



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Московский государственный технический университет
им. Н.Э. Баумана
(МГТУ им. Н.Э. Баумана)

Радиотехнический факультет (РТ)

Лабораторная работа № 7

По дисциплине: «Базовые компоненты интернет технологий»

Тема: «Разработать программу, реализующую работу с LINQ to Objects.»

Выполнил: Наврузов Э.Р.

студент группы РТ5-31

Проверил: Гапанюк Ю.Е.,

Преподаватель каф. ИУ5

г. Москва 2017 г.

Описание задания лабораторной работы:

Часть 1. Разработать программу, использующую делегаты

1. Программа должна быть разработана в виде консольного приложения на языке C#.

2. Создайте класс «Сотрудник», содержащий поля:

- ID записи о сотруднике;
- Фамилия сотрудника;
- ID записи об отделе.

3. Создайте класс «Отдел», содержащий поля:

- ID записи об отделе;
- Наименование отдела.

4. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением один-ко-многим разработайте следующие запросы:

- Выведите список всех сотрудников и отделов, отсортированный по отделам.
- Выведите список всех сотрудников, у которых фамилия начинается с буквы «П».
- Выведите список всех отделов и количество сотрудников в каждом отделе.
- Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы «П».
- Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «П».

5. Создайте класс «Сотрудники отдела», содержащий поля:

- ID записи о сотруднике;
- ID записи об отделе.

6. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением много-ко-многим с использованием класса «Сотрудники отдела» разработайте следующие запросы:

- Выведите список всех отделов и список сотрудников в каждом отделе.
- Выведите список всех отделов и количество сотрудников в каждом отделе.

Program.cs:

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

namespace ConsoleApp38
{
    class Program
    {
        public class Worker
        {
            public int id;
            public string name;
            public int unit_id;

            public Worker(int i, string n, int u_id)
            {
                this.id = i;
                this.name = n;
                this.unit_id = u_id;
            }

            public override string ToString()
            {
                return "(id=" + this.id.ToString() + "; name=" + this.name + "; place
id=" + this.unit_id + ")";
            }
        }
        public class Unit
        {
            public int id;
            public string name;

            public Unit(int i, string n)
            {
                this.id = i;
                this.name = n;
            }

            public override string ToString()
            {
                return "(id=" + this.id.ToString() + "; name=" + this.name + ")";
            }
        }
        public class WorkersOfUnit
        {
            public int worker;
            public int unit;

            public WorkersOfUnit(int i1, int i2)
            {
                this.worker = i1;
                this.unit = i2;
            }
        }
        static List<Worker> worker = new List<Worker>(){
            new Worker(1, "Наврузов", 1),
            new Worker(2, "Лузеров", 6),
            new Worker(3, "Алексеев", 3),
            new Worker(4, "Саврасов", 2),
        }
    }
}
```

```

new Worker(5, "Арнаут", 2),
new Worker(6, "Бродский", 4),
new Worker(7, "Астанов", 1),
new Worker(8, "Артемов", 5),
new Worker(9, "Монокарпова", 3),
new Worker(10, "Поликарпова", 4),
new Worker(11, "Горшенев", 2),
new Worker(12, "Путин", 4),
new Worker(13, "Навальный", 6),
new Worker(14, "Нурмагамедов", 5),
new Worker(15, "Макгрегор", 3),
new Worker(16, "Исмаилов", 5),
new Worker(17, "Лазарев", 6),
new Worker(18, "Фамильев", 1),
};
static List<Unit> unit = new List<Unit>(){
    new Unit(1, "Отдел Деканат"),
    new Unit(2, "Отдел Студенты"),
    new Unit(3, "Отдел Завхозы"),
    new Unit(4, "Отдел Сисадминов"),
    new Unit(5, "Отдел Директоров"),
    new Unit(6, "Бухгалтерия"),
};
static List<WorkersOfUnit> wou = new List<WorkersOfUnit>()
{
    new WorkersOfUnit(1,1),
    new WorkersOfUnit(1,2),
    new WorkersOfUnit(2,3),
    new WorkersOfUnit(2,4),
    new WorkersOfUnit(3,5),
    new WorkersOfUnit(3,6),
    new WorkersOfUnit(18,6),
    new WorkersOfUnit(18,5),
    new WorkersOfUnit(17,4),
    new WorkersOfUnit(17,3),
    new WorkersOfUnit(16,2),
    new WorkersOfUnit(16,1),
};

static void Main(string[] args)
{
    Console.Title= "База сотрудников";
    #region Список всех сотрудников и отделов, отсортированный по отделам
    Console.WriteLine("Список всех сотрудников и отделов, отсортированный по
отделам:\n");
    var q1 = from x in unit
              join y in worker on x.id equals y.unit_id into temp
              select new { id = x.id, name = x.name, d2Group = temp };
    foreach (var x in q1)
    {
        Console.WriteLine(x.id + " " + x.name);
        foreach (var y in x.d2Group)
            Console.WriteLine("    " + y);
    }
    #endregion
    #region Список всех сотрудников, у которых фамилия начинается на букву П
    Regex regex = new Regex("П");
    Console.WriteLine("\nСписок всех сотрудников, у которых фамилия начинается на
букву П \n");
    var q2 = from x in worker
              where regex.IsMatch(x.name)
              select x;
    foreach (var x in q2) Console.WriteLine(x);
    #endregion
    #region Список всех отделов и количество сотрудников в каждом отделе.

```

