

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

**Факультет «Радиотехнический»  
Кафедра ИУ5 «Системы обработки информации и управления»**

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

**Пояснительная записка по домашнему заданию**

Выполнил:

студент группы РТ5-31Б  
Борисочкин М.И.

Подпись и дата:

Проверил:

к.т.н., доцент  
Гапанюк Ю. Е.

Подпись и дата:

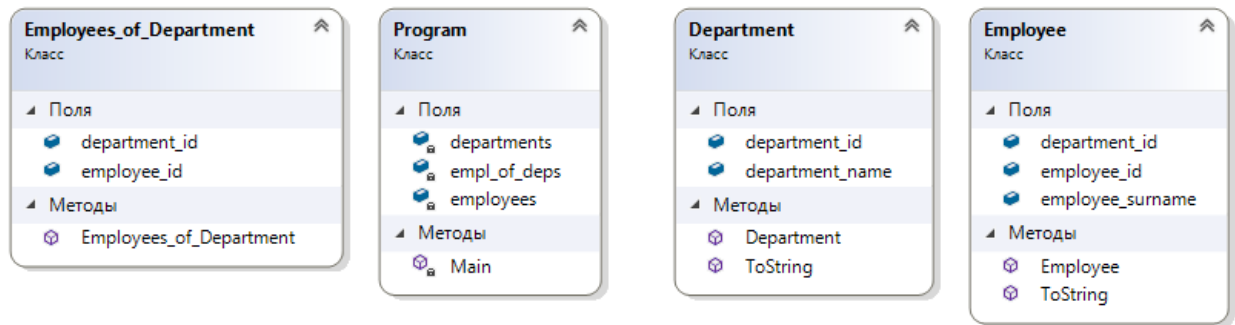
Москва, 2020 г.

## 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.

## Диаграмма классов



## Текст программы

### Employee.cs

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

namespace LINQProj
{
    /// <summary>
    /// Класс "Сотрудник"
    /// </summary>
    class Employee
    {
        /// <summary>
        /// ID сотрудника
        /// </summary>
        public int employee_id;

        /// <summary>
        /// Фамилия сотрудника
        /// </summary>
        public string employee_surname;

        /// <summary>
        /// ID отдела
        /// </summary>
        public int department_id;

        /// <summary>
        /// Конструктор
        /// </summary>
        public Employee(int ei, string esn, int di)
        {
            this.employee_id = ei;
            this.employee_surname = esn;
            this.department_id = di;
        }
    }
}
```

```

    /// <summary>
    /// Приведение к строке
    /// </summary>
    public override string ToString()
    {
        return "ID сотрудника: " + this.employee_id.ToString() + "; Фамилия
сотрудника: " + this.employee_surname + "; ID отдела:" + this.department_id.ToString();
    }
}
}

```

## Department.cs

```

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

namespace LINQProj
{
    /// <summary>
    /// Класс "Отдел"
    /// </summary>
    class Department
    {
        /// <summary>
        /// ID отдела
        /// </summary>
        public int department_id;

        /// <summary>
        /// Наименование отдела
        /// </summary>
        public string department_name;

        /// <summary>
        /// Конструктор
        /// </summary>
        public Department(int di, string dn)
        {
            this.department_id = di;
            this.department_name = dn;
        }

        /// <summary>
        /// Приведение к строке
        /// </summary>
        public override string ToString()
        {
            return "ID отдела: " + this.department_id.ToString() + "; Наименование
отдела: " + this.department_name;
        }
    }
}

```

## Employees\_of\_Department.cs

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

namespace LINQProj
{
    class Employees_of_Department
    {
        public int employee_id;

        public int department_id;

        public Employees_of_Department(int eid, int did)
        {
            this.employee_id = eid;
            this.department_id = did;
        }
    }
}
```

## Program.cs

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

namespace LINQProj
{
    class Program
    {
        // <summary>
        /// Сотрудники
        /// </summary>
        static List<Employee> employees = new List<Employee>()
        {
            new Employee(1, "Афанасьев", 1),
            new Employee(2, "Демченков", 1),
            new Employee(3, "Борисочкин", 2),
            new Employee(4, "Герасимов", 2),
            new Employee(5, "Чернов", 1),
            new Employee(6, "Рогова", 1),
            new Employee (7, "Аникин", 3),
            new Employee (8, "Азхар", 3),
            new Employee (9, "Алиев", 3)
        };

        // <summary>
        /// Отделы
        /// </summary>
        static List<Department> departments = new List<Department>()
        {
            new Department (2, "Учебный"),
            new Department (1, "Научный"),
            new Department (3, "Административный")
        };

        // <summary>
        /// Сотрудники отдела
    }
}
```

```

    /// </summary>
    static List<Employees_of_Department> empl_of_deps = new
List<Employees_of_Department>()
    {
        new Employees_of_Department(1,1),
        new Employees_of_Department(2,1),
        new Employees_of_Department(3,2),
        new Employees_of_Department(4,2),
        new Employees_of_Department(5,1),
        new Employees_of_Department(6,1),
        new Employees_of_Department(7,3),
        new Employees_of_Department(8,3),
        new Employees_of_Department(9,3)
    };

    static void Main(string[] args)
    {
        // Сортировка сотрудников по отделу
        Console.WriteLine("Список сотрудников, отсортированных по отделу");
        var q1 = from x in employees
                orderby x.department_id, x.employee_surname
                select x;
        foreach (var x in q1)
            Console.WriteLine(x);

