

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2017 г.

"__" _____ 2017 г.

Отчет по лабораторной работе № 6 по курсу Базовые компоненты интернет-технологий

9

(количество листов)

Студент группы ИУ5-33
Коционова Анна

Москва МГТУ 2017

СОДЕРЖАНИЕ

1. Описание задания лабораторной работы
2. Текст программы
3. Диаграмма классов
4. Результат работы программы

1. Описание задания лабораторной работы

Лабораторная работа №6

Часть 1. Разработать программу, использующую делегаты.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Определите делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа.
3. Напишите метод, соответствующий данному делегату.
4. Напишите метод, принимающий разработанный Вами делегат, в качестве одного из входным параметров. Осуществите вызов метода, передавая в качестве параметра-делегата:
 - метод, разработанный в пункте 3;
 - лямбда-выражение.
5. Повторите пункт 4, используя вместо разработанного Вами делегата, обобщенный делегат `Func< >` или `Action< >`, соответствующий сигнатуре разработанного Вами делегата.

Часть 2. Разработать программу, реализующую работу с рефлексией.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создайте класс, содержащий конструкторы, свойства, методы.
3. С использованием рефлексии выведите информацию о конструкторах, свойствах, методах.
4. Создайте класс атрибута (унаследован от класса `System.Attribute`).
5. Назначьте атрибут некоторым свойствам классам. Выведите только те свойства, которым назначен атрибут.
6. Вызовите один из методов класса с использованием рефлексии.

2. Текст программы

Main 6_1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab7
{
    class Program
    {
        /// <summary>
        /// класс "Сотрудник"
        /// </summary>
        public class Staff
        {
            /// <summary>
            /// ID сотрудника
            /// </summary>
            public int id_staff;
            /// <summary>
            /// фамилия сотрудника
            /// </summary>
            public string staff_surname;
            /// <summary>
            /// записи об отделе
            /// </summary>
            public int id_department;

            public Staff(int id_s, string surname, int id_d)
            {
                id_staff = id_s;
                staff_surname = surname;
                id_department = id_d;
            }

            public override string ToString()
            {
                return "id staff=" + id_staff.ToString() + " |surname=" +
staff_surname.ToString() + " |id department=" + id_department.ToString();
            }
        }

        public class Department
        {
            /// <summary>
            /// ID отдела
            /// </summary>
            public int id_department;
            /// <summary>
            /// название отдела
            /// </summary>
            public string name_department;
            public Department(int id_d, string name)
            {
                id_department = id_d;
                name_department = name;
            }
        }
    }
}
```

```

        public override string ToString()
        {
            return "id department=" + id_department.ToString() + " | name
department=" + name_department.ToString() + " | id department=";
        }
    }
    public class Staffes_of_department
    {
        /// <summary>
        /// ID сотрудника
        /// </summary>
        public int id_staff;
        /// <summary>
        /// ID отдела
        /// </summary>
        public int id_department;
        /// <summary>
        /// Конструктор
        /// </summary>
        public Staffes_of_department(int i, int i_d)
        {
            id_staff = i;
            id_department = i_d;
        }
        /// <summary>
        /// Приведение к строке
        /// </summary>
        public override string ToString()
        {
            return "id = " + id_staff.ToString() + " | id_department = " +
id_department.ToString();
        }
    }
    static List<Staff> s = new List<Staff>()
    {
        new Staff(1, "Kotsionova", 12),
        new Staff(2, "Zhelankina", 14),
        new Staff(3, "Fedorova", 12),
        new Staff(4, "Nurlyeva", 11),
        new Staff(5, "Alexxxevna", 13),
    };

    static List<Department> d = new List<Department>()
    {
        new Department(11, "music"),
        new Department(12, "coding"),
        new Department(13, "sleep"),
        new Department(14, "dance"),
    };

    static List<Staffes_of_department> e_d = new List<Staffes_of_department>()
    {
        new Staffes_of_department(1, 12),
        new Staffes_of_department(2, 14),
        new Staffes_of_department(3, 12),
        new Staffes_of_department(4, 11),
        new Staffes_of_department(5, 13),
        new Staffes_of_department(6, 11),
        new Staffes_of_department(7, 13),
        new Staffes_of_department(8, 12),
    };

    static void Main(string[] args)

```

```

{
    Console.WriteLine("Список всех сотрудников и отделов, отсортированный по
отделам.");
    var q1 = from x in s
              orderby x.id_department descending, x.id_staff ascending
              select x;
    foreach (var x in q1)
        Console.WriteLine(x);

    Console.WriteLine("\nСписок всех сотрудников, у которых фамилия начинается с
буквы «А».");
    var q2 = from x in s
              where x.staff_surname[0] is 'A'
              orderby x.staff_surname ascending, x.id_staff descending
              select x;
    foreach (var x in q2)
        Console.WriteLine(x);

    Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом
отделе.");
    var q3 = from x in d
              join y in s on x.id_department equals y.id_department into temp
              from t in temp
              select new { v1 = x.name_department, v2 = t.id_department, cnt =
temp.Count() };
    q3 = q3.Distinct();
    foreach (var x in q3) Console.WriteLine(x);

    Console.WriteLine("\nСписок отделов, в которых у всех сотрудников фамилия
начинается с буквы «А».");
    var q4_1 = from x in s
                join y in q2 on x.id_department equals y.id_department into temp
                from t in temp
                select new { v1 = x.id_department, cnt = temp.Count() };
    q4_1 = q4_1.Distinct();
    var q4 = from x in q3
              from y in q4_1
              where (x.cnt == y.cnt) && (x.v2 == y.v1)
              select new { v1 = x.v1 };
    q4 = q4.Distinct();
    foreach (var x in q4)
        Console.WriteLine(x);

    Console.WriteLine("\nСписок отделов, в которых хотя бы у одного сотрудника
фамилия начинается с буквы «А».");
    var q5_1 = from x in s
                where x.staff_surname[0] is 'A'
                select new { v1 = x.id_department };
    q5_1 = q5_1.Distinct();
    var q5 = from x in d
              from y in q5_1
              where x.id_department == y.v1
              select new { v1 = x.name_department };
    q5 = q5.Distinct();
    foreach (var x in q5)
        Console.WriteLine(x);
    //------

    Console.WriteLine("\nСписок всех отделов и список сотрудников в каждом
отделе.");
    var q6_1 = from x in s
                join l in e_d on x.id_staff equals l.id_staff into temp
                from t1 in temp
                join y in d on t1.id_department equals y.id_department into temp2
                from t2 in temp2

```

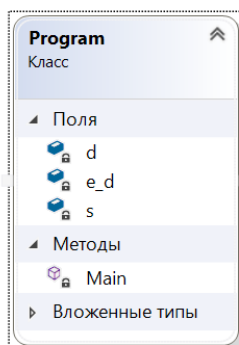
```

        select new { id = x.id_department, name = t2.name_department };
q6_1 = q6_1.Distinct();
foreach (var x in q6_1)
    Console.WriteLine(x);
var q6_2 = from x in s
           join l in e_d on x.id_staff equals l.id_staff into temp
           from t1 in temp
           join y in s on t1.id_staff equals y.id_staff into temp2
           from t2 in temp2
           select new { id = x.id_staff, surname = t2.staff_surname };
q6_2 = q6_2.Distinct();
foreach (var x in q6_2)
    Console.WriteLine(x);

Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом
отделе. ");
var q7_1 = from x in e_d
           join y in s on x.id_department equals y.id_department into temp
           from t in temp
           select new { number = temp.Count(), id = t.id_department };
q7_1 = q7_1.Distinct();
var q7_2 = from x in s
           join ed in e_d on x.id_staff equals ed.id_staff into temp
           from t1 in temp
           join y in d on t1.id_department equals y.id_department into temp2
           from t2 in temp2
           select new { name = t2.name_department, id = t2.id_department };
q7_2 = q7_2.Distinct();
var q7 = from x in q7_1
         from y in q7_2
         where x.id == y.id
         select new { name = y.name, number = x.number };
q7 = q7.Distinct();
foreach (var x in q7)
    Console.WriteLine(x);
Console.ReadLine();
    }
}
}

```

3. Диаграмма классов



4. Результат выполнения программы

```
Список всех сотрудников и отделов, отсортированный по отделам.
id staff=2 |surname=Zhelankina |id department=14
id staff=5 |surname=Alexxxevna |id department=13
id staff=1 |surname=Kotsionova |id department=12
id staff=3 |surname=Fedorova |id department=12
id staff=4 |surname=Nurlyeva |id department=11

Список всех сотрудников, у которых фамилия начинается с буквы <А>.
id staff=5 |surname=Alexxxevna |id department=13

Список всех отделов и количество сотрудников в каждом отделе.
{ v1 = music, v2 = 11, cnt = 1 }
{ v1 = coding, v2 = 12, cnt = 2 }
{ v1 = sleep, v2 = 13, cnt = 1 }
{ v1 = dance, v2 = 14, cnt = 1 }

Список отделов, в которых у всех сотрудников фамилия начинается с буквы <А>.
{ v1 = sleep }

Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы <А>.
{ v1 = sleep }

Список всех отделов и список сотрудников в каждом отделе.
{ id = 12, name = coding }
{ id = 14, name = dance }
{ id = 11, name = music }
{ id = 13, name = sleep }
{ id = 1, surname = Kotsionova }
{ id = 2, surname = Zhelankina }
{ id = 3, surname = Fedorova }
{ id = 4, surname = Nurlyeva }
{ id = 5, surname = Alexxxevna }

Список всех отделов и количество сотрудников в каждом отделе.
{ name = coding, number = 2 }
{ name = dance, number = 1 }
{ name = music, number = 1 }
{ name = sleep, number = 1 }
```