**Отчет по лабораторной работе №7 по курсу С#**

**8**

**(количество листов)**

Студентка группы ИУ5-32

Губайдуллина Карина

Дата: 21.11.2017

Руководитель:

Гапанюк Ю.Е.

Подпись:

Дата:

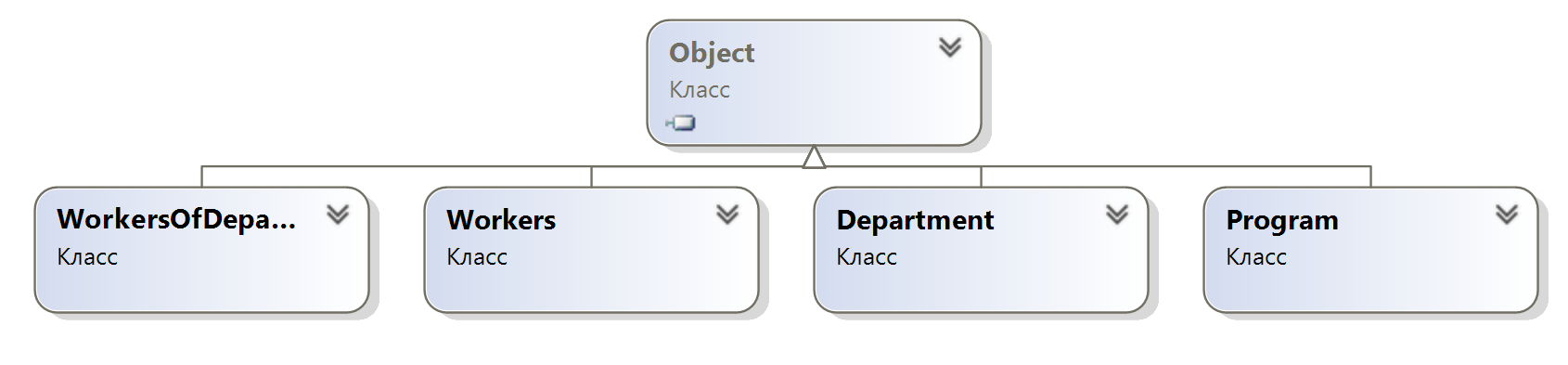
**Задание:**

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:
   1. ID записи о сотруднике;
   2. Фамилия сотрудника;
   3. ID записи об отделе.
3. Создайте класс «Отдел», содержащий поля:
   1. ID записи об отделе;
   2. Наименование отдела.
4. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением один-ко-многим разработайте следующие запросы:
   1. Выведите список всех сотрудников и отделов, отсортированный по отделам.
   2. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А».
   3. Выведите список всех отделов и количество сотрудников в каждом отделе.
   4. Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы «А».
   5. Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А».
5. Создайте класс «Сотрудники отдела», содержащий поля:
   1. ID записи о сотруднике;
   2. ID записи об отделе.
6. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением много-ко-многим с использованием класса «Сотрудники отдела» разработайте следующие запросы:
   1. Выведите список всех отделов и список сотрудников в каждом отделе.

Выведите список всех отделов и количество сотрудников в каждом отделе.

**Диаграмма классов:**



**Текст программы:**

***Program.cs***

using System;

using System.Collections.Generic;

using System.Linq;

namespace lab\_7

{

class Program

{

static List<Workers> staff = new List<Workers>

{

new Workers(1,"Иванов", 1),

new Workers(2,"Абрамов", 2),

new Workers(3,"Кузнецова", 3),

new Workers(4,"Козлова", 4),

new Workers(5,"Александрова", 5),

new Workers(6,"Жуков", 6),

new Workers(7,"Филиппов", 7),

new Workers(8,"Орехов", 8),

new Workers(9,"Горшкова", 9),

new Workers(10,"Петухов", 1),

new Workers(11,"Ильин", 2),

new Workers(12,"Калашникова", 3),

new Workers(13,"Князев", 4),

};

