Автор: Сиромятников Марк КІТ-119а

**Лабораторна робота 5**

**Тема**. Серіалізація в С #. Делегати

Задачі:

1. Забезпечити відображення у вигляді таблиці даних особових справ усіх студентів обраної групи, спеціальності, ф-ту, вузу.
2. Реалізувати групове та видалення особових справ студентів (за групою, спеціальністю, ф-ту, вузу чи власному критерію).
3. Продемонструвати ефективне використання делегатів та забезпечити: розрахунок середнього віку всіх студентів обраної групи, спеціальності, ф-ту, вузу; розрахунок середньої успішності всіх студентів обраної групи, спеціальності, ф-та, вузу.

Опис класів

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

ContainerEnumerator – клас, який реалізує інтерфейс IEnumerator;

StudentExtension – клас, який виконує обробку даних студента;

Текст програми

Container.cs

using System;

using System.Collections;

using syromiatnikov01;

using syromiatnikov03;

using syromiatnikov05.Comparators;

namespace syromiatnikov05

{

/// <summary>

/// Class Container

/// class that implements class container

/// for collection of students

/// </summary>

public class ContainerLab05 : ContainerLab03

{

/// <summary>

///

/// </summary>

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

public ContainerLab05(Student[] students) : base(students)

{

}

/// <summary>

/// Method that clears the collection

/// </summary>

public void Clear()

{

\_students = null;

}

/// <summary>

/// Method that removes student by chosen criteria

/// </summary>

/// <returns>True if student was removed otherwise false</returns>

public bool RemoveByCriteria()

{

IComparer comparator = null;

Console.WriteLine("Enter criteria of the deletion:");

Console.WriteLine("1) group");

Console.WriteLine("2) specialty");

Console.WriteLine("3) faculty\n");

var input = Console.ReadLine();

switch (input)

{

case "group":

Console.WriteLine("Write group:");

input = Console.ReadLine();

comparator = new CompareGroup();

break;

case "specialty":

Console.WriteLine("Write specialty:");

input = Console.ReadLine();

comparator = new CompareSpecialty();

break;

case "faculty":

Console.WriteLine("Write faculty:");

input = Console.ReadLine();

comparator = new CompareFaculty();

break;

default:

input = string.Empty;

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

break;

}

if (!string.IsNullOrEmpty(input))

{

var previousSize = \_students.Length;

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

{

if (comparator.Compare(\_students[i], input) == 0)

{

Remove(\_students[i]);

i--;

}

}

if (previousSize != \_students.Length)

{

return true;

}

}

return false;

}

}

}

StudentExtension.cs

using System;

using System.Collections;

using syromiatnikov01;

using syromiatnikov05.Comparators;

namespace menshakov05

{

public static class StudentExtension

{

delegate int IsEqual(Student[] student);

/// <summary>

/// Method that counts chosen average value of a given collection

/// </summary>

/// <returns>Returns average value of a chosen field</returns>

public static int CountAverage(this Student[] \_students)

{

IComparer comparator = null;

IsEqual func = null;

Console.WriteLine("Count avg age or academic performance:");

Console.WriteLine("1) Age");

Console.WriteLine("2) Performance");

var input = Console.ReadLine();

if (input == "Age")

{

func = CountAvgAge;

}

else if (input == "Performance")

{

func = CountAvgPerformance;

}

else

{

Console.WriteLine("Invalid option");

return -1;

}

Console.WriteLine("Enter criteria of the counting:");

Console.WriteLine("1) group index");

Console.WriteLine("2) specialty");

Console.WriteLine("3) faculty\n");

input = Console.ReadLine();

switch (input)

{

case "group index":

Console.WriteLine("Write group index:");

input = Console.ReadLine();

comparator = new CompareGroup();

break;

case "specialty":

Console.WriteLine("Write specialty:");

input = Console.ReadLine();

comparator = new CompareSpecialty();

break;

case "faculty":

Console.WriteLine("Write faculty:");

input = Console.ReadLine();

comparator = new CompareFaculty();

break;

default:

input = string.Empty;

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

break;

}

if (!string.IsNullOrEmpty(input))

{

var size = 0;

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

{

if (comparator.Compare(\_students[i], input) == 0)

{

size++;

}

}

var students = new Student[size];

size = 0;

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

{

if (comparator.Compare(\_students[i], input) == 0)

{

students[size] = \_students[i];

size++;

}

}

return func(students);

}

return -1;

}

/// <summary>

/// Method that counts average students` age of a given collection

/// </summary>

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

/// <returns>Returns average value of an age field</returns>

private static int CountAvgAge(Student[] students)

{

var count = 0;

foreach (var student in students)

{

count += DateTime.Now.Year - student.DateOfBirth.Year;

}

return count / students.Length;

}

/// <summary>

/// Method that counts average students` performance of a given collection

/// </summary>

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

/// <returns>Returns average value of an performance field</returns>

private static int CountAvgPerformance(Student[] students)

{

var count = 0;

foreach (var student in students)

{

count += student.AcademicPerformance;

}

return count / students.Length;

}

}

}

CompareFaculty.cs

using syromiatnikov01;

using System.Collections;

namespace syromiatnikov05.Comparators

{

/// <summary>

/// Class CompareFaculty

/// class that implements IComparer interface

/// for the faculty of a student

/// </summary>

public class CompareFaculty : IComparer

{

public int Compare(object x, object y)

{

var student = (Student)x;

var data = (string)y;

return student.Faculty.CompareTo(data);

}

}

}

CompareGroupIndex.cs}

using syromiatnikov01;

using System;

using System.Collections;

namespace syromiatnikov05.Comparators

{

/// <summary>

/// Class CompareGroups

/// class that implements IComparer interface

/// for the group of a student

/// </summary>

public class CompareGroup : IComparer

{

public int Compare(object x, object y)

{

var student = (Student)x;

var criteria = (string)y;

return student.Group.CompareTo(criteria);

}

}

}

CompareSpecialty.cs}

using syromiatnikov01;

using System.Collections;

namespace syromiatnikov05.Comparators

{

/// <summary>

/// Class CompareSpecialty

/// class that implements IComparer interface

/// for the specialty of a student

/// </summary>

public class CompareSpecialty : IComparer

{

public int Compare(object x, object y)

{

var student = (Student)x;

var data = (string)y;

return student.Specialty.CompareTo(data);

}

}

}

Program.cs

using System;

using syromiatnikov01;

namespace syromiatnikov05

{

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 ContainerLab05(students);

list.Add(customStudent);

list.RemoveByCriteria();

/\*list.WriteToFile();

list.ReadFromFile();\*/

/\*list.ShowData(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"), 'a', "CIT", "Computer engineering", 90));

foreach (var item in list)

{

Console.WriteLine(item.ToString());

}

var stud = list.Find(customStudent);\*/

list.RemoveByCriteria();

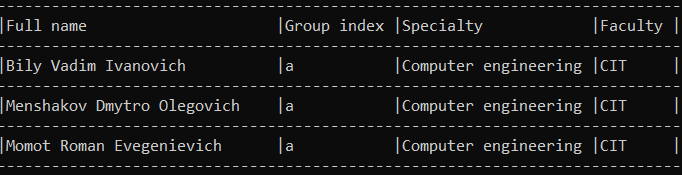
list.Clear();

Console.ReadLine();

}

}

}



Результати роботи програми

**Висновок**: у результаті виконання лабораторної роботи було забезпечено відображення у вигляді таблиці даних особових справ усіх студентів обраної групи, спеціальності, ф-ту, вузу; реалізовано групове та видалення особових справ студентів та продемонстровано ефективне використання делегатів.