

**Московский государственный технический  
университет им. Н.Э. Баумана.**

**Факультет «Информатика и управление»**

**Кафедра ИУ5.**

**Курс «Базовые компоненты интернет-технологий»**

**Отчет по ДЗ**

Выполнил:  
студент группы ИУ5-31Б

Гришин Станислав

Подпись и дата:  
28.12.2020

Проверил:  
преподаватель каф.  
ИУ5

Гапанюк Ю.Е.

Подпись и дата:  
28.12.2020

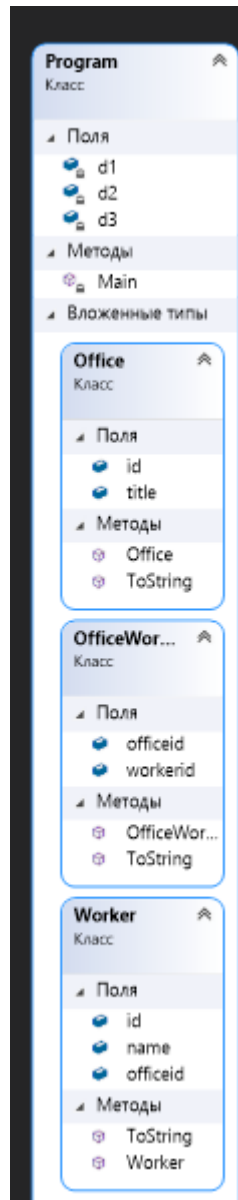
г. Москва, 2020 г.

## Постановка задачи

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.

## Разработка интерфейса класса



## Листинг программы

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

namespace Laba7
{
    class Program
    {
        public class Worker
        {
            /// <summary>
            /// Ключ
            /// </summary>
            public int id;

            public string name;
```

```

        public int officeid;

        /// <summary>
        /// Конструктор
        /// </summary>
        public Worker(int i, string g, int v)
        {
            this.id = i;
            this.name = g;
            this.officeid = v;
        }

        /// <summary>
        /// Приведение к строке
        /// </summary>
        public override string ToString()
        {
            return "(ID:" + this.id.ToString() + "; Фамилия:" + this.name + "; ID
офиса:" + this.officeid + ")";
        }
    }
    public class Office
    {
        /// <summary>
        /// Ключ
        /// </summary>
        public int id;

        public string title;
        public Office(int i, string g)
        {
            this.id = i;
            this.title = g;
        }
        public override string ToString()
        {
            return "(ID:" + this.id.ToString() + "; Название:" + this.title + ")";
        }
    }
    public class OfficeWorker
    {
        public int workerid;
        public int officeid;
        public OfficeWorker(int w, int o)
        {
            this.workerid = w;
            this.officeid = o;
        }
        public override string ToString()
        {
            return "(ID сотрудника:" + this.workerid.ToString() + "; ID офиса:" +
this.officeid + ")";
        }
    }
    static List<Worker> d1 = new List<Worker>()
    {
        new Worker(1, "Гришин", 2),
        new Worker(2, "Петров", 3),
        new Worker(3, "Трифонов", 4),
        new Worker(4, "Еремин", 5),
        new Worker(5, "Анищенков", 1),
        new Worker(6, "Сусаров", 3),
        new Worker(7, "Порядин", 2),
        new Worker(8, "Иванов", 3),
        new Worker(9, "Антонов", 1),
    }

```

```

        new Worker(10, "Алексеев", 1)
    };
    static List<Office> d2 = new List<Office>()
    {
        new Office(1, "Первый" ),
        new Office(2, "Второй"),
        new Office(3, "Третий"),
        new Office(4, "Четвертый"),
        new Office(5, "Пятый")
    };
    static List<OfficeWorker> d3 = new List<OfficeWorker>()
    {
        new OfficeWorker(1,2),
        new OfficeWorker(2,3),
        new OfficeWorker(3,4),
        new OfficeWorker(4,5),
        new OfficeWorker(5,1),
        new OfficeWorker(6,3),
        new OfficeWorker(7,2),
        new OfficeWorker(8,3),
        new OfficeWorker(9,4),
        new OfficeWorker(10,1)
    };
    static void Main(string[] args)
    {
        Console.WriteLine("Список сотрудников:");
        var q1 = from x in d1 select x;
        foreach (var x in q1) Console.WriteLine(x);
        Console.WriteLine("Список офисов:");
        var q2 = from y in d2 select y;
        foreach (var x in q2) Console.WriteLine(x);
        Console.WriteLine("Список всех сотрудников и отделов, отсортированный по
отделам:");
        var q3 = from x in d1
            join y in d2 on x.officeid equals y.id
            orderby y.title ascending
            select new { ID = x.id, Name = x.name, Title = y.title };
        foreach (var x in q3) Console.WriteLine(x);
        Console.WriteLine("Список всех сотрудников, у которых фамилия начинается с
буквы «А»:");
        var q4 = from x in d1
            where x.name.StartsWith('A') == true
            select new { ID=x.id, Name=x.name };
        foreach (var x in q4) Console.WriteLine(x);
        Console.WriteLine("Список отделов, в которых хотя бы у одного сотрудника
фамилия начинается с буквы «А»:");
        var q5 = from x in d1
            from y in d2
            where x.officeid == y.id && x.name.StartsWith('A') == true
            select new { ID = y.id, Name = y.title };
        foreach (var x in q5.Distinct()) Console.WriteLine(x);
        Console.WriteLine("Список всех отделов и количество сотрудников в каждом
отделе:");
        var q6 = from y in d2
            join x in d1 on y.id equals x.officeid into temp
            from t in temp
            select new { ID = y.id, Name = y.title, Count = temp.Count() };
        foreach (var x in q6.Distinct()) Console.WriteLine(x);
        Console.WriteLine("Список всех отделов и список сотрудников в каждом
отделе:");
        var q8 = from y in d2
            join z in d3 on y.id equals z.officeid into temp
            from x in d1
            from t in temp
            where x.id == t.workerid

```

