

**Лабораторная работа № 7 по курсу**  
**“Базовые компоненты интернет-технологий”**

*Алексеев А.В.*  
*РТ5-31*  
*МГТУ им. Баумана*

### **Описание задания лабораторной работы.**

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.

### Код программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
namespace Linq
{
    class Program
    {
        class Worker
        {
            public int WorkerID;
            public string Surname;
            public int IDdep;
            public Worker(int a, string b, int c)
            {
                this.WorkerID = a;
                this.Surname = b;
                this.IDdep = c;
            }
            public override string ToString()
            {
```

```

        return this.WorkerID.ToString() + ": Фамилия:" + this.Surname + ": ID
отделения сотрудника =" + this.IDdep;
    }
}

class Department
{
    public int IDdep;
    public string DepartmentName;
    public Department(int id, string name)
    {
        this.IDdep = id;
        this.DepartmentName = name;
    }
    public override string ToString()
    {
        return "(ID отделения =" + this.IDdep.ToString() + "; Название
отделения:" + this.DepartmentName + ")";
    }
}

class DepartmentWorker
{
    public int WorkerID;
    public int IDdep;
    public DepartmentWorker(int wid, int iddep)
    {
        this.WorkerID = wid;
        this.IDdep = iddep;
    }
}

static List<Worker> WorkerList = new List<Worker>()
{
    new Worker(1, "Алексеев", 1),
    new Worker(2, "Наврузов", 1),
    new Worker(3, "Арнаут", 3),
    new Worker(4, "Азарев", 3),
    new Worker(5, "МакГрегор", 2),
    new Worker(6, "Нурмагамедов", 2),
    new Worker(7, "Петров", 1),
    new Worker(8, "Иванов", 1),
};

static List<Department> DepartmentList = new List<Department>()
{
    new Department(1, "Кредитование"),
    new Department(2, "Вклады"),
    new Department(3, "Обслуживающий персонал"),
};

static List<DepartmentWorker> DWList = new List<DepartmentWorker>()
{
    new DepartmentWorker(1, 1),
    new DepartmentWorker(2, 1),
    new DepartmentWorker(3, 3),
    new DepartmentWorker(4, 3),
    new DepartmentWorker(5, 2),
    new DepartmentWorker(6, 2),
    new DepartmentWorker(7, 1),
    new DepartmentWorker(8, 1),
};

