Московский государственный технический университет имени Н.Э.Баумана

(МГТУ им. Н.Э.Баумана)

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

**«РАЗРАБОТКА ПРОГРАММЫ**

**ДЛЯ РАБОТЫ С ФАЙЛАМИ»**

**ПО ДИСЦИПЛИНЕ «БАЗОВЫЕ КОМПОНЕНТЫ ИНТЕРНЕТ-ТЕХНОЛОГИЙ»**

Выполнил(а): Хапов А.В.

студент группы ИУ5-31

Проверил: Гапанюк Ю.Е.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Москва, 2017

1. **Задание**

Разработать программу, реализующую работу с LINQ to Objects.

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.
7. **Текст программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9

{

class Program

{

public class worker

{

public int id;

public string surname;

public int officeID;

public worker(int i, string s, int o)

{

this.id = i;

this.surname = s;

this.officeID = o;

}

public override string ToString()

{

return "id=" + this.id.ToString() + "| surname=" + this.surname + "| OfficeID=" + this.officeID + "|";

}

}

public class OfficeWorker

{

public int id;

public int officeID;

public OfficeWorker(int i, int o)

{

this.id = i;

this.officeID = o;

}

public override string ToString()

{

return "workerid=" + this.id.ToString() + "| OfficeID=" + this.officeID + "|";

}

}

public class office

{

public int officeID;

public string officeName;

public office(int i, string on)

{

this.officeID = i;

this.officeName = on;

}

public override string ToString()

{

return "officeID=" + this.officeID.ToString() + "| officeName=" + this.officeName.ToString() + "|";

}

}

static List<worker> workers = new List<worker>()

{

new worker(1, "Alexandrov", 3),

new worker(2, "Alexeev ", 2),

new worker(3, "Akinfeev ", 2),

new worker(4, "Ivanov ", 3),

new worker(5, "Petrov ", 3),

new worker(6, "Vodka ", 1),

new worker(7, "Andreev ", 1),

new worker(8, "Ushakov ", 1),

new worker(9, "Lapochkin ", 3)

};

static List<office> rooms = new List<office>()

{

new office(1, "Economics "),

new office(2, "Publicity "),

new office(3, "Programmers")

};

static List<OfficeWorker> OW = new List<OfficeWorker>()

{

new OfficeWorker(1, 1),

new OfficeWorker(2, 2),

new OfficeWorker(3, 2),

new OfficeWorker(4, 3),

new OfficeWorker(5, 3),

new OfficeWorker(6, 1),

new OfficeWorker(7, 1),

new OfficeWorker(8, 1),

new OfficeWorker(5, 2),

new OfficeWorker(6, 3),

new OfficeWorker(7, 2),

new OfficeWorker(8, 3),

new OfficeWorker(9, 3),

};

static void Main(string[] args)

{

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

var q1 = from x in workers select x;

foreach (var x in q1) Console.WriteLine(x);

Console.WriteLine("Перечисление всех офисов:");

var q2 = from x in rooms select x;

foreach (var x in q2) Console.WriteLine(x);

Console.WriteLine("Cписок всех сотрудников, отсортированный по отделам");

var q3 = from x in workers

where x.officeID >= 1

orderby x.officeID 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 workers

where x.surname[0] is 'A'

orderby x.officeID ascending

select x;

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

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

var q5 = from x in rooms

join y in workers on x.officeID equals y.officeID into temp

from t in temp

select new { RoomNumber = x.officeID, RoomName = x.officeName, number = 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 workers

from y in rooms

where (x.surname[0] is 'A') & (x.officeID == y.officeID)

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 workers

join y in q4 on x.officeID equals y.officeID 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.RoomNumber == y.RoomNumber)

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 workers

join x in OW on z.officeID equals x.officeID into temp

from t1 in temp

join y in rooms on t1.officeID equals y.officeID into temp2

from t2 in temp2

select new { id = z.officeID, name = t2.officeName };

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

foreach (var x in q8\_1)

Console.WriteLine(x);

var q8\_2 = from x in workers

join l in OW on x.id equals l.id into temp

from t1 in temp

join y in workers on t1.id equals y.id into temp2

from t2 in temp2

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

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 OW

join y in workers on x.officeID equals y.officeID into temp

from t in temp

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

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

var q9\_2 = from x in workers

join ed in OW on x.id equals ed.id into temp

from t1 in temp

join y in rooms on t1.officeID equals y.officeID into temp2

from t2 in temp2

select new { name = t2.officeName, id = t2.officeID };

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);

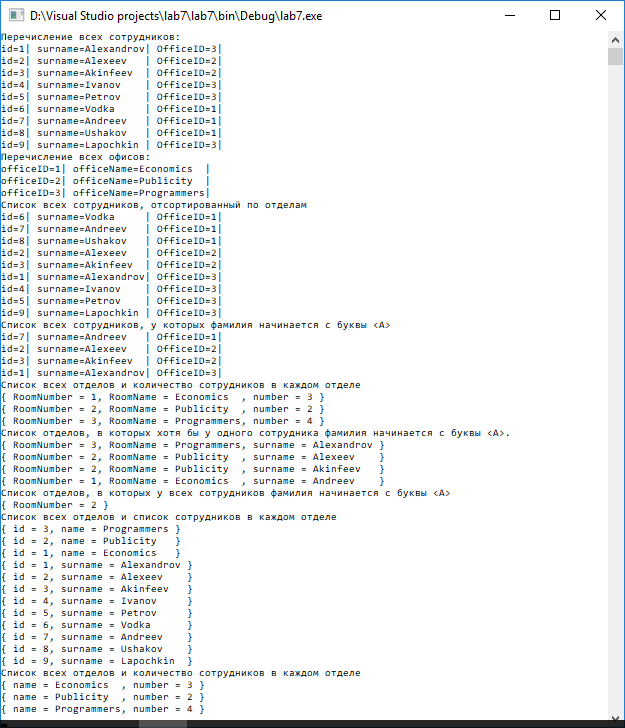
Console.ReadLine();

}

}

}

1. **Результат работы программы**



1. **Диаграмма классов**

