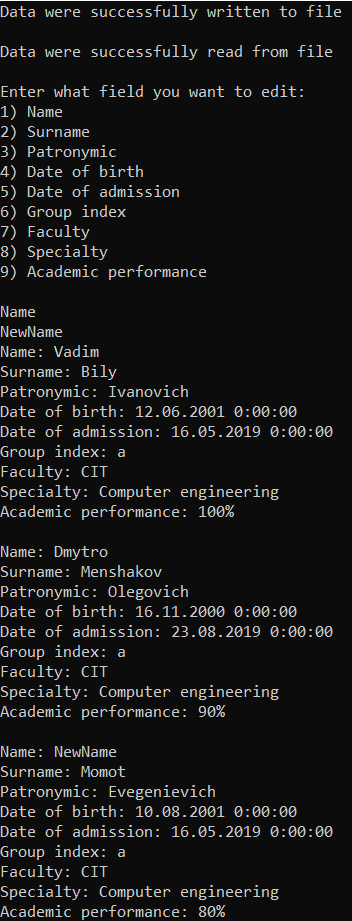
{

Console.WriteLine(item.ToString());

}

var stud = list.Find(customStudent);

Console.ReadLine();



Результати роботи програми

**Висновок:** у результаті виконання лабораторної роботи було модифіковано клас Container для збереження поточних даних у файл у форматі Json та їх відновлення, було додано методи пошуку редагування та видалення особистих даних студента.