**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «**Информатика и системы управления**»

Кафедра ИУ5. Группа 31

Отчет по лабораторной работе № 7

по предмету: «Базовые компоненты интернет-технологий»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студентка группы ИУ5-31 |  |
| Лебедева Дарья |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

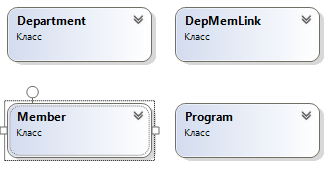
г. Москва, 2017 г.

**Лабораторная работа №7**

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**

**1)Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_7

{

class Program

{

static List<Member> memberList = new List<Member>()

{

new Member(1,"Lebedeva",1),

new Member(2,"Zorin",1),

new Member(3,"Kachanyk",1),

new Member(4,"Dmitrieva",3),

new Member(5,"Smirnov",3),

new Member(6,"Hapov",3),

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

new Member(8,"Afanasyev",2)

};

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

{

new Department(1,"Managment Department"),

new Department(2,"Bookkeeping"),

new Department(3, "Purchasing Department")

};

static List<DepMemLink> oneToMany = new List<DepMemLink>()

{

new DepMemLink(1,1),

new DepMemLink(2,1),

new DepMemLink(3,1),

new DepMemLink(4,2),

new DepMemLink(5,1),

new DepMemLink(6,2),

new DepMemLink(7,2),

new DepMemLink(4,3),

new DepMemLink(5,2),

new DepMemLink(6,1),

new DepMemLink(7,1),

new DepMemLink(8,3)

};

static void Main(string[] args)

{

for (int i = 0; i < 160; i++) Console.Write('#');

Console.WriteLine("All members which are sorted by Department ID\n");

var allMemb = from t in departmentList

join s in memberList on t.property\_1 equals s.departmentID into temp

select new { Department = t.property\_1, Member = temp };

foreach (var s in allMemb)

{

Console.WriteLine("!!!!!!!!!!!!!!DepartmentID!!!!!!!!!!!! = " + s.Department);

foreach (var y in s.Member)

Console.WriteLine(y);

}

for (int i = 0; i < 160; i++) Console.Write('#');

//=======================================================================================================================

//=======================================================================================================================

Console.WriteLine("\nAll members which surname starts at 'A'\n");

var MembFirstA = from t in memberList where t.surname.StartsWith("A") select t;

foreach (Member s in MembFirstA) Console.WriteLine(s);

for (int i = 0; i < 160; i++) Console.Write('#');

//=======================================================================================================================

//=======================================================================================================================

Console.WriteLine("\nAll departments and quantity of members\n");

var DepartAndQuantity = from a in departmentList

join b in memberList on a.property\_1 equals b.departmentID into temp

select new { Department = a, Quantity = temp.Count() };

foreach (var c in DepartAndQuantity)

{

Console.WriteLine(c.Department + "\nQuantity of members = " + c.Quantity);

}

for (int i = 0; i < 160; i++) Console.Write('#');

//=======================================================================================================================

//=======================================================================================================================

Console.WriteLine("\nAll departments, where all member's surname starts 'A' \n");

var DepartAllMembFirstA = (from s in departmentList

from t in memberList

group t by t.departmentID into g

where g.All(t => t.surname.StartsWith("A"))

select new { Department = (from s in departmentList where s.property\_1 == g.Key select s) });

foreach (var s in DepartAllMembFirstA)

{

foreach (var b in s.Department)

{

Console.WriteLine(b);

}

}

for (int i = 0; i < 160; i++) Console.Write('#');

//=======================================================================================================================

//=======================================================================================================================

Console.WriteLine("\nAll departments, where is at least one member which surname starts 'A' \n");

var DepartMembFirstA = (from s in departmentList

from t in memberList

group t by t.departmentID into g

where g.Any(t => t.surname.StartsWith("A"))

select new { Department = (from s in departmentList where s.property\_1 == g.Key select s) });

foreach (var s in DepartMembFirstA)

{

foreach (var b in s.Department)

{

Console.WriteLine(b);

