**Московский государственный технический**

**университет им. Н. Э. Баумана**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №7

«Работа с LINQ»

Группа: ИУ5-32Б

Студент: Селедкина А. С.

Преподаватель: Гапанюк Ю. Е.

Москва, 2018 г.

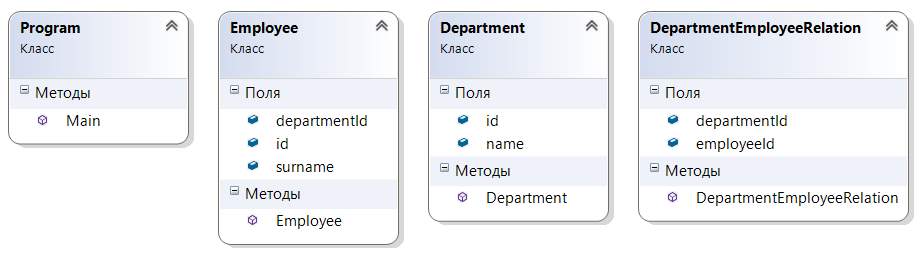
**Описание задания**

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**

Файл **Employee.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7

{

class Employee

{

public int id;

public string surname;

public int departmentId;

public Employee(int \_id, string \_surname, int \_departmentId)

{

id = \_id;

surname = \_surname;

departmentId = \_departmentId;

}

}

}

Файл **Department.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7

{

class Department

{

public int id;

public string name;

public Department(int \_id, string \_name)

{

id = \_id;

name = \_name;

}

}

}

Файл **DepartmentEmployeeRelation.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7

{

class DepartmentEmployeeRelation

{

public int employeeId;

public int departmentId;

public DepartmentEmployeeRelation(int \_empId, int \_depId)

{

employeeId = \_empId;

departmentId = \_depId;

}

}

}

Файл **Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

using System.Runtime.InteropServices;

using System.Text.RegularExpressions;

namespace lab7

{

class Program

{

public static void Main(string[] args)

{

List<Employee> employees = new List<Employee>

{

new Employee(1, "Seledkina", 1),

new Employee(2, "Sysoev", 1),

new Employee(3, "Dyatlenko", 1),

new Employee(4, "Brusnikina", 1),

new Employee(5, "Belova", 2),

new Employee(6, "Chernishev", 2),

new Employee(7, "Andreev", 2),

new Employee(8, "Alekseev", 3),

new Employee(9, "Arseniev", 3),

};

List<Department> departments = new List<Department>

{

new Department(1, "Development"),

new Department(2, "Sales"),

new Department(3, "Marketing")

};

List<DepartmentEmployeeRelation> depEmpRels = new List<DepartmentEmployeeRelation>

{

new DepartmentEmployeeRelation(1, 1),

new DepartmentEmployeeRelation(1, 2),

new DepartmentEmployeeRelation(2, 1),

new DepartmentEmployeeRelation(2, 3),

new DepartmentEmployeeRelation(3, 2),

new DepartmentEmployeeRelation(4, 1),

new DepartmentEmployeeRelation(5, 1),

new DepartmentEmployeeRelation(6, 2),

new DepartmentEmployeeRelation(6, 3),

new DepartmentEmployeeRelation(7, 2),

new DepartmentEmployeeRelation(7, 3),

new DepartmentEmployeeRelation(8, 2),

new DepartmentEmployeeRelation(9, 3),

};

Console.WriteLine("Departments:");

var q1 = from dep in departments

orderby dep.id

select dep.name;

foreach (var department in q1)

{

Console.WriteLine(department);

}

Console.WriteLine();

Console.WriteLine("Employees:");

var q2 = from emp in employees

orderby emp.departmentId

select emp.surname;

foreach (var employee in q2)

{

Console.WriteLine(employee);

}

Console.WriteLine();

Console.WriteLine("Employees with surnames starting with \"A\":");

var q3 = from emp in employees

where emp.surname[0] == 'A'

select emp.surname;

foreach (var employee in q3)

{

Console.WriteLine(employee);

}

Console.WriteLine();

Console.WriteLine("Группировка с функциями агрегирования");

var q4 = from dep in departments

join emp in employees on dep.id equals emp.departmentId into temp

select new { Name = dep.name, Cnt = temp.Count()};

foreach (var dep in q4)

{

Console.WriteLine("{0}: {1} employees", dep.Name, dep.Cnt);

}

Console.WriteLine();

Console.WriteLine("Group - All");

var q5 = from dep in departments

join emp in employees on dep.id equals emp.departmentId into temp

where temp.All(x => x.surname[0] == 'A')

select dep.name;

foreach (var dep in q5)

{

Console.WriteLine(dep);

}

Console.WriteLine();

Console.WriteLine("Group - Any");

var q6 = from dep in departments

join emp in employees on dep.id equals emp.departmentId into temp

where temp.Any(x => x.surname[0] == 'A')

select dep.name;

foreach (var dep in q6)

{

Console.WriteLine(dep);

}

Console.WriteLine();

Console.WriteLine("Many-to-many relation:");

var q7 = from dep in departments

join depEmpRel in depEmpRels on dep.id equals depEmpRel.departmentId into matchingRels

from depEmpRel in matchingRels

join emp in employees on depEmpRel.employeeId equals emp.id into matchingEmps

from link in matchingEmps

select new {Dep = dep.name, Emps = link.surname};

var q8 = from line in q7

group line by line.Dep into depEmps

select new { Dep = depEmps.Key, Emps = depEmps};

foreach (var dep in q8)

{

Console.WriteLine(dep.Dep);

foreach (var emp in dep.Emps)

{

Console.WriteLine("\t"+emp.Emps);

}

}

Console.WriteLine();

Console.WriteLine("Many-to-many relation (count):");

var q9 = from dep in departments

join depEmpRel in depEmpRels on dep.id equals depEmpRel.departmentId into matchingRels

from depEmpRel in matchingRels

join emp in employees on depEmpRel.employeeId equals emp.id into matchingEmps

from link in matchingEmps

select new {Dep = dep.name, Emps = link.surname};

var q10 = from line in q9

group line by line.Dep into depEmps

select new { Dep = depEmps.Key, Emps = depEmps.Count()};

foreach (var dep in q10)

{

Console.WriteLine(dep.Dep);

Console.WriteLine("Employee count : {0}\n", dep.Emps);

}

Console.ReadLine();

}

}

}

**Примеры выполнения программы**

