МГТУ имени Н.Э. Баумана
Факультет «Информатика и системы управления»

Базовые компоненты интернет технологий. Отчёт по лабораторной работе № 7.

Белкина Екатерина
Группа ИУ5-31Б

24 декабря 2018

Условие 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.

Текст программы

Department.cs

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

```
using System.Threading.Tasks;

namespace Lab7_Belkina
{
    class Department
    {
        public int Id { get; set; }
        public string Name { get; set; }

        public Department() { }
        public Department(int id, string name)
        {
            Id = id;
            Name = name;
        }

        public override string ToString()
        {
            return "Id = " + Id + "; Name = " + Name;
        }
    }
}
```

DepartmentsEmployees.cs

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

namespace Lab7_Belkina
{
    class DepartmentsEmployees
    {
        public int EmployeeId { get; set; }
        public int DepartmentId { get; set; }

        public DepartmentsEmployees(){ }

        public DepartmentsEmployees(int employeeId, int departmentId)
```

```
        {
            EmployeeId = employeeId;
            DepartmentId = departmentId;
        }

        public override string ToString()
        {
            return "Employee ID = " + EmployeeId + "; Department ID = " + DepartmentId;
        }
    }
}
```

Employee.cs

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

namespace Lab7_Belkina
{
    class Employee
    {
        public int Id { get; set; }
        public string Surname { get; set; }
        public int DepartmentId { get; set; }

        public Employee() { }
        public Employee(int id, string surname, int departmentId)
        {
            Id = id;
            Surname = surname;
            DepartmentId = departmentId;
        }

        public override string ToString()
        {
            return "Id = " + Id + "; Surname = " + Surname + "; Department Id = " + DepartmentId;
        }
    }
```

```
}
```

Program.cs

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

namespace Lab7_Belkina
{
   class Program
   {
      static void Main(string[] args)
      {
         var departments = new List<Department>
         {
            new Department(1, "Руководство"),
            new Department(2, "Бухгалтерия"),
            new Department(3, "Отдел производства"),
            new Department(4, "Отдел разработки"),
            new Department(5, "Отдел проектирования")
         };

         var employees = new List<Employee>
         {
            new Employee(1, "Амирова", 1),
            new Employee(2, "Ахубекова", 2),
            new Employee(3, "Петров", 2),
            new Employee(4, "Емельяненко", 5),
            new Employee(5, "Коновалов", 4),
            new Employee(6, "Горяев", 3)
         };

            Console.WriteLine("Список всех сотрудников и отделов, отсортированный по
   отделам");

         var list1 =
            from employee in employees
            join department in departments on employee.DepartmentId equals department.Id
```

```
    orderby department.Name
    select new
    {
        employee.Surname,
        DepartmentName = department.Name
    };

foreach (var item in list1)
{
    Console.WriteLine(item);
}

Console.WriteLine();
```

Console.WriteLine("Список всех сотрудников, у которых фамилия начинается с буквы «A»");

```
var list2 =
    from employee in employees
    where employee.Surname[0] == 'A'
    select employee;

foreach (var item in list2)
{
    Console.WriteLine(item);
}

Console.WriteLine();
```

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

```
var list3 =
    from department in departments
        join employee in employees on department.Id equals employee.DepartmentId into employeesInDepartment
    select new
```

```
        {
            department.Name,
            CountOfEmployees = employeesInDepartment.Count()
        };

    foreach (var item in list3)
    {
        Console.WriteLine(item);
    }

    Console.WriteLine();
```

Console.WriteLine("Список отделов, в которых у всех сотрудников фамилия начинается с буквы «A»");

```
    var list4 =
        from department in departments
            join employee in employees on department.Id equals employee.DepartmentId into employeesInDepartment
        where employeesInDepartment.All(employee => employee.Surname[0] == 'A')
        select department;

    foreach (var item in list4)
    {
        Console.WriteLine(item);
    }

    Console.WriteLine();
```

Console.WriteLine("Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «A»");

```
    var list5 =
        from department in departments
            join employee in employees on department.Id equals employee.DepartmentId into employeesInDepartment
        where employeesInDepartment.Any(employee => employee.Surname[0] == 'A')
```

