

Московский государственный технический
университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Лабораторная работа №7

Выполнила: Цветкова Алена

Группа: ИУ5-31

Дата:

Проверил: Гапанюк Ю.Е.

Дата:

Задание

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;
using System.Text;
using System.Threading.Tasks;
```

```
namespace лаб7
```

```

{
    class Program
    {
        static List<Member> memberList = new List<Member>()
        {
            new Member(1, "Ivanov", 1),
            new Member(2, "Alekseev", 1),
            new Member(3, "Petrov", 1),
            new Member(4, "Dmitrieva", 3),
            new Member(5, "Smirnov", 3),
            new Member(6, "Andreev", 2),
            new Member(7, "Afanasyev", 2)
        };

        static List<Department> departmentList = new List<Department>()
        {
            new Department(1, "Department one"),
            new Department(2, "Department two"),
            new Department(3, "Department three")
        };

        static List<DepMemLink> oneToMany = new List<DepMemLink>()
        {
            new DepMemLink(1, 1), //сотрудники отдела
            new DepMemLink(2, 1),
            new DepMemLink(3, 1),
            new DepMemLink(4, 2),
            new DepMemLink(5, 1),
            new DepMemLink(6, 2),
            new DepMemLink(7, 2),
            new DepMemLink(4, 3),
            new DepMemLink(5, 2),
            new DepMemLink(6, 1),
            new DepMemLink(7, 1),
            new DepMemLink(8, 3)
        };

        static void Main(string[] args)
        {
            // for (int i = 0; i < 100; i++) Console.Write('_');
            Console.WriteLine("Список всех сотрудников и отделов, отсортированный по
отделам\n");
            var allMemb = from t in departmentList
                          join s in memberList on t.property_1 equals
                          s.departmentID into temp
                          select new
                          {
                              Department = t.property_1,
                              Member = temp
                          };
            foreach (var s in allMemb)
            {
                Console.WriteLine("DepartmentID = " +
s.Department);
                foreach (var y in s.Member)
                    Console.WriteLine(y);
            }
            for (int i = 0; i < 100; i++) Console.Write('_');
            //=====
            //=====
            Console.WriteLine("\nСписок всех сотрудников, у которых фамилия начинается с буквы «А».\n");
            var MembFirstA = from t in memberList
                             where
t.surname.StartsWith("A")
                             select t;
            foreach (Member s in MembFirstA) Console.WriteLine(s);
        }
    }
}

```

```

        for (int i = 0; i < 100; i++) Console.Write('_');
//=====
//=====
//=====
Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе\n");
    var DepartAndQuantity = from a in departmentList
                             join b in memberList on a.property_1 equals
                             b.departmentID into temp
                             select new
                             {
                                 Department = a,
                                 Quantity =
                                     temp.Count()
                             };
    foreach (var c in DepartAndQuantity)
    {
        Console.WriteLine(c.Department + "\nQuantity of members = " +
            c.Quantity);
    }
    for (int i = 0; i < 100; i++) Console.Write('_');
//=====
//=====
//=====
Console.WriteLine("\nСписок отделов, в которых у всех сотрудников фамилия начинается с буквы
«А»\n");
var DepartAllMembFirstA = (from s in departmentList
                             from t in memberList
                             group t by t.departmentID into g
                             where g.All(t =>
                                 t.surname.StartsWith("A"))
                             select new
                             {
                                 Department = (from s in
                                                 departmentList
                                                 where s.property_1 == g.Key
                                                 select s)
                             });
    foreach (var s in DepartAllMembFirstA)
    {
        foreach (var b in s.Department)
        {
            Console.WriteLine(b);
        }
    }
    for (int i = 0; i < 100; i++) Console.Write('_');
//=====
//=====
//=====
Console.WriteLine("\nСписок отделов, в которых хотя бы у одного сотрудника фамилия
начинается с буквы «А»\n");
var DepartMembFirstA = (from s in departmentList
                         from t in memberList
                         group t by t.departmentID into g
                         where g.Any(t => t.surname.StartsWith("A"))
                         select new
                         {
                             Department = (from s in
                                             departmentList
                                             where s.property_1 == g.Key
                                             select s)
                         });
    foreach (var s in DepartMembFirstA)

```