}

}

for (int i = 0; i < 160; i++) Console.Write('#');

//=======================================================================================================================

//=======================================================================================================================

Console.WriteLine("\nAll departments and all members in this department \n");

var AllDepartAndMembers = (from t in memberList

join r in oneToMany on t.memberID equals r.memberID into temp

from t1 in temp

group t by t1.departmentID into g

from t in departmentList

where t.property\_1 == g.Key

select new { Members = g, department = t });

foreach (var s in AllDepartAndMembers)

{

for (int i = 0; i < 80; i++) Console.Write('\_');

Console.WriteLine(s.department);

for (int i = 0; i < 80; i++) Console.Write('\_');

foreach (var f in s.Members) Console.WriteLine(f);

}

//=======================================================================================================================

//=======================================================================================================================

Console.WriteLine("\nAll departments and quantity of members in this department \n");

var AllDepartAndQuantityOfMemb = (from t in memberList

join r in oneToMany on t.memberID equals r.memberID into temp

from t1 in temp

group t by t1.departmentID into g

from t in departmentList

where t.property\_1 == g.Key

select new { Quantity = g.Count(), department = t });

foreach (var s in AllDepartAndQuantityOfMemb) Console.WriteLine(s.department + "\nQuantity of members = " + s.Quantity);

//=======================================================================================================================

//=======================================================================================================================

Console.ReadLine();

}

}

}

**2)Members.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

namespace LAB\_7

{

public class Member : IComparable

{

public int memberID;

public string surname;

public int departmentID;

public Member(int m, string s, int d)

{

memberID = m;

surname = s;

departmentID = d;

}

public override string ToString()

{

return ("\nMember ID= " + memberID + "\nSurname= " + surname + "\nDepartment ID=" + departmentID);

}

public int CompareTo(object a)

{

Member p = (Member)a;

if (p.departmentID > this.departmentID) return -1;

else if (p.departmentID < this.departmentID) return 1;

else return 0;

}}}

**3)DepMemLink.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

namespace LAB\_7

{

class DepMemLink

{

public int memberID;

public int departmentID;

public DepMemLink(int mID, int dID)

{

this.memberID = mID;

this.departmentID = dID;

}

}

}

**4)Department.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

namespace LAB\_7

{

class Department

{

int departmentID;

string NameOfDepartment;

public Department(int id, string name)

{

this.departmentID = id;

this.NameOfDepartment = name;

}

public int property\_1

{

get { return this.departmentID; }

set { }

}

public override string ToString()

{

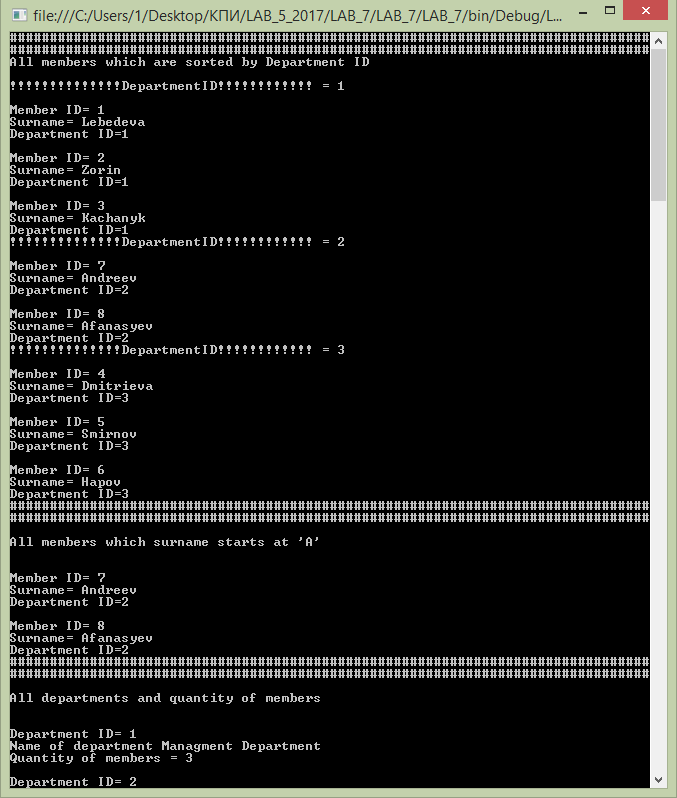
return ("\nDepartment ID= " + departmentID + "\nName of department " + NameOfDepartment);