```

        Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом
отделе. \n");
        var q3 = from x in unit
                  select new { uid = x.id, uname = x.name, ucount = worker.Count(z =>
z.unit_id == x.id) };
        foreach (var x in q3)
            Console.WriteLine(x);
        #endregion
        #region Список отделов, в которых у всех сотрудников фамилия начинается с
буквы «П»
        Console.WriteLine("\nСписок отделов, в которых у всех сотрудников фамилия
начинается с буквы «П» \n");
        var q4 = worker.GroupBy(x => x.unit_id);
        foreach (var x in q4.Where(z => z.All(p => regex.IsMatch(p.name))))
            Console.WriteLine("{0}", x.Key);
        #endregion
        #region Список отделов, в которых хотя бы у одного сотрудника фамилия
начинается с буквы 'П'
        Console.WriteLine("Список отделов, в которых хотя бы у одного сотрудника
фамилия начинается с буквы 'П'");
        foreach (var x in q4.Where(z => z.Any(p => regex.IsMatch(p.name))))
            Console.WriteLine("{0}", x.Key);
        #endregion
        #region Список всех отделов и список сотрудников в каждом отделе
        Console.WriteLine("\nСписок всех отделов и список сотрудников в каждом
отделе. \n");
        var wou1 = from x in worker
                    join l in wou on x.id equals l.worker into temp
                    from t1 in temp
                    join y in unit on t1.unit equals y.id into temp2
                    from t2 in temp2
                    select new { id1 = x.id, id2 = t2.id };
        foreach (var x in wou1) Console.WriteLine(x);
        #endregion

        Console.WriteLine("\nPress any key to exit");
        Console.ReadKey();
    }
}

```

Пример консольного вывода:

```

База сотрудников
Список всех сотрудников и отделов, отсортированный по отделам:
1 Отдел Деканат
(id=1; name=Наврузов; place id=1)
(id=7; name=Астанов; place id=1)
(id=18; name=Фамилев; place id=1)
2 Отдел Студенты
(id=4; name=Саврасов; place id=2)
(id=5; name=Арнаут; place id=2)
(id=11; name=Горшенев; place id=2)
3 Отдел Завхозы
(id=3; name=Алексеев; place id=3)
(id=9; name=Монокарпова; place id=3)
(id=15; name=Макгрегор; place id=3)
4 Отдел Сисадминов
(id=6; name=Бродский; place id=4)
(id=10; name=Поликарпова; place id=4)
(id=12; name=Путин; place id=4)
5 Отдел Директоров
(id=8; name=Артемов; place id=5)
(id=14; name=Нурмагамедов; place id=5)
(id=16; name=Исмаилов; place id=5)
6 Бухгалтерия
(id=2; name=Лузеров; place id=6)
(id=13; name=Навальный; place id=6)
(id=17; name=Лазарев; place id=6)

Список всех сотрудников, у которых фамилия начинается на букву П
(id=10; name=Поликарпова; place id=4)
(id=12; name=Путин; place id=4)

Список всех отделов и количество сотрудников в каждом отделе.
{ uid = 1, upame = Отдел Деканат, ucount = 3 }
{ uid = 2, upame = Отдел Студенты, ucount = 3 }
{ uid = 3, upame = Отдел Завхозы, ucount = 3 }
{ uid = 4, upame = Отдел Сисадминов, ucount = 3 }
{ uid = 5, upame = Отдел Директоров, ucount = 3 }
{ uid = 6, upame = Бухгалтерия, ucount = 3 }

Список отделов, в которых у всех сотрудников фамилия начинается с буквы <П>
Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы 'П'
4

Список всех отделов и список сотрудников в каждом отделе.
{ id1 = 1, id2 = 1 }
{ id1 = 1, id2 = 2 }
{ id1 = 2, id2 = 3 }
{ id1 = 2, id2 = 4 }
{ id1 = 3, id2 = 5 }
{ id1 = 3, id2 = 6 }
{ id1 = 16, id2 = 2 }
{ id1 = 16, id2 = 1 }
{ id1 = 17, id2 = 4 }
{ id1 = 17, id2 = 3 }
{ id1 = 18, id2 = 6 }
{ id1 = 18, id2 = 5 }

Press any key to exit

```

Диаграмма классов:

