

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

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

«Работа с LINQ to Objects»

Выполнил:

студент группы ИУ5-31Б

Васюнин Михаил Андреевич

Проверил:

преподаватель каф. ИУ5

Гапанюк Юрий
Евгеньевич

Подпись и дата:

Подпись и дата:

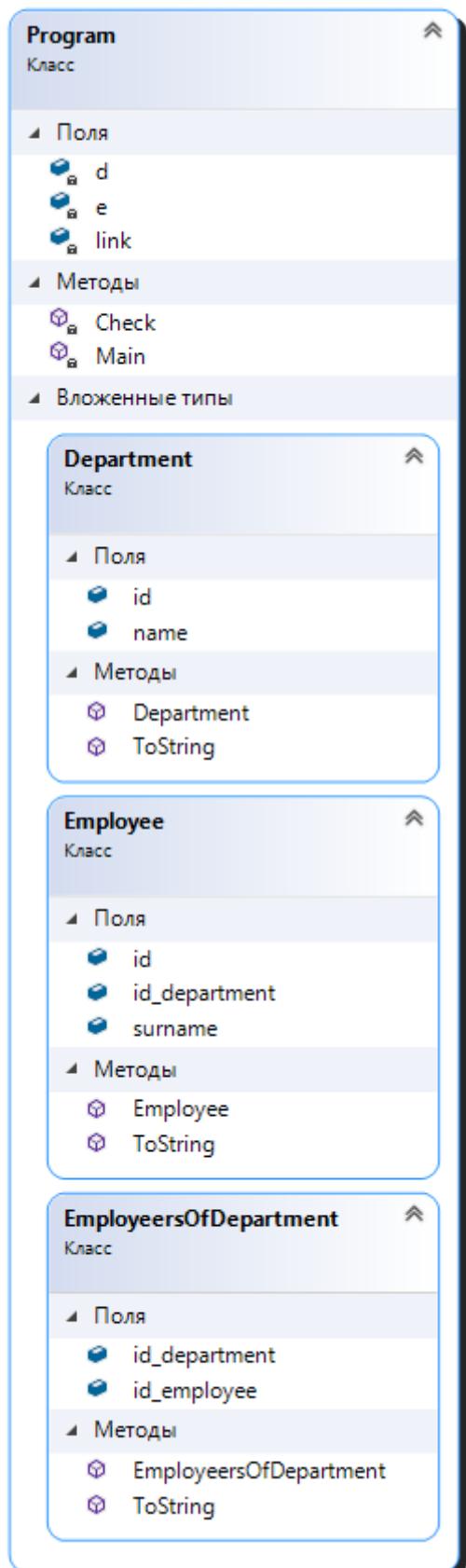
Москва, 2020 г

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;

namespace DZ_LINQ
{
    class Program
    {
        /// <summary>
        /// Класс "Сотрудник"
        /// </summary>
        public class Employee
        {
            public int id;
            public string surname;
            public int id_department;
            public Employee(int id_, string surname_, int id_department_)
            {
                this.id = id_;
                this.surname = surname_;
                this.id_department = id_department_;
            }
            public override string ToString()
            {
                return "id = " + id + "; сотрудник: " + surname + "; номер отдела: " + id_department;
            }
        }
        /// <summary>
        /// Класс "Отдел"
        /// </summary>
        public class Department
        {
            public int id;
            public string name;
            public Department(int id_, string name_)
            {
                this.id = id_;
                this.name = name_;
            }
            public override string ToString()
            {
                return "id = " + id + "; название: " + name;
            }
        }
        public class EmployeersOfDepartment
        {
```

```

public int id_employee;
public int id_department;
public EmployeersOfDepartment(int idEmployee, int idDepartment)
{
    this.id_employee = idEmployee;
    this.id_department = idDepartment;
}
public override string ToString()
{
    return "ID сотрудника: " + id_employee + "; ID отдела " + id_department;
}
}

static List<Employee> e = new List<Employee>()
{
    new Employee(1, "Льдов", 1),
    new Employee(2, "Семчук", 1),
    new Employee(3, "Аверьянов", 2),
    new Employee(4, "Миронов", 3),
    new Employee(5, "Чистова", 4),
    new Employee(6, "Аверин", 4),
    new Employee(7, "Антонов", 2)
};

static List<Department> d = new List<Department>()
{
    new Department(1, "Отдел разработки"),
    new Department(2, "Отдел кадров"),
    new Department(3, "Сборочный цех"),
    new Department(4, "Отдел продаж"),
    new Department(5, "Цех установки")
};

static List<EmployeersOfDepartment> link = new List<EmployeersOfDepartment>()
{
    new EmployeersOfDepartment (1,1),
    new EmployeersOfDepartment (2,1),
    new EmployeersOfDepartment (3,2),
    new EmployeersOfDepartment (4,3),
    new EmployeersOfDepartment (5,4),
    new EmployeersOfDepartment (6,4),
    new EmployeersOfDepartment (7,2),
    new EmployeersOfDepartment (1,3),
    new EmployeersOfDepartment (2,4)
};

static void Main(string[] args)
{
}

```

```

Console.WriteLine("Список всех сотрудников, отсортированных по отделам");
var t1 = from x in e
    orderby x.id_department
    select x;

foreach (var x in t1) Console.WriteLine(x);