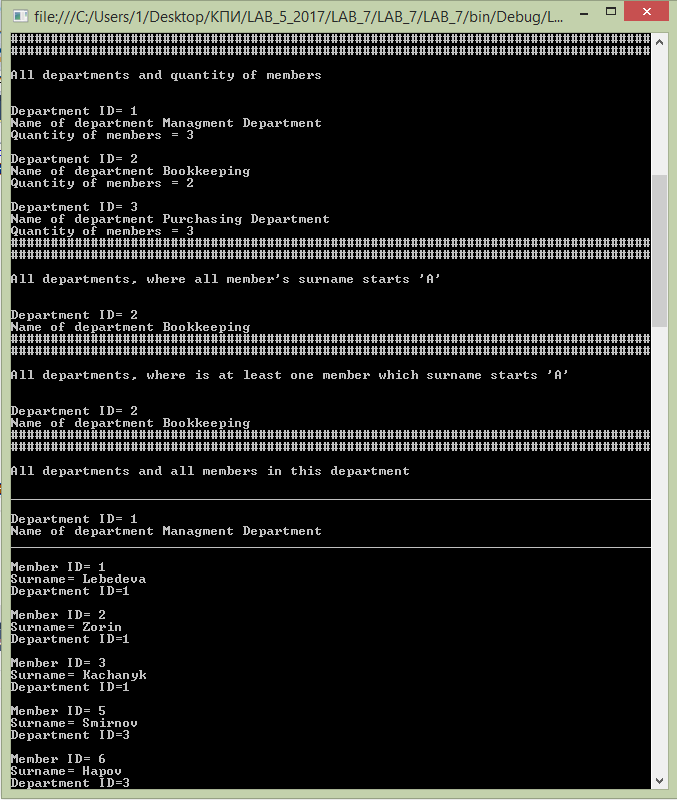
}

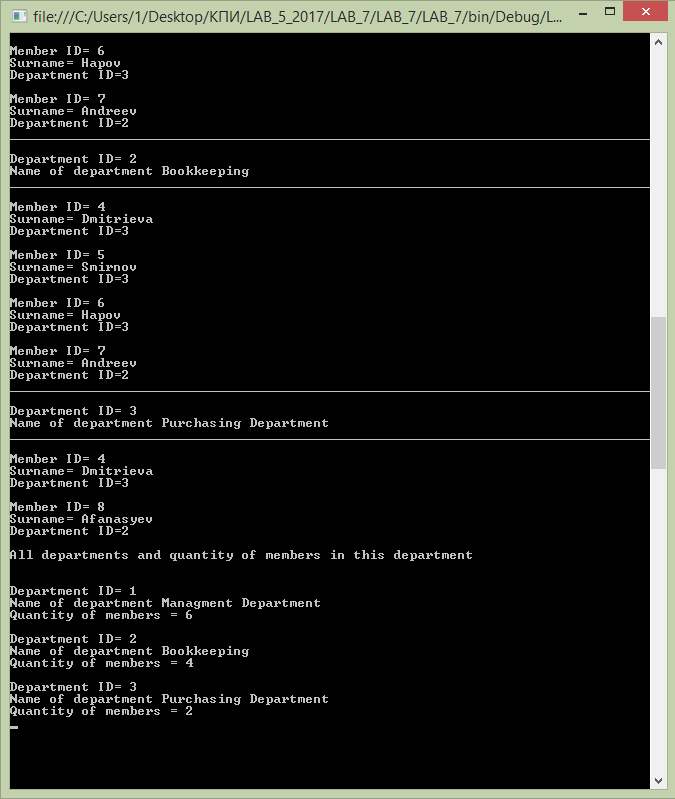
}

}

**Экранные формы**

****

****

****