```

        {
            foreach (var b in s.Department)
            {
                Console.WriteLine(b);
            }
        }
        for (int i = 0; i < 100; i++) Console.Write('_');
//=====
//=====
//=====
Console.WriteLine("\nСписок всех сотрудников и отделов, отсортированный по отделам\n");
var AllDepartAndMembers = (from t in memberList
                            join r in oneToMany on t.memberID equals
                            r.memberID into temp
                            from t1 in temp
                            group t by t1.departmentID into g
                            from t in departmentList
                            where t.property_1 == g.Key
                            select new { Members = g, department = t });
foreach (var s in AllDepartAndMembers)
{
    for (int i = 0; i < 80; i++) Console.Write('_');
    Console.WriteLine(s.department);
    for (int i = 0; i < 80; i++) Console.Write('_');
    foreach (var f in s.Members) Console.WriteLine(f);
}
//=====
//=====
//=====
Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе.\n");
var AllDepartAndQuantityOfMemb = (from t in memberList
                                   join r in oneToMany on t.memberID
                                   equals r.memberID into temp
                                   from t1 in temp
                                   group t by t1.departmentID into g
                                   from t in departmentList
                                   where t.property_1 == g.Key
                                   select new
                                   {
                                       Quantity =
                                           g.Count(),
                                       department = t
                                   });
foreach (var s in AllDepartAndQuantityOfMemb)
    Console.WriteLine(s.department + "\nQuantity of members = " + s.Quantity);
//=====
//=====
//=====
Console.ReadLine();
}
}
public class Member : IComparable
{
    public int memberID;
    public string surname;
    public int departmentID;
    public Member(int m, string s, int d)
    {
        memberID = m;
        surname = s;
        departmentID = d;
    }
}

```

```

        public override string ToString()
        {
            return ("\nMember ID= " + memberID + "\nSurname=" + surname + "\nDepartment ID = " + departmentID);
        }
        public int CompareTo(object a)
        {
            Member p = (Member)a;
            if (p.departmentID > this.departmentID) return -1;
            else if (p.departmentID < this.departmentID) return 1;
            else return 0;
        }
    }
}
class Department
{
    int departmentID;
    string NameOfDepartment;
    public Department(int id, string name)
    {
        this.departmentID = id;
        this.NameOfDepartment = name;
    }
    public int property_1
    {
        get { return this.departmentID; }
        set { }
    }
    public override string ToString()
    {
        return ("\nDepartment ID= " + departmentID + "\nName of department " + NameOfDepartment);
    }
}
class DepMemLink
{
    public int memberID;
    public int departmentID;
    public DepMemLink(int mID, int dID)
    {
        this.memberID = mID;
        this.departmentID = dID;
    }
}
}

```

Результат

Список всех сотрудников и отделов, отсортированный по отделам

DepartmentID = 1

Member ID= 1

Surname=Ivanov

Department ID = 1

Member ID= 2

Surname=Alekseev

Department ID = 1

Member ID= 3
Surname=Petrov
Department ID = 1
DepartmentID = 2

Member ID= 6
Surname=Andreev
Department ID = 2

Member ID= 7
Surname=Afanasyev
Department ID = 2
DepartmentID = 3

Member ID= 4
Surname=Dmitrieva
Department ID = 3

Member ID= 5
Surname=Smirnov
Department ID = 3

Список всех сотрудников, у которых фамилия начинается с буквы <А>.

Member ID= 2
Surname=Alekseev
Department ID = 1

Member ID= 6
Surname=Andreev
Department ID = 2

Member ID= 7
Surname=Afanasyev
Department ID = 2

Список всех отделов и количество сотрудников в каждом отделе

Department ID= 1
Name of department Department one
Quantity of members = 3

Department ID= 2
Name of department Department two
Quantity of members = 2

Department ID= 3
Name of department Department three

Quantity of members = 2

Список отделов, в которых у всех сотрудников фамилия начинается с буквы <А>

Department ID= 2

Name of department Department two

Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы <А>

Department ID= 1

Name of department Department one

Department ID= 2

Name of department Department two

Список всех сотрудников и отделов, отсортированный по отделам

Department ID= 1

Name of department Department one

Member ID= 1

Surname=Ivanov

Department ID = 1

Member ID= 2

Surname=Alekseev

Department ID = 1

Member ID= 3

Surname=Petrov

Department ID = 1

Member ID= 5

Surname=Smirnov

Department ID = 3

Member ID= 6

Surname=Andreev

Department ID = 2

Member ID= 7

Surname=Afanasyev

Department ID = 2

Department ID= 2

Name of department Department two

Member ID= 4
Surname=Dmitrieva
Department ID = 3

Member ID= 5
Surname=Smirnov
Department ID = 3

Member ID= 6
Surname=Andreev
Department ID = 2

Member ID= 7
Surname=Afanasyev
Department ID = 2

Department ID= 3
Name of department Department three

Member ID= 4
Surname=Dmitrieva
Department ID = 3

Список всех отделов и количество сотрудников в каждом отделе.

Department ID= 1
Name of department Department one
Quantity of members = 6

Department ID= 2
Name of department Department two
Quantity of members = 4

Department ID= 3
Name of department Department three
Quantity of members = 1