//-----
Console.WriteLine();
//-----

Console.WriteLine("Список сотрудников, у которых фамилия начинается с буквы
'A\");
var t2 = from x in e
    where x.surname[0] == 'A'
    select x;
foreach (var x in t2) Console.WriteLine(x);

//-----
Console.WriteLine();
//-----


Console.WriteLine("Список отделов и количество сотрудников в каждом");
var t3 = from y in d
    join x in e on y.id equals x.id_department into temp
    from t in temp
    select new { y, count = temp.Count() };
foreach (var x in t3.Distinct()) Console.WriteLine(x);

//-----
Console.WriteLine();
//-----


Console.WriteLine("Список отделов, в которых у всех сотрудников фамилия
начинается с буквы 'A\");
var t41 = from y in d
    from x in e
    where (y.id == x.id_department && x.surname[0] != 'A')
    select y;
var t42 = from y in d
    from x in e
    where y.id == x.id_department
    select y;
t42 = t42.Distinct();
List<Department> null_list = new List<Department>();
Check(t42, ref null_list, d);
foreach (var x in d)
{
    bool tr = true;

```

```

        foreach (var y in t41)
            if (y.id == x.id) tr = false;
        foreach (var nul in null_list)
            if (x.id == nul.id) tr = true;
        if (tr) Console.WriteLine(x);
    }

//-----
Console.WriteLine();
//-----
```

Console.WriteLine("Отделы, в которых хотя бы у одного сотрудника фамилия начинается с буквы \'A\'");

```

var t5 = from y in d
    from x in e
    where (y.id == x.id_department && x.surname[0] == 'A')
    select y;
foreach (var x in t5.Distinct()) Console.WriteLine(x);

//*****
//-----
Console.WriteLine();
//-----
```

Console.WriteLine("Список всех отделов и список сотрудников в каждом");

```

var t6 = from y in d
    join l in link on y.id equals l.id_department into temp
    from x in e
    from tm1 in temp
    where x.id == tm1.id_employee
    select new { d = y, e = x };
var t61 = from x in t6.Union(t6)
    group x by x.d.name into temp
    select new { Key = temp.Key, values = temp };
foreach (var x in t61)
{
    Console.WriteLine(x.Key);
    foreach (var k in x.values)
        Console.WriteLine("\t id = " + k.e.id + "; " + k.e.surname);
}

//-----
Console.WriteLine();
//-----
```

Console.WriteLine("Список всех отделов и количество сотрудников в каждом");

