



**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Э. БАУМАНА**

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления (ИУ5)»

Дисциплина: «Базовые компоненты Интернет-технологий»

Отчет по лабораторной работе № 7

Выполнила: Журавлева Полина Валерьевна

Группа: ИУ5-31Б

Преподаватель: Гапанюк Юрий Евгеньевич

Дата: 23.12.2018

Подпись:

Описание задания:

Разработать программу, реализующую работу с LINQ to Objects. В качестве примера используйте проект «SimpleLINQ» из примера «Введение в LINQ».

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 «А».
 - Выведите список всех отделов и количество сотрудников в каждом отделе.
 - Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы «А».
 - Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А».
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.Globalization;

namespace Лаб7
{
    public class Worker
    {
        public int ID;
        public string surname;
        public int ID_D;
        public Worker(int i, string s, int d)
        {
            this.ID = i;
            this.surname = s;
            this.ID_D = d;
        }
    }

    public class Department
    {
        public int ID_D;
        public string name;
        public Department(int id, string n)
        {
            this.ID_D = id;
            this.name = n;
        }
    }

    public class Workers
    {
        public int IDs;
        public int ID_D;
        public Workers(int i, int d)
        {
            this.IDs = i;
            this.ID_D = d;
        }
    }
}
```

```

}
class Program
{
    static void Main(string[] args)
    {
        //заполнение данных
        List<Worker> work = new List<Worker>()
        { new
            Worker(1,"Алёшкин",1)
            , new
            Worker(2,"Артемов",1)
            ), new
            Worker(3,"Иванов",2),
            new Worker(4,"Петров",3),
            new Worker(5,"Артуров",2)
        };
        List<Department> div = new List<Department>()
        {
            new Department(1,"Отдел1"),
            new Department(2,"Отдел2"),
            new Department(3,"Отдел3"),
        };
        var q = work.Join(div, x => x.ID_D, y => y.ID_D, //Join -
соединяет две коллекции по определенному признаку
(x, y) => new { IDD = y.ID_D, named = y.name, ID_W = x.ID,
Sur =
x.surname });
        // Выведите список всех сотрудников и отделов,
отсортированный по отделам Console.WriteLine("Список
всех сотрудников и отделов, отсортированный по
отделам:");
        foreach (var z in q.OrderBy(t => t.named)) //OrderBy -
упорядочивает элементы по возрастанию
        {
            Console.WriteLine("{0} {1} {2} {3}", z.IDD, z.named, z.ID_W,
z.Sur);
        }

        //Выведите список всех сотрудников, у которых фамилия
начинается с буквы «А». Console.WriteLine("Список всех
сотрудников, у которых фамилия начинается с
буквы «А»:");
    }
}

```

```

        foreach (var z in q.Where(x => x.Sur.StartsWith("А",
true, CultureInfo.CurrentCulture))) //Where - определяет фильтр
выборки; StartsWith - определяет, совпадает ли

//начало этого экземпляра строки с заданной строкой
    {
        Console.WriteLine("{0} {1} {2} {3}", z.IDD, z.named, z.ID_W,
z.Sur);
    }

//Выведите список всех отделов и количество
сотрудников в каждом отделе. Console.WriteLine("Список
всех отделов и количество сотрудников в каждом
отделе:");
    foreach (var z in work.Join(div, x => x.ID_D, y => y.ID_D,
(x, y) => new { namediv = y.name, IDD = y.ID_D, count =
work.Where(w => w.ID_D == y.ID_D).Count() }).Distinct()) //Count -
подсчитывает количество элементов коллекции,

//которые удовлетворяют определенному условию;

//Distinct - удаляет дублирующиеся элементы из коллекции
    {
        Console.WriteLine("{0} {1} {2}", z.IDD, z.namediv, z.count);
    }

//Выведите список отделов, в которых у всех сотрудников
фамилия начинается с
буквы «А»
    Console.WriteLine("Список отделов, в которых у всех
сотрудников фамилия начинается с буквы «А»:");
    foreach (var z in q.GroupBy(x => x.IDD) //GroupBy - группирует
элементы по
ключу
        .Where(p => p.All(f => f.Sur.StartsWith("А", true,
CultureInfo.CurrentCulture)))) //All - определяет, все ли
элементы коллекции удовлетворяют определенному
условию
    {
        Console.WriteLine("{0}", z.Key);
    }

//Выведите список отделов, в которых хотя бы у одного
сотрудника фамилия начинается с буквы «А»

```

```

        Console.WriteLine("Список отделов, в которых хотя бы у
одного сотрудника фамилия начинается с буквы «А»:");
        foreach (var z in q.GroupBy(x => x.IDD)
            .Where(p => p.Any(f => f.Sur.StartsWith("А", true,
CultureInfo.CurrentCulture)))) //Any - определяет, удовлетворяет хотя бы
один элемент
коллекции определенному условию
        {
            Console.WriteLine("{0}", z.Key);
        }
        List<Workers> works = new List<Workers>()
        {
            new
            Workers(101,
                1), new
            Workers(102,
                1), new
            Workers(103,
                2),
            new Workers(104,3),
            new Workers(104,2)
        };
        //Выведите список всех отделов и список сотрудников
в каждом отделе var qq = works.Join(div, x => x.ID_D, y
=> y.ID_D,
        (x, y) => new { IDDs = y.ID_D, names = y.name, ID_Ws =
x.IDs }); Console.WriteLine("Список всех отделов и
список сотрудников в каждом
отделе:");
        foreach (var zz in qq)
        {
            Console.WriteLine(zz.IDDs + " (" + zz.names + ") " + zz.ID_Ws);
        }

        //Выведите список всех отделов и количество
сотрудников в каждом отделе Console.WriteLine("Список
всех отделов и количество сотрудников в каждом
отделе:");
        foreach (var zz in works.Join(div, x => x.ID_D, y => y.ID_D,
            (x, y) => new { namediv = y.name, IDD = y.ID_D, count =
works.Where(w => w.ID_D == y.ID_D).Count() }).Distinct())
        {
            Console.WriteLine("{0} {1} {2}", zz.IDD, zz.namediv, zz.count);
        }

```

```

        Console.ReadKey();
    }
}

```

Диаграмма классов:



Пример выполнения программы:

```

file:///C:/Users/Домашний/documents/visual studio 2015/Projects/Ла67/Ла67...
Список всех сотрудников, у которых фамилия начинается с буквы <А>:
1 Отдел1 1 Алёшкин
1 Отдел1 2 Артемьев
2 Отдел2 5 Артуров
Список всех отделов и количество сотрудников в каждом отделе:
1 Отдел1 2
2 Отдел2 2
3 Отдел3 1
Список отделов, в которых у всех сотрудников фамилия начинается с буквы <А>:
1
Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы
<А>:
1
2
Список всех отделов и список сотрудников в каждом отделе:
1 <Отдел1> 101
1 <Отдел1> 102
2 <Отдел2> 103
3 <Отдел3> 104
2 <Отдел2> 104
Список всех отделов и количество сотрудников в каждом отделе:
1 Отдел1 2
2 Отдел2 2
3 Отдел3 1

```