**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: Астанов Э.М,

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.

Текст программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace linqlab

{

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;

}

/// <summary>

/// Приведение к строке

/// </summary>

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<Unit> unit = new List<Unit>(){

new Unit(1,"Отдел продаж"),

new Unit(2,"Отдел кадров"),

new Unit(3,"Отдел маркетинга"),

new Unit(4,"Отдел IT"),

new Unit(5,"Отдел административно-хозяйственный"),

new Unit(6,"Бухгалтерия"),

};

static List<Worker> worker=new List<Worker>(){

new Worker(1,"Продавайкин",1),

new Worker(2,"Продавцов",1),

new Worker(3,"Наваров",1),

new Worker(4,"Увольняева",2),

new Worker(5,"Набираева",2),

new Worker(6,"Разбираев",2),

new Worker(7,"Рекламов",3),

new Worker(8,"Креативов",3),

new Worker(9,"Пропихаева",3),

new Worker(10,"Сисадминов",4),

new Worker(11,"Эникеев",4),

new Worker(12,"Тыжпрограммистов",4),

new Worker(13,"Прихватова",5),

new Worker(14,"Прихватов",5),

new Worker(15,"Петров",5),

new Worker(16,"Считалова",6),

new Worker(17,"Уклоняева",6),

new Worker(18,"Офшоров",6),

};

static List<WorkersOfUnit> wou = new List<WorkersOfUnit>()

{

new WorkersOfUnit(1,1),

new WorkersOfUnit(1,2),

new WorkersOfUnit(12,4),

new WorkersOfUnit(12,1),

new WorkersOfUnit(2,2),

new WorkersOfUnit(2,4),

new WorkersOfUnit(14,6),

new WorkersOfUnit(14,5)

};

static void Main(string[] args)

{

Console.WriteLine(" Выведите 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);

}

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

Regex regex = new Regex("П");

Console.WriteLine("\n Вывод список 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);

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine("\n Выведите список 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);

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine("\nВыведите список 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);

Console.WriteLine("Отделы, 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);

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine("\nВыведите список 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);

Console.WriteLine("\n Выведите список всех отделов и количество сотрудников в каждом отделе. \n");

var wou2 = from x in worker

let temp1 = from l in wou where l.worker == x.id select l

from t1 in temp1

let temp2 = from y in unit

where y.id == t1.unit && y.id ==x.id

select y

where temp2.Count() > 0

select x;

foreach (var x in wou2) Console.WriteLine(x);

Console.ReadLine();

}

}

}

