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

7

(количество листов)

Студент группы ИУ5-34:

Тюлькина Нина

Подпись Дата

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Руководитель: Гапанюк Ю.Е.

Подпись Дата

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Москва, 2017

*Условие 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».

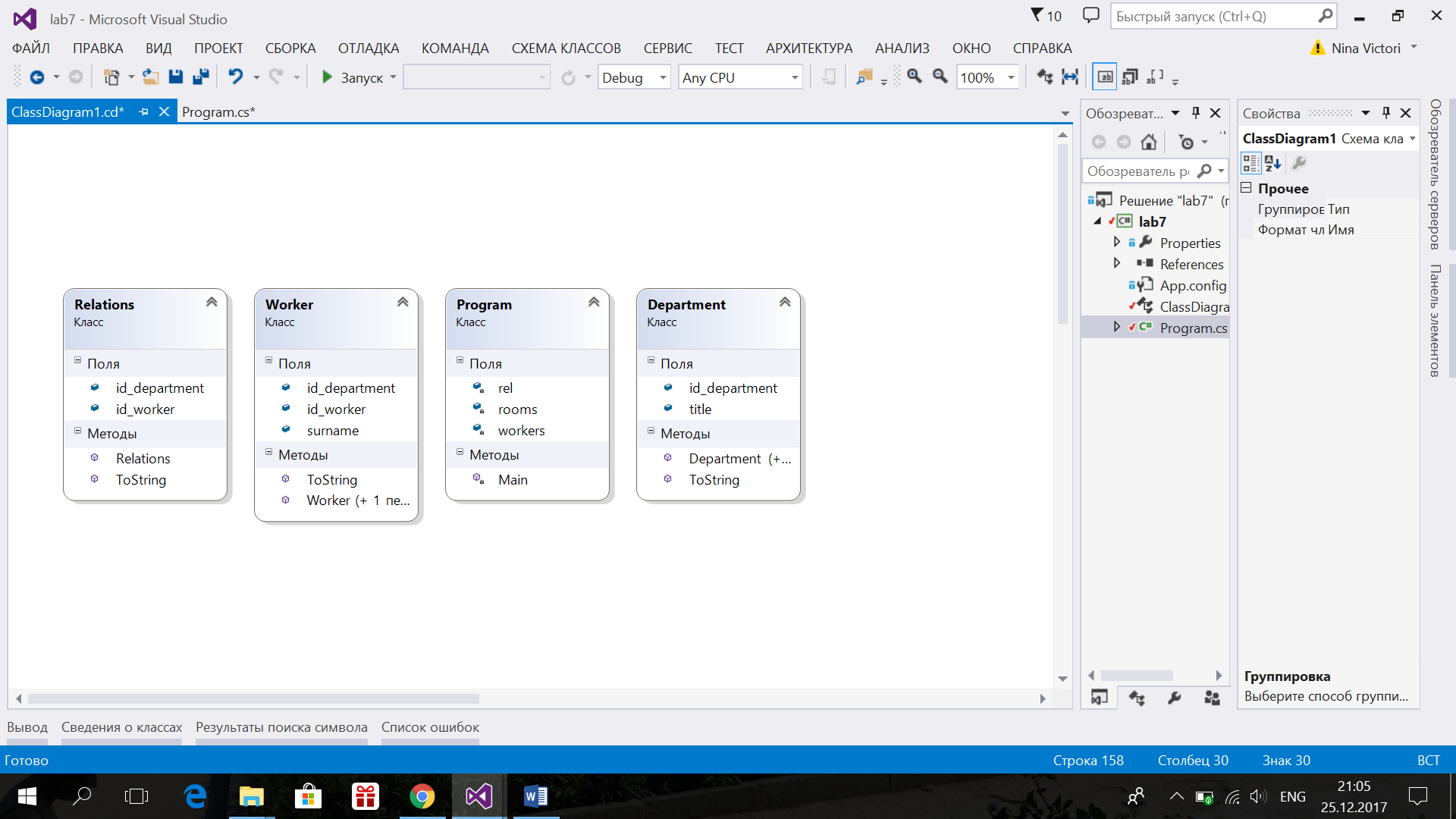
5. Создайте класс «Сотрудники отдела», содержащий поля:

* ID записи о сотруднике;
* ID записи об отделе.

6. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением 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*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7

{

class Program

{

static List<office.Worker> workers = new List<office.Worker>()

{

new office.Worker(1, "Лескина ", 1),

new office.Worker(2, "Брысина ", 4),

new office.Worker(3, "Байбарин ", 2),

new office.Worker(4, "Гаврилюк ", 3),

new office.Worker(5, "Баскакова ", 3),

new office.Worker(6, "Кондрашева ", 4),

new office.Worker(7, "Тимаков ", 1),

new office.Worker(8, "Авдеев ", 2),

new office.Worker(9, "Александрова ", 2),

};

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

{

new office.Department(1, "Отдел финансов "),

new office.Department(2, "Общий отдел "),

new office.Department(3, "Отдел кадров "),

new office.Department(4, "IT отдел ")

};

static List<office.Relations> rel = new List<office.Relations>()

{

new office.Relations(1, 1),

new office.Relations(2, 4),

new office.Relations(3, 2),

new office.Relations(4, 3),

new office.Relations(5, 3),

new office.Relations(6, 4),

new office.Relations(7, 1),

new office.Relations(8, 2),

new office.Relations(9, 2),

new office.Relations(6, 3),

new office.Relations(7, 2),

new office.Relations(8, 4),

new office.Relations(9, 1)

};

static void Main(string[] args)

{

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

var i1 = from x in workers select x;

foreach (var x in i1) Console.WriteLine(x);

Console.WriteLine("\nПеречисление всех офисов:");

var i2 = from x in rooms select x;

foreach (var x in i2) Console.WriteLine(x);

Console.WriteLine("\nCписок всех сотрудников, отсортированный по отделам:");

foreach (var r in rooms)

{

var i3 = from x in workers where r.id\_department == x.id\_department select x;

foreach (var x in i3) Console.WriteLine(x);

}

Console.WriteLine("\nCписок всех сотрудников, фамилия которых начинается на А:");

foreach (var x in workers)

if (x.surname[0] == 'А') Console.WriteLine(x);

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

foreach (var x in rooms)

{

int num = workers.Count(y => y.id\_department == x.id\_department);

Console.WriteLine(x + "Количество=" + num);

}

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

var i4 = from x in rooms

where (workers.Count(y => y.surname[0] == 'А' && y.id\_department == x.id\_department) == workers.Count(y => y.id\_department == x.id\_department))

select x;

foreach (var x in i4) Console.WriteLine(x);

if (i4.Count() == 0)

{

Console.WriteLine("Таких отделов нет");

}

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

var i5 = from x in rooms

where (workers.Count(y => y.surname[0] == 'А' && y.id\_department == x.id\_department) > 0)

select x;

foreach (var x in i5) Console.WriteLine(x);

Console.WriteLine("\nДля связи Отдел и Сотрудник М:М:");

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

foreach (var x in rooms)

{

//Перебор по связям отдел-сотрудник

var i6 = from y in rel

where (y.id\_department == x.id\_department)

select y;

//Перебор по списку сотрудников

var i7 = from y in workers

from z in i6

where (z.id\_worker == y.id\_worker)

select y;

Console.WriteLine(x);

foreach (var y in i7) Console.WriteLine(y);

}

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

foreach (var x in rooms)

{

//Перебор по связям отдел-работник

var i8 = from y in rel

where (y.id\_department == x.id\_department)

select y;

Console.WriteLine(x + " " + i8.Count());

}

Console.ReadLine();

}

}

}

namespace office

{

class Worker

{

public int id\_worker;

public string surname;

public int id\_department;

public Worker() {}

public Worker(int a, string s, int b)

{

this.id\_worker = a;

this.surname = s;

this.id\_department = b;

}

public override string ToString()

{

StringBuilder b = new StringBuilder();

return "id\_worker=" + this.id\_worker.ToString() + "| surname=" + this.surname + b.Append(Convert.ToChar(32), 20 - this.surname.Length )

+ "| id\_department= " + this.id\_department + "|";

}

}

class Department

{

public int id\_department;

public string title;

public Department() { }

public Department(int a, string s)

{

this.id\_department = a;

this.title = s;

}

public override string ToString()

{

StringBuilder b = new StringBuilder();

return "id\_department=" + this.id\_department.ToString() + "| title=" + this.title.ToString() + b.Append(Convert.ToChar(32), 15 - this.title.Length) + "|";

}

}

class Relations {

public int id\_worker;

public int id\_department;

public Relations(int a, int b)

{

this.id\_worker = a;

this.id\_department = b;

}

public override string ToString()

{

return "workerid=" + this.id\_worker.ToString() + "| OfficeID=" + this.id\_department + "|";

}

}

}

*Вывод*