```

var t7 = from y in d
    join l in link on y.id equals l.id_department into temp
```

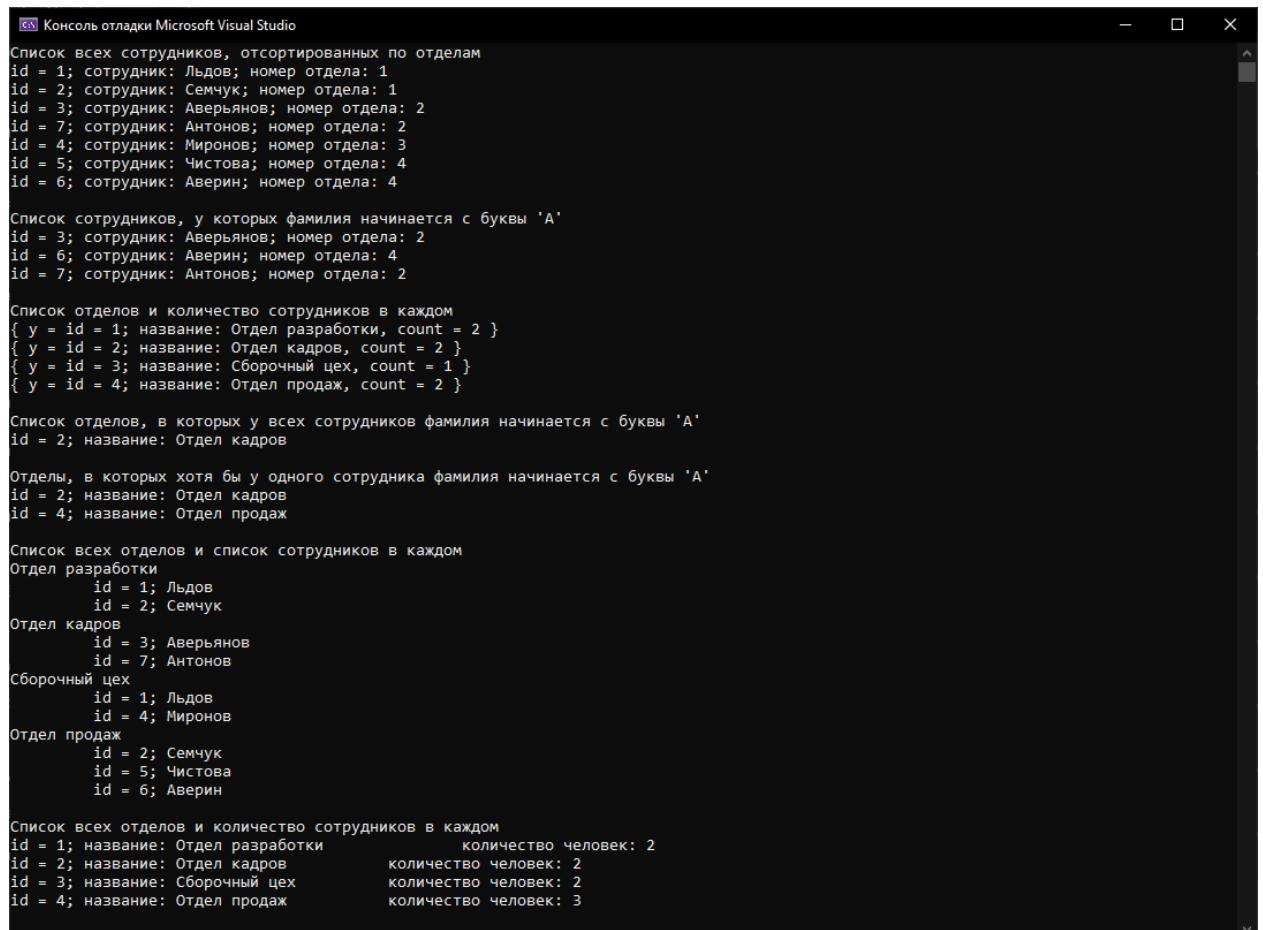
```

from x in e
from tm1 in temp
where x.id == tm1.id_employee
select new { y, e = temp.Count() };
foreach (var x in t7.Distinct()) Console.WriteLine(x.y + "\t\t количество человек: " + x.e);
}

static void Check(IEnumerable<Department> t42, ref List<Department> null_list,
List<Department> d)
{
    if (t42.Count() != d.Count())
        foreach (var x in d)
    {
        bool tr = true;
        foreach (var y in t42)
            if (x.id == y.id) tr = false;
        if (tr) null_list.Add(x);
    }
}
}
}

```

Анализ результатов



Консоль отладки Microsoft Visual Studio

```

Список всех сотрудников, отсортированных по отделам
id = 1; сотрудник: Льдов; номер отдела: 1
id = 2; сотрудник: Семчук; номер отдела: 1
id = 3; сотрудник: Аверьянов; номер отдела: 2
id = 7; сотрудник: Антонов; номер отдела: 2
id = 4; сотрудник: Миронов; номер отдела: 3
id = 5; сотрудник: Чистова; номер отдела: 4
id = 6; сотрудник: Аверин; номер отдела: 4

Список сотрудников, у которых фамилия начинается с буквы 'А'
id = 3; сотрудник: Аверьянов; номер отдела: 2
id = 6; сотрудник: Аверин; номер отдела: 4
id = 7; сотрудник: Антонов; номер отдела: 2

Список отделов и количество сотрудников в каждом
{ y = id = 1; название: Отдел разработки, count = 2 }
{ y = id = 2; название: Отдел кадров, count = 2 }
{ y = id = 3; название: Сборочный цех, count = 1 }
{ y = id = 4; название: Отдел продаж, count = 2 }

Список отделов, в которых у всех сотрудников фамилия начинается с буквы 'А'
id = 2; название: Отдел кадров

Отделы, в которых хотя бы у одного сотрудника фамилия начинается с буквы 'А'
id = 2; название: Отдел кадров
id = 4; название: Отдел продаж

Список всех отделов и список сотрудников в каждом
Отдел разработки
    id = 1; Льдов
    id = 2; Семчук
Отдел кадров
    id = 3; Аверьянов
    id = 7; Антонов
Сборочный цех
    id = 1; Льдов
    id = 4; Миронов
Отдел продаж
    id = 2; Семчук
    id = 5; Чистова
    id = 6; Аверин

Список всех отделов и количество сотрудников в каждом
id = 1; название: Отдел разработки          количество человек: 2
id = 2; название: Отдел кадров              количество человек: 2
id = 3; название: Сборочный цех            количество человек: 2
id = 4; название: Отдел продаж             количество человек: 3

```