Московский государственный технический университет

им. Н.Э. Баумана

Отчет по лабораторной работе № 7

|  |  |
| --- | --- |
|  | Выполнила:  Персиянова В.  ИУ5-33 |

г. Москва

2017 г.

**Описание задания лабораторной работы:**

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 многие-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 на языке C#.**

using System;

using System.Collections.Generic;

using System.Linq;

namespace Laba7

{

class Program

{

class Worker

{

public int id;

public string last\_name;

public int id\_department;

public Worker(int i, string l, int d)

{

id = i;

last\_name = l;

id\_department = d;

}

public override string ToString()

{

return id + " Фамилия: " + last\_name + "\n ID отдела: " + id\_department;

}

}

class Department

{

public int id;

public string name;

public Department(int i, string n)

{

id = i;

name = n;

}

public override string ToString()

{

return id + " " + name;

}

}

class Departament\_workers

{

public int worker\_id;

public int dep\_id;

public Departament\_workers(int i, int j)

{

worker\_id = i;

dep\_id = j;

}

}

static void Main(string[] args)

{

List<Worker> workers = new List<Worker>()

{

new Worker(1, "Алеутдинов", 1),

new Worker(2, "Воснецов", 1),

new Worker(3, "Иванов", 1),

new Worker(4, "Васильев", 2),

new Worker(5, "Антонов", 3),

new Worker(6, "Медяников", 2),

new Worker(7, "Аргентинов", 3)

};

List<Department> departments = new List<Department>()

{

new Department(1, "Отдел тех .защиты"),

new Department(2, "Отдел обслуживания"),

new Department(3, "Отдел мед. помощи")

};

List<Departament\_workers> dep\_workers = new List<Departament\_workers>()

{

new Departament\_workers(1, 1),

new Departament\_workers(1, 2),

new Departament\_workers(2, 1),

new Departament\_workers(3, 1),

new Departament\_workers(4, 2),

new Departament\_workers(4, 3),

new Departament\_workers(5, 3),

new Departament\_workers(6, 2),

new Departament\_workers(7, 3)

};

Console.WriteLine("Список всех сотрудников: ");

var q1 = from x in departments

join y in workers on x.id equals y.id\_department into t

select new { dep = x, dep\_workers = t };

foreach (var x in q1)

{

Console.WriteLine(x.dep);

foreach (var y in x.dep\_workers)

{

Console.WriteLine(" " + y);

}

Console.WriteLine();

}

Console.WriteLine("Сотрудники с фамилией, начинающейся на А");

var q2 = from x in workers

where x.last\_name[0] == 'А'

select x;

foreach (var x in q2)

{

Console.WriteLine(x);

}

Console.WriteLine("\n" + "Отделы и количество сотрудников");

var q3 = from x in departments select x;

foreach (var x in q3)

{

Console.WriteLine(x);

var q4 = from y in workers

where y.id\_department == x.id

select y;

Console.WriteLine("Количество сотрудников: " + q4.Count());

}

Console.WriteLine("\nОтделы, в которых у всех сотрудников фамилия начинается с буквы А");

var q5 = from x in workers

group x by x.id\_department into g

where g.All(x => x.last\_name[0] == 'А')

select g.Key;

foreach (var x in q5)

{ var q6 = from y in departments

where y.id == x

select y;

Console.WriteLine(q6.ElementAt(0));

}

Console.WriteLine("\nОтделы, в которых хотя бы у одного сотрудника фамилия на А");

var q7 = from x in workers

group x by x.id\_department into g

where g.Any(x => x.last\_name[0] == 'А')

select g.Key;

foreach (var x in q7)

{

var q8 = from y in departments

where y.id == x

select y;

Console.WriteLine(q8.ElementAt(0));

}

Console.WriteLine("\nМногие-ко-многим");

var q9 = from x in departments

join l in dep\_workers on x.id equals l.dep\_id into temp

from t in temp

group t by t.dep\_id into g

select new { dep = g.Key, work = g };

foreach (var x in q9)

{

Console.WriteLine(x.dep + " отдел");

foreach (var y in x.work)

{

Console.WriteLine(" " + y.worker\_id + " сотрудник");

}

}

Console.WriteLine("\nКоличество сотрудников");

foreach (var x in q9)

{

Console.WriteLine(x.dep + " отдел, количество сотрудников - " + x.work.Count());

}

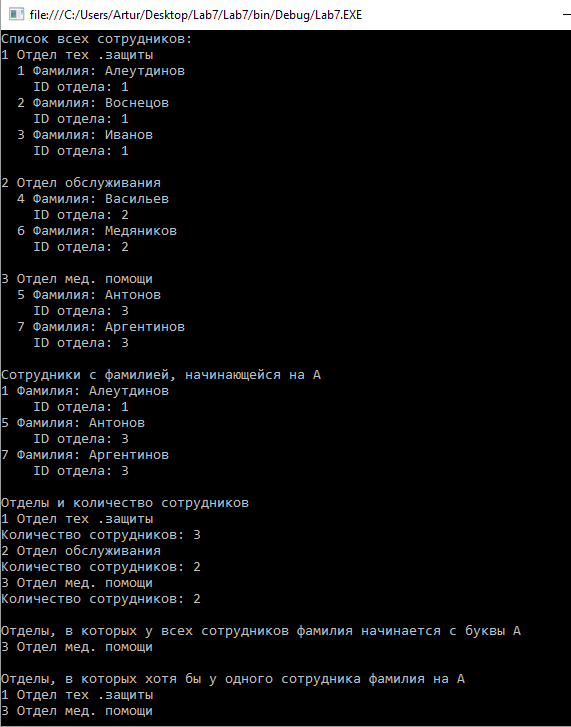
Console.ReadKey();

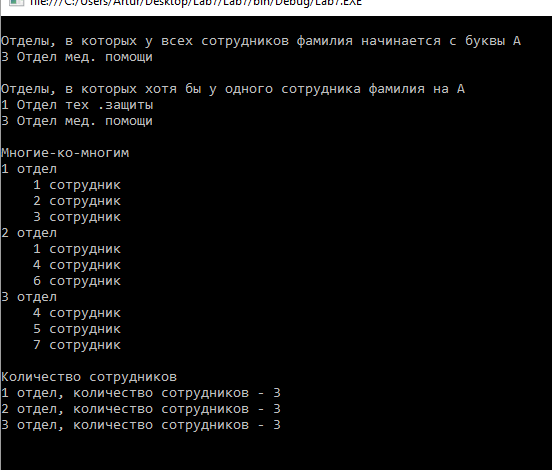
}

}

}

**Результаты выполнения программы:**





**Диаграмма классов:**