        // Сотрудники с фамилией на А
        Console.WriteLine("\nСписок сотрудников с фамилией на А \n");
        var q3 = from x in employees
                where (x.employee_surname[0].ToString().ToUpper() == "A")
                select x;
        foreach (var x in q3)
            Console.WriteLine(x);

        // Количество сотрудников в отделах
        Console.WriteLine("\nСписок отделов с количеством сотрудников в них");
        var q4 = from x in employees
                group x by x.department_id into egrp
                select new { ID = egrp.Key, Value = egrp.Count()};
        foreach(var x in q4)
        {
            Department dtemp = new Department(0, " ");

            var f1 = (from y in departments select y).First(y => y.department_id ==
x.ID);

            dtemp = f1;

            Console.WriteLine(dtemp.department_name + ": " + x.Value.ToString());
        }

        // Отделы, в которых у всех сотрудников фамилия на А
        Console.WriteLine("\nСписок отделов, в которых у всех сотрудников фамилия
начинается на А");
        var q5 = from x in q3
                group x by x.department_id into egrp
                select new { ID = egrp.Key, Value = egrp.Count()};

        foreach (var x in q5)
        {
            foreach(var y in q4)
            {
                if(x.ID == y.ID && x.Value == y.Value)
                {
                    Department dtemp = new Department(0, " ");

```

```

        var f1 = (from z in departments select z).First(z =>
z.department_id == x.ID);
        dtemp = f1;

        Console.WriteLine(dtemp.department_name);
    }
}

// Отделы, в которых хотя бы у одного сотрудника фамилия на А
Console.WriteLine("\nСписок отделов, в которых хотя бы у одного сотрудника
фамилия начинается на А");
foreach (var x in q5)
{
    foreach (var y in q4)
    {
        if (x.ID == y.ID)
        {
            Department dtemp = new Department(0, " ");

            var f1 = (from z in departments select z).First(z =>
z.department_id == x.ID);
            dtemp = f1;

            Console.WriteLine(dtemp.department_name);
        }
    }
}

// Список сотрудников каждого отдела
Console.WriteLine("\nСписок отделов и сотрудников в них");
var q6 = from y in departments
        join l1 in empl_of_deps on y.department_id equals l1.department_id
into temp1
        from t1 in temp1
        join l2 in employees on t1.employee_id equals l2.employee_id into
temp2
        from t2 in temp2
        select new {ID_d = t1.department_id, Name_d = y.department_name,
ID_e = t1.employee_id, Name_e = t2.employee_surname};

foreach(var i in departments)
{
    Console.WriteLine("\n" + i.department_name + ":");
    foreach(var x in q6)
    {
        if (x.ID_d == i.department_id)
            Console.WriteLine(x.Name_e);
    }
}

// Количество сотрудников в отделах
Console.WriteLine("\nСписок отделов и количество сотрудников в них");
var q7 = from x in employees
        group x by x.department_id into egrp
        select new { ID = egrp.Key, Value = egrp};

foreach (var x in q7)
{
    Department dtemp = new Department(0, " ");
    var f1 = (from y in departments select y).First(y => y.department_id ==
x.ID);
    dtemp = f1;

```

```

        Console.WriteLine(dtemp.department_name + ": " +
x.Value.Count().ToString());
    }
    Console.ReadLine();
}
}
}

```

## Примеры выполнения программы

```

Выбрать C:\Users\Михаил\Documents\Visual Studio 2017\Projects\C#\LINQProj\LINQProj\bin\Debug\LINQProj.exe
Список сотрудников, отсортированных по отделу
ID сотрудника: 1; Фамилия сотрудника: Афанасьев; ID отдела:1
ID сотрудника: 2; Фамилия сотрудника: Демченков; ID отдела:1
ID сотрудника: 6; Фамилия сотрудника: Рогова; ID отдела:1
ID сотрудника: 5; Фамилия сотрудника: Чернов; ID отдела:1
ID сотрудника: 3; Фамилия сотрудника: Борисочкин; ID отдела:2
ID сотрудника: 4; Фамилия сотрудника: Герасимов; ID отдела:2
ID сотрудника: 8; Фамилия сотрудника: Азхар; ID отдела:3
ID сотрудника: 9; Фамилия сотрудника: Алиев; ID отдела:3
ID сотрудника: 7; Фамилия сотрудника: Аникин; ID отдела:3

Список сотрудников с фамилией на А
ID сотрудника: 1; Фамилия сотрудника: Афанасьев; ID отдела:1
ID сотрудника: 7; Фамилия сотрудника: Аникин; ID отдела:3
ID сотрудника: 8; Фамилия сотрудника: Азхар; ID отдела:3
ID сотрудника: 9; Фамилия сотрудника: Алиев; ID отдела:3

Список отделов с количеством сотрудников в них
Научный: 4
Учебный: 2
Административный: 3

Список отделов, в которых у всех сотрудников фамилия начинается на А
Административный

Список отделов, в которых хотя бы у одного сотрудника фамилия начинается на А
Научный
Административный

Список отделов и сотрудников в них
Учебный:
Борисочкин
Герасимов

Научный:
Афанасьев
Демченков
Чернов
Рогова

Административный:
Аникин
Азхар
Алиев

Список отделов и количество сотрудников в них
Научный: 4
Учебный: 2
Административный: 3

```