**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

# Кафедра ИУ5.

# Курс «Базовые компоненты интернет-технологий»

Отчет по Домашнему заданию.

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Муханов Ержан |  | Гапанюк Юрий Евгеньевич |
| Подпись и дата: |  | Подпись и дата: |

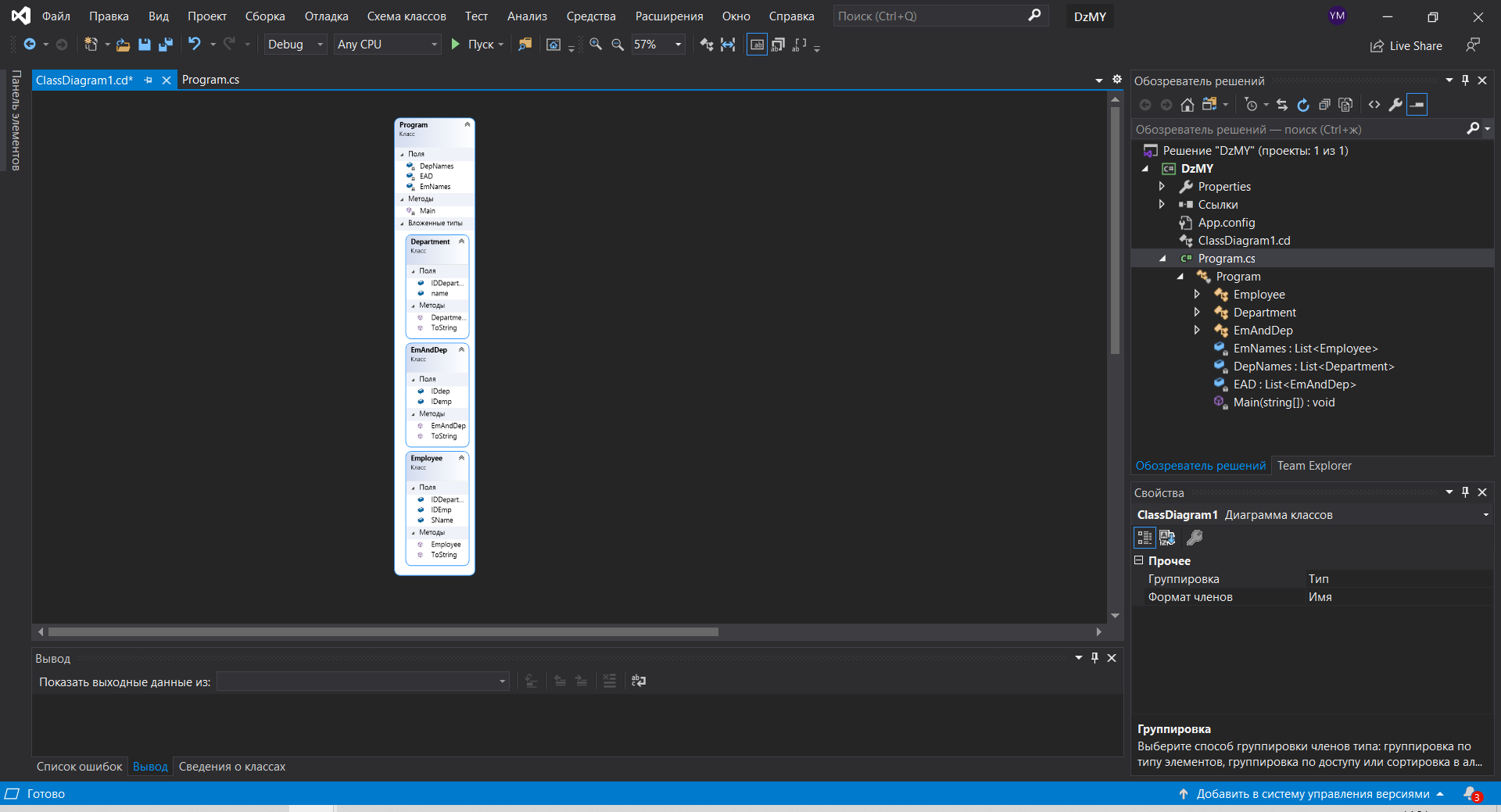
г. Москва, 2020 г.

**Постановка задачи**

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.