```

        select new { d = y, e = x };
var q9 = from x in q8.Union(q8)
        group x by x.d.title into temp
        select new { Key = temp.Key, Values = temp };
foreach (var x in q9)
{
    Console.WriteLine(x.Key);
    foreach (var k in x.Values)
        Console.WriteLine("ID: " + k.e.id + "; Name:" + k.e.name);
}
Console.WriteLine("Список всех отделов и количество сотрудников в каждом
отделе:");
var q10 = from y in d2
        join z in d3 on y.id equals z.officeid into temp
        join x in d1 on y.id equals x.officeid
        from t in temp
        select new { ID = y.id, Name = y.title, Count = temp.Count() }; ;
foreach (var x in q10.Distinct())
    Console.WriteLine(x);
}
}
}

```

## Анализ результатов

```

Список сотрудников:
(ID:1; Фамилия:Гришин; ID офиса:2)
(ID:2; Фамилия:Петров; ID офиса:3)
(ID:3; Фамилия:Трифонов; ID офиса:4)
(ID:4; Фамилия:Еремин; ID офиса:5)
(ID:5; Фамилия:Анищенков; ID офиса:1)
(ID:6; Фамилия:Сусаров; ID офиса:3)
(ID:7; Фамилия:Порядин; ID офиса:2)
(ID:8; Фамилия:Иванов; ID офиса:3)
(ID:9; Фамилия:Антонов; ID офиса:1)
(ID:10; Фамилия:Алексеев; ID офиса:1)
Список офисов:
(ID:1; Название:Первый)
(ID:2; Название:Второй)
(ID:3; Название:Третий)
(ID:4; Название:Четвертый)
(ID:5; Название:Пятый)
Список всех сотрудников и отделов, отсортированный по отделам:
{ ID = 1, Name = Гришин, Title = Второй }
{ ID = 7, Name = Порядин, Title = Второй }
{ ID = 5, Name = Анищенков, Title = Первый }
{ ID = 9, Name = Антонов, Title = Первый }
{ ID = 10, Name = Алексеев, Title = Первый }
{ ID = 4, Name = Еремин, Title = Пятый }
{ ID = 2, Name = Петров, Title = Третий }
{ ID = 6, Name = Сусаров, Title = Третий }
{ ID = 8, Name = Иванов, Title = Третий }
{ ID = 3, Name = Трифонов, Title = Четвертый }
Список всех сотрудников, у которых фамилия начинается с буквы <А>:
{ ID = 5, Name = Анищенков }
{ ID = 9, Name = Антонов }
{ ID = 10, Name = Алексеев }
Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы

{ ID = 4, Name = Четвертый, Count = 1 }
{ ID = 5, Name = Пятый, Count = 1 }
Список всех отделов и список сотрудников в каждом отделе:
Первый
ID: 5; Name:Анищенков
ID: 10; Name:Алексеев
Второй
ID: 1; Name:Гришин
ID: 7; Name:Порядин
Третий
ID: 2; Name:Петров
ID: 6; Name:Сусаров

```

ID: 8; Name:Иванов

Четвертый

ID: 3; Name:Трифонов

ID: 9; Name:Антонов

Пятый

ID: 4; Name:Еремин

Список всех отделов и количество сотрудников в каждом отделе:

{ ID = 1, Name = Первый, Count = 2 }

{ ID = 2, Name = Второй, Count = 2 }

{ ID = 3, Name = Третий, Count = 3 }

{ ID = 4, Name = Четвертый, Count = 2 }

{ ID = 5, Name = Пятый, Count = 1 }