static void Main(string[] args)

```

```

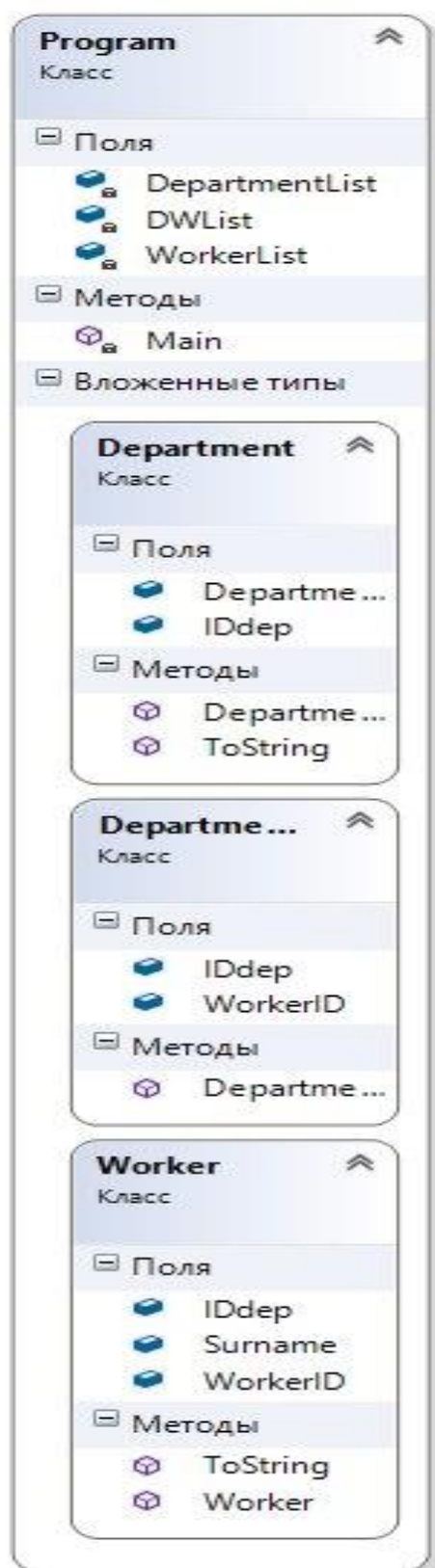
{
    Console.WriteLine("Список всех сотрудников и отделов, отсортированный по
отделам.");
    var AllWorkers = from x in DepartmentList
                      join s in WorkerList on x.IDdep equals s.IDdep into temp
                      select new { Department = x.DepartmentName, Worker = temp };
    foreach (var x in AllWorkers)
    {
        Console.WriteLine("                Тип отделения:=" +
x.Department);
        foreach (var s in x.Worker)
            Console.WriteLine("        " + s);
    }
    Console.WriteLine("Сотрудники на А");
    var AWorkers = from x in WorkerList where x.Surname.StartsWith("А") select
new { ID = x.WorkerID, Фамилия = x.Surname, };
    foreach (var x in AWorkers)
    {
        Console.WriteLine("Фамилия: " + x.Фамилия);
    }
    Console.WriteLine("Список всех отделов и количество сотрудников в них.");
    var NumWorkers = from x in DepartmentList
                      join s in WorkerList on x.IDdep equals s.IDdep into temp
                      select new { Department = x.DepartmentName, Quantity =
temp.Count() };
    foreach (var x in NumWorkers)
    {
        Console.WriteLine(x.Department + ":" + x.Quantity);
    }
    Console.WriteLine("Список всех отделов, в которых хотя бы у одного сотрудника
фамилия начинается на А.");
    var AnyWorkersA = from x in DepartmentList
                       join s in WorkerList on x.IDdep equals s.IDdep into temp
                       where temp.Any(x => x.Surname.StartsWith("А"))
                       select new { Department = x.DepartmentName };
    foreach (var x in AnyWorkersA)
    {
        Console.WriteLine("Название отдела:" + x.Department);
    }
    Console.WriteLine("Список всех отделов, в которых у каждого сотрудника фамилия
начинается на А.");
    var AllWorkersA = from x in DepartmentList
                       join s in WorkerList on x.IDdep equals s.IDdep into temp
                       where temp.All(x => x.Surname.StartsWith("А"))
                       select new { Department = x.DepartmentName };
    foreach (var x in AllWorkersA)
    {
        Console.WriteLine("Название отдела:" + x.Department);
    }
    Console.WriteLine("Список всех отделов и список сотрудников в каждом
отделе(Связь много-ко-многим).");
    var DWlink = from x in WorkerList
                  join y in DWList on x.WorkerID equals y.WorkerID into temp
                  from z1 in temp
                  join h in DepartmentList on z1.IDdep equals h.IDdep into temp2

                  from z2 in temp2
                  group z2 by z2.DepartmentName into g
                  select new
                  { Department = g.Key, Count = g.Count() };
    foreach (var d in DWlink)
    {
        Console.WriteLine(d.Department + ":" + d.Count);
    }
}

```

```
    }  
  }  
}
```

**Диаграмма классов:**



## Пример консольного вывода:

```
C:\Windows\system32\cmd.exe

Список всех сотрудников и отделов, отсортированный по отделам.
                                Тип отделения: =Кредитование
1: Фамилия:Алексеев: ID отделения сотрудника =1
2: Фамилия:Наврузов: ID отделения сотрудника =1
7: Фамилия:Петров: ID отделения сотрудника =1
8: Фамилия:Иванов: ID отделения сотрудника =1
                                Тип отделения: =Вклады
5: Фамилия:МакГрегор: ID отделения сотрудника =2
6: Фамилия:Нурмаганедов: ID отделения сотрудника =2
                                Тип отделения: =Обслуживающий персонал
3: Фамилия:Арнаут: ID отделения сотрудника =3
4: Фамилия:Азарев: ID отделения сотрудника =3

Сотрудники на А
Фамилия: Алексеев
Фамилия: Арнаут
Фамилия: Азарев
Список всех отделов и количество сотрудников в них.
Кредитование: 4
Вклады: 2
Обслуживающий персонал: 2
Список всех отделов, в которых хотя бы у одного сотрудника фамилия начинается на А
.
Название отдела: Кредитование
Название отдела: Обслуживающий персонал
Список всех отделов, в которых у каждого сотрудника фамилия начинается на А.
Название отдела: Обслуживающий персонал
Список всех отделов и список сотрудников в каждом отделе (Связь много-ко-многим )
.
Кредитование: 4
Обслуживающий персонал: 2
Вклады: 2
Для продолжения нажмите любую клавишу . . .
```