static List<Department> dep = new List<Department>

{

new Department(1, "Снабжение"),

new Department(2, "Отдел кадров"),

new Department(3, "Бухгалтерия"),

new Department(4, "Диспетчерский"),

new Department(5, "Маркетинговый "),

new Department(6, "Финансовый"),

new Department(7, "Закупки"),

new Department(8, "Управление"),

new Department(9, "Охрана"),

};

static List<WorkersOfDepartments> link = new List<WorkersOfDepartments>

{

new WorkersOfDepartments(1,1),

new WorkersOfDepartments(2,2),

new WorkersOfDepartments(3,3),

new WorkersOfDepartments(4,4),

new WorkersOfDepartments(5,5),

new WorkersOfDepartments(6,6),

new WorkersOfDepartments(7,7),

new WorkersOfDepartments(8,8),

new WorkersOfDepartments(9,9),

new WorkersOfDepartments(10,1),

new WorkersOfDepartments(11,2),

new WorkersOfDepartments(12,3),

new WorkersOfDepartments(13,4),

};

static void Main(string[] args)

{

// ПУНКТ 44444444444

Console.WriteLine("список всех сотрудников и отделов, отсортированный по отделам");

var selectStuff = from s in staff

join t in dep on s.dep\_id equals t.id

orderby t.id

select new { Id = s.id, Name = s.name, IdDep = t.id, NameDep = t.name };

foreach (var t in selectStuff)

Console.WriteLine("ID сотрудника : {0}, Фамилия : {1}, Отдел : {2}", t.Id, t.Name, t.NameDep);

Console.WriteLine("\n\nсписок всех сотрудников, у которых фамилия начинается с буквы «А»");

var selectStuff2 = from t in staff

where t.name.ToUpper().StartsWith("А")

orderby t.dep\_id

select t;

foreach (var t in selectStuff2)

Console.WriteLine(t);

Console.WriteLine("\n\nсписок всех отделов и количество сотрудников в каждом отделе");

var selectStuff3 = from t in selectStuff

group t by t.NameDep into depGroup

select new

{

Dep = depGroup.Key,

Count = depGroup.Count()

};

foreach (var t in selectStuff3)

Console.WriteLine("Название отдела : {0}, Количество сотрудников: {1}", t.Dep, t.Count);

Console.WriteLine("\n\nсписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А»");

var selectStuff4 = from t in selectStuff

where t.Name.ToUpper().StartsWith("А")

orderby t.IdDep

select t;

foreach (var t in selectStuff4)

Console.WriteLine("{1} : {0}", t.NameDep, t.IdDep);

Console.WriteLine("\n\nсписок отделов, в которых у всех сотрудников фамилия начинается с буквы «А»");

var selectStuff5 = from t in selectStuff

group t by t.NameDep into depGroup

where depGroup.All(t => t.Name.ToUpper().StartsWith("А"))

select new

{

Dep = depGroup.Key

};

foreach (var t in selectStuff5)

Console.WriteLine("{0}", t.Dep);

// ПУНКТ 6666666666666

Console.WriteLine("\n\n\n\nсписок всех отделов и список сотрудников в каждом отделе");

var link1 = from x in staff

join l in link on x.id equals l.w into temp

from t1 in temp

join y in dep on t1.d equals y.id into temp2

from t2 in temp2

group t2 by t2.id into office

select new

{

ID = office.Key,

Count = office.Count(),

Worker = from w in staff

where w.dep\_id == office.Key

select w

};

foreach (var off in link1)

{

Console.WriteLine("{0} : {1}", off.ID, off.Count);

foreach (Workers worker in off.Worker)

Console.WriteLine(worker.name);

}

}

}

}

***Workers.cs***

namespace lab\_7

{

class Workers

{

public int id;

public string name;

public int dep\_id;

public Workers(int i, string n, int d)

{

this.id = i;

this.name = n;

this.dep\_id = d;

}

public override string ToString()

{

return "id=" + this.id.ToString() + "; фамилия : " + this.name.ToString() + "; отдел : " + this.dep\_id.ToString();

}

}

}

***Department.cs***

namespace lab\_7

{

class Department

{

public int id;

public string name;

public Department(int i, string n)

{

this.id = i;

this.name = n;

}

public override string ToString()