```
        select department;

    foreach (var item in list5)
    {
        Console.WriteLine(item);
    }

    Console.WriteLine();

    var departmentsEmployees = new List<DepartmentsEmployees>
    {
        new DepartmentsEmployees(1,3),
        new DepartmentsEmployees(1,2),
        new DepartmentsEmployees(2,3),
        new DepartmentsEmployees(2,5),
        new DepartmentsEmployees(3,3),
        new DepartmentsEmployees(3,1),
        new DepartmentsEmployees(4,2),
        new DepartmentsEmployees(5,1),
        new DepartmentsEmployees(5,4),
        new DepartmentsEmployees(5,3),
        new DepartmentsEmployees(6,4),
        new DepartmentsEmployees(1,5)
    };

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

    var list6 =
        from de in departmentsEmployees
        group de by de.DepartmentId into deps
        select new
        {
            Department = departments.Single(dep => dep.Id == deps.Key),
                                Employees = string.Join(", ", employees.FindAll(employee =>
employee.DepartmentId == deps.Key))
        };

    foreach (var item in list6)
    {
        Console.WriteLine(item);
```

```
        }

        Console.WriteLine();



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

        var list7 =
            from de in departmentsEmployees
            group de by de.DepartmentId into deps
            select new
            {
                Department = departments.Single(dep => dep.Id == deps.Key),
                    EmployeesCount = employees.FindAll(employee => employee.DepartmentId == deps.Key).Count
            };

        foreach (var item in list7)
        {
            Console.WriteLine(item);
        }

        Console.WriteLine();
        Console.Read();
    }
  }
}
```

Примеры выполнения программы

```
Список всех сотрудников и отделов, отсортированный по отделам
{ Surname = Ахубекова, DepartmentName = Бухгалтерия }
{ Surname = Петров, DepartmentName = Бухгалтерия }
{ Surname = Емельяненко, DepartmentName = Отдел проектирования }
{ Surname = Горяев, DepartmentName = Отдел производства }
{ Surname = Коновалов, DepartmentName = Отдел разработки }
{ Surname = Амирова, DepartmentName = Руководство }

Список всех сотрудников, у которых фамилия начинается с буквы <А>
Id = 1; Surname = Амирова; Department Id = 1
Id = 2; Surname = Ахубекова; Department Id = 2

Список всех отделов и количество сотрудников в каждом отделе
{ Name = Руководство, CountOfEmployees = 1 }
{ Name = Бухгалтерия, CountOfEmployees = 2 }
{ Name = Отдел производства, CountOfEmployees = 1 }
{ Name = Отдел разработки, CountOfEmployees = 1 }
{ Name = Отдел проектирования, CountOfEmployees = 1 }

Список отделов, в которых у всех сотрудников фамилия начинается с буквы <А>
Id = 1; Name = Руководство

Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы
 <А>
Id = 1; Name = Руководство
Id = 2; Name = Бухгалтерия

Список всех отделов и список сотрудников в каждом отделе
{ Department = Id = 3; Name = Отдел производства, Employees = Id = 6; Surname =
Горяев; Department Id = 3 }
{ Department = Id = 2; Name = Бухгалтерия, Employees = Id = 2; Surname = Ахубеко
ва; Department Id = 2, Id = 3; Surname = Петров; Department Id = 2 }
{ Department = Id = 5; Name = Отдел проектирования, Employees = Id = 4; Surname
= Емельяненко; Department Id = 5 }
{ Department = Id = 1; Name = Руководство, Employees = Id = 1; Surname = Амирова
; Department Id = 1 }
{ Department = Id = 4; Name = Отдел разработки, Employees = Id = 5; Surname = Ко
новалов; Department Id = 4 }

Список всех отделов и количество сотрудников в каждом отделе
{ Department = Id = 3; Name = Отдел производства, EmployeesCount = 1 }
{ Department = Id = 2; Name = Бухгалтерия, EmployeesCount = 2 }
{ Department = Id = 5; Name = Отдел проектирования, EmployeesCount = 1 }
{ Department = Id = 1; Name = Руководство, EmployeesCount = 1 }
{ Department = Id = 4; Name = Отдел разработки, EmployeesCount = 1 }
```