#### Московский государственный технический университет имени Н.Э. Баумана

## Лабораторная работа по дисциплине БКИТ №7

*Выполнил: Ханмурзин Тагир, ИУ5-34, 24.10.2017*

**1. 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.

**2. Листинг программного кода**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**namespace SimpleLINQ**

**{**

**class Program**

**{**

**public class Employee //Сотрудник**

**{**

**public int employeeID;**

**public string employeeSurname;**

**public int departmentID;**

**public Employee(int i, string s, int d)**

**{**

**this.employeeID = i;**

**this.employeeSurname = s;**

**this.departmentID = d;**

**}**

**public override string ToString()**

**{**

**return "Employee ID = " + this.employeeID.ToString() + " | Employee Surname = " + this.employeeSurname + " | Department ID = " + this.departmentID.ToString();**

**}**

**}**

**public class Department //Отдел**

**{**

**public int departmentID;**

**public string departmentName;**

**public Department(int i, string n)**

**{**

**this.departmentID = i;**

**this.departmentName = n;**

**}**

**public override string ToString()**

**{**

**return "Department ID = " + this.departmentID.ToString() + " | DepartmentName = " + this.departmentName;**

**}**

**}**

**public class EmployeesDepartment //Сотрудники отдела**

**{**

**public int employeeID;**

**public int departmentID;**

**public EmployeesDepartment(int ei, int di)**

**{**

**this.employeeID = ei;**

**this.departmentID = di;**

**}**

**public override string ToString()**

**{**

**return "Employee ID = " + this.employeeID.ToString() + "Department ID = " + this.departmentID.ToString();**

**}**

**}**

**static List<Employee> employee = new List<Employee>()**

**{**

**new Employee(1, "Sasha ", 3),**

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

**new Employee(3, "Tagir ", 2),**

**new Employee(4, "Sergei ", 3),**

**new Employee(5, "Cyrus ", 3),**

**new Employee(6, "Akimov ", 1),**

**new Employee(7, "Christina ", 1),**

**new Employee(8, "Paul ", 1),**

**new Employee(9, "Zhenya ", 3)**

**};**

**static List<Department> department = new List<Department>()**

**{**

**new Department(1, "Management "),**

**new Department(2, "Designers "),**

**new Department(3, "Programmers ")**

**};**

**static List<EmployeesDepartment> ed = new List<EmployeesDepartment>()**

**{**

**new EmployeesDepartment(1, 1),**

**new EmployeesDepartment(2, 2),**

**new EmployeesDepartment(3, 2),**

**new EmployeesDepartment(4, 3),**

**new EmployeesDepartment(5, 3),**

**new EmployeesDepartment(6, 1),**

**new EmployeesDepartment(7, 1),**

**new EmployeesDepartment(8, 1),**

**new EmployeesDepartment(5, 2),**

**new EmployeesDepartment(6, 3),**

**new EmployeesDepartment(7, 2),**

**new EmployeesDepartment(8, 3),**

**new EmployeesDepartment(9, 3),**

**};**

**static void Main(string[] args)**

**{**

**Console.ForegroundColor = ConsoleColor.Green; Console.WriteLine("Cписок всех сотрудников, отсортированный по отделам:"); Console.ResetColor();**

**var q3 = from x in employee**

**where x.departmentID >= 1**

**orderby x.departmentID ascending**

**select x;**

**foreach (var x in q3) Console.WriteLine(x);**

**//Console.WriteLine("Cписок всех сотрудников, у которых фамилия начинается с буквы «А»:");**

**//var q4 = from x in employee**

**// where x.employeeSurname[0] is 'A'**

**// orderby x.departmentID ascending**

**// select x;**

**//foreach (var x in q4) Console.WriteLine(x);**

**Console.ForegroundColor = ConsoleColor.Green; Console.WriteLine("\nCписок всех отделов и количество сотрудников в каждом отделе:"); Console.ResetColor();**

**var q5 = from x in department**

**join y in employee on x.departmentID equals y.departmentID into temp**

**from t in temp**

**select new { DepartmentNumber = x.departmentID, DepartmentName = x.departmentName, NumberEmployee = temp.Count() };**

**q5 = q5.Distinct();**

**foreach (var x in q5) Console.WriteLine(x);**

**//Console.WriteLine("Cписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А»:");**

**//var q6 = from x in employee**

**// from y in department**

**// where (x.employeeSurname[0] is 'A') &(x.departmentID == y.departmentID)**

**// select new { RoomNumber = y.officeID, RoomName = y.officeName, surname = x.surname };**

**//foreach (var x in q6) Console.WriteLine(x);**

**//Console.WriteLine("Cписок отделов, в которых у всех сотрудников фамилия начинается с буквы «А»:");**

**//var q7\_1 = from x in employee**

**// join y in q4 on x.departmentID equals y.departmentID into temp**

**// from t in temp**

**// select new { RoomNumber = x.officeID, number = temp.Count() };**

**//q7\_1 = q7\_1.Distinct();**

**//var q7 = from x in q5**

**// from y in q7\_1**

**// where (x.number == y.number) && (x.NumberDepartment == y.NumberDepartment)**

**// select new { RoomNumber = x.RoomNumber };**

**//q7 = q7.Distinct();**

**//foreach (var x in q7)**

**// Console.WriteLine(x);**

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

**var q8\_1 = from z in employee**

**join x in ed on z.departmentID equals x.departmentID into temp**

**from t1 in temp**

**join y in department on t1.departmentID equals y.departmentID into temp2**

**from t2 in temp2**

**select new { id = z.departmentID, name = t2.departmentName };**

**q8\_1 = q8\_1.Distinct();**

**foreach (var x in q8\_1)**

**Console.WriteLine(x);**

**var q8\_2 = from x in employee**

**join l in ed on x.employeeID equals l.employeeID into temp**

**from t1 in temp**

**join y in employee on t1.employeeID equals y.employeeID into temp2**

**from t2 in temp2**

**select new { id = x.employeeID, surname = t2.employeeSurname };**

**q8\_2 = q8\_2.Distinct();**

**foreach (var x in q8\_2)**

**Console.WriteLine(x);**

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

**var q9\_1 = from x in ed**

**join y in employee on x.departmentID equals y.departmentID into temp**

**from t in temp**

**select new { number = temp.Count(), id = t.departmentID };**

**q9\_1 = q9\_1.Distinct();**

**var q9\_2 = from x in employee**

**join ed in ed on x.employeeID equals ed.employeeID into temp**

**from t1 in temp**

**join y in department on t1.departmentID equals y.departmentID into temp2**

**from t2 in temp2**

**select new { name = t2.departmentName, id = t2.departmentID };**

**q9\_2 = q9\_2.Distinct();**

**var q9 = from x in q9\_1**

**from y in q9\_2**

**where x.id == y.id**

**select new { name = y.name, number = x.number };**

**q9 = q9.Distinct();**

**foreach (var x in q9)**

**Console.WriteLine(x);**

**Console.ReadKey();**

**}**

**}**

**}**

**3. Тест работы**