**Разработка интерфейса класса**



**Листинг программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DzMY

{

class Program

{

public class Employee

{

public string SName;

public int IDEmp;

public int IDDepartment;

public Employee(string name, int Emp, int Dep)

{

this.SName = name;

this.IDDepartment = Dep;

this.IDEmp = Emp;

}

public override string ToString()

{

return "Id = " + IDEmp + "; Surname: " + SName + "; ID of the Department: " + IDDepartment;

}

}

public class Department

{

public string name;

public int IDDepartment;

public Department( int ID, string name)

{

this.name = name;

this.IDDepartment = ID;

}

public override string ToString()

{

return "Id = " + IDDepartment + "; Name: " + name;

}

}

public class EmAndDep

{

public int IDemp;

public int IDdep;

public EmAndDep(int em, int dep)

{

this.IDdep = dep;

this.IDemp = em;

}

public override string ToString()

{

return "ID сотрудника: " + IDemp + "; ID отдела " + IDdep;

}

}

static List<Employee> EmNames = new List<Employee>()

{

new Employee( "Frolov", 1, 1),

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

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

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

new Employee( "Zinchenko", 5, 1),

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

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

new Employee( "Akzhol", 8, 2),

new Employee( "Shoqan", 9, 2),

new Employee( "Shevchenko", 10, 3),

new Employee( "Sayan", 11, 3),

new Employee( "Drobitko", 12, 3),

new Employee( "Alibekov", 13, 4),

new Employee( "Alekseev", 14, 4),

new Employee( "Always", 15, 4)

};

static List<Department> DepNames = new List<Department>()

{

new Department(1,"Designers"),

new Department(2, "Developers"),

new Department(3, "CEO"),

new Department(4, "Marketing")

};

static List<EmAndDep> EAD = new List<EmAndDep>()

{

new EmAndDep (1,1),

new EmAndDep (2,1),

new EmAndDep (3,1),

new EmAndDep (4,1),

new EmAndDep (5,1),

new EmAndDep (6,2),

new EmAndDep (7,2),

new EmAndDep (8,2),

new EmAndDep (9,2),

new EmAndDep (10,3),

new EmAndDep (11,3),

new EmAndDep (12,3),

new EmAndDep (13,4),

new EmAndDep (14,4),

new EmAndDep (15,4),

};

static void Main(string[] args)

{

Console.WriteLine("Список всех сотрудников, отсортированных по отделам");

var t1 = from x in EmNames

orderby x.IDDepartment

select x;

foreach (var x in t1) Console.WriteLine(x);

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine();

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine("Список сотрудников, у которых фамилия начинается с буквы \'А\'");

var t2 = from x in EmNames

where x.SName.StartsWith("A")

select x;

foreach (var x in t2) Console.WriteLine(x);

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine();

//++++++++++++++++++++++++++++++++++++++++++++++++++

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

var t3 = from y in DepNames

join x in EmNames on y.IDDepartment equals x.IDDepartment into temp

from t in temp

select new { y, count = temp.Count() };

foreach (var x in t3.Distinct()) Console.WriteLine(x);

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine();

//++++++++++++++++++++++++++++++++++++++++++++++++++

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

foreach (Department q in DepNames)

{

var t4 = from x in EmNames

where q.IDDepartment == x.IDDepartment

select x;

if ( t4.All(x => x.SName.StartsWith("A")))

{ Console.WriteLine(q + "\n"); }

}

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine("Отделы, в которых хотя бы у одного сотрудника фамилия начинается с буквы \'А\'");

var t5 = from y in DepNames

from x in EmNames

where (y.IDDepartment == x.IDDepartment && x.SName.StartsWith("A"))

select y;

foreach (var x in t5.Distinct()) Console.WriteLine(x);

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine();

//++++++++++++++++++++++++++++++++++++++++++++++++++

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

var t6 = from y in DepNames

join l in EAD on y.IDDepartment equals l.IDdep into temp

from x in EmNames

from tm1 in temp

where x.IDEmp == tm1.IDemp

select new { d = y, e = x };

var t61 = from x in t6.Union(t6)

group x by x.d.name into temp

select new { Key = temp.Key, values = temp };

foreach (var x in t61)

{

Console.WriteLine(x.Key);

foreach (var k in x.values)

Console.WriteLine("\t id = " + k.e.IDEmp + "; " + k.e.SName);

}

//++++++++++++++++++++++++++++++++++++++++++++++++++

Console.WriteLine();

//++++++++++++++++++++++++++++++++++++++++++++++++++

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

var t7 = from y in DepNames

join l in EAD on y.IDDepartment equals l.IDdep into temp

from x in EmNames

from tm1 in temp

where x.IDEmp == tm1.IDemp

select new { y, e = temp.Count() };

foreach (var x in t7.Distinct()) Console.WriteLine(x.y + "\t\t количество человек: " + x.e);

}

}

}

**Анализ результатов**