{

return "id=" + this.id.ToString() + "; наименование отдела : " + this.name.ToString();

}

}

}

***WorkersOfDepartments.cs***

namespace lab\_7

{

class WorkersOfDepartments

{

public int w;

public int d;

public WorkersOfDepartments(int i1, int i2)

{

this.w = i1;

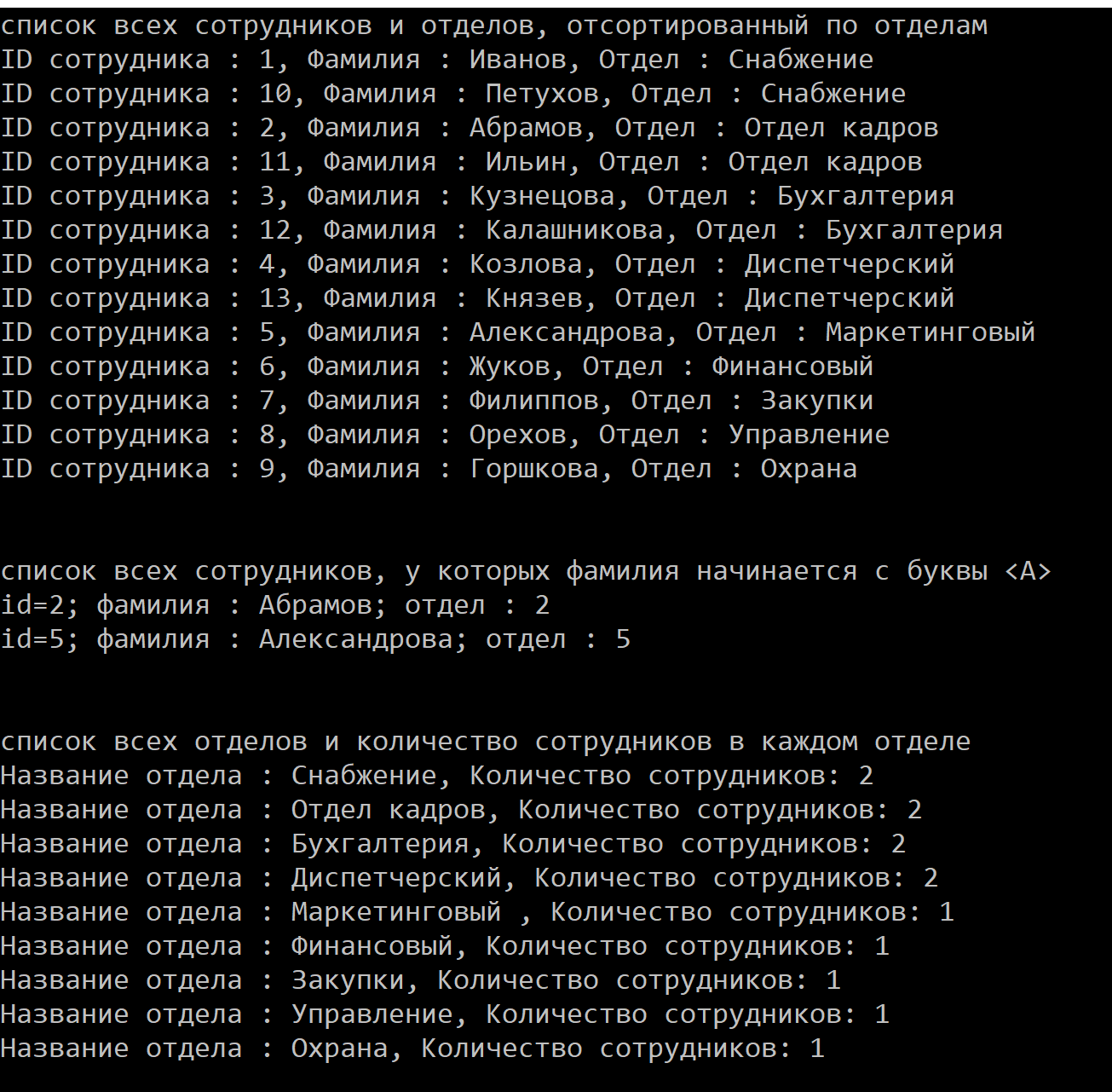
this.d = i2;

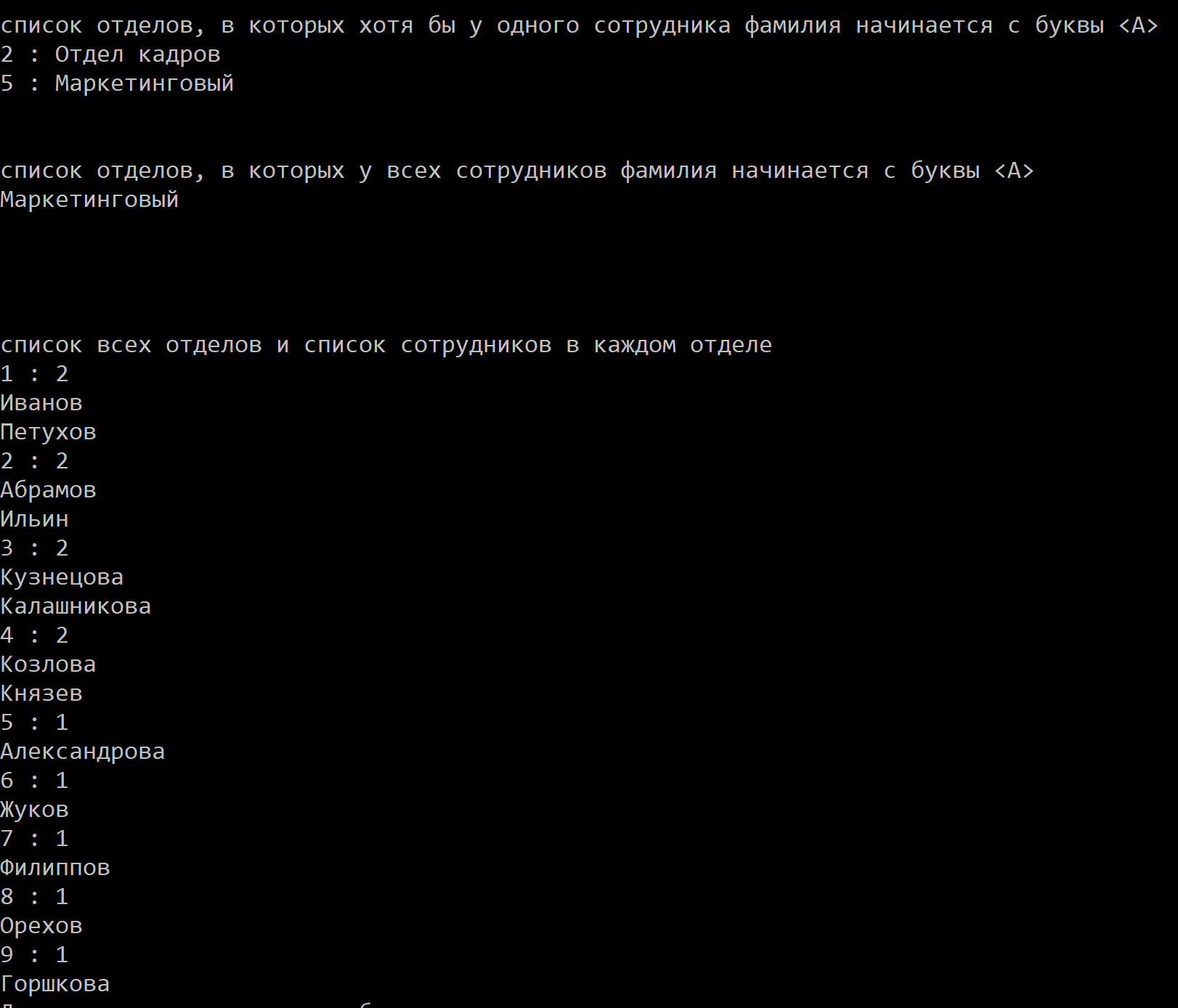
}

}

}

Результаты:





|  |
| --- |
|  |