**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №7

«Трек курса «Функциональное программирование»»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студентка группы ИУ5-33 |  | преподаватель каф. ИУ5 |
| А.Д. Рыбина |  | Ю.Е. Гапанюк. |
| Подпись и дата: |  | Подпись и дата: |

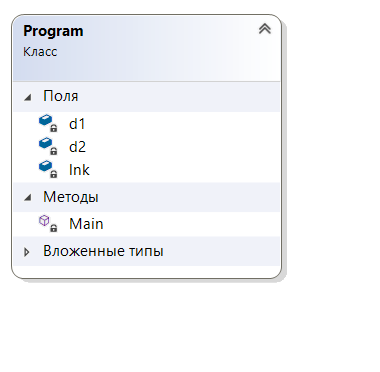
Москва, 2020 г.

**Описание задания**

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;

namespace Lab7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Cортировка по отделу:");

Console.WriteLine("Cross Join (Inner Join) с использованием Join");

var q8 = from x in d2

join y in d1 on x.ID\_записи\_об\_отделе equals y.ID\_записи\_об\_отделе

orderby x.Наименование\_отдела ascending

select new { v1 = x.Наименование\_отдела, v2 = y.Фамилия\_сотрудника };

foreach (var x in q8) Console.WriteLine(x);

//Фамилия с буквы "А"

Console.WriteLine("\nФамилия с буквы А:");

foreach (var x in q8)

{

string z = x.v2;

z = z.Remove(1);

string a = "А";

int res = String.Compare(z, a);

if (res == 0) Console.WriteLine(x);

}

//количестов сотрудников

Console.WriteLine("\nКоличество сотрудников:");

var q11 = from x in d2

join y in d1 on x.ID\_записи\_об\_отделе equals y.ID\_записи\_об\_отделе into temp

from t in temp

select new { v1 = x.Наименование\_отдела, cnt = temp.Count() };

foreach (var x in q11) Console.WriteLine(x);

//отдел все фамилии на А

Console.WriteLine("\nОтдел все фамилии на А:");

var q12 = from x in d2

join y in d1 on x.ID\_записи\_об\_отделе equals y.ID\_записи\_об\_отделе into temp

select new { v1 = x.Наименование\_отдела, d2Group = temp };

foreach (var x in q12)

{

bool flag = true;

int k = 0;

foreach (var y in x.d2Group)

{

k++;

string отд = y.Фамилия\_сотрудника;

string z = y.Фамилия\_сотрудника;

z = z.Remove(1);

string a = "А";

int res = String.Compare(z, a);

if (res != 0) flag = false;

}

if (flag == true && k > 0)

Console.WriteLine(x.v1);

}

//отдел хоть одна фамилия на А

Console.WriteLine("\nОтдел хоть одна фамилия на А:");

foreach (var x in q12)

{

int k = 0;

foreach (var y in x.d2Group)

{

string отд = y.Фамилия\_сотрудника;

string z = y.Фамилия\_сотрудника;

z = z.Remove(1);

string a = "А";

int res = String.Compare(z, a);

if (res == 0) k++;

}

if (k > 0)

Console.WriteLine(x.v1);

}

Console.WriteLine("\nИмитация связи много-ко-многим:");

var lnk1 = from x in d1

join l in lnk on x.ID\_записи\_о\_сотруднике equals l.ID\_записи\_о\_сотруднике into temp

from t1 in temp

join y in d2 on t1.ID\_записи\_об\_отделе equals y.ID\_записи\_об\_отделе into temp2

from t2 in temp2

select new { id2 = t2.Наименование\_отдела, id1 = x.Фамилия\_сотрудника };

foreach (var x in lnk1) Console.WriteLine(x);

}

public class Сотрудник

{

public int ID\_записи\_о\_сотруднике;

public string Фамилия\_сотрудника;

public int ID\_записи\_об\_отделе;

public Сотрудник(int i, string g, int v)

{

this.ID\_записи\_о\_сотруднике = i;

this.Фамилия\_сотрудника = g;

this.ID\_записи\_об\_отделе = v;

}

public override string ToString()

{

return "(id=" + this.ID\_записи\_о\_сотруднике.ToString() + "; Фамилия=" + this.Фамилия\_сотрудника + "; ID записи об отделе=" + this.ID\_записи\_об\_отделе + ")";

}

}

public class Отдел

{

public int ID\_записи\_об\_отделе;

public string Наименование\_отдела;

public Отдел(int i, string g)

{

this.ID\_записи\_об\_отделе = i;

this.Наименование\_отдела = g;

}

public override string ToString()

{

return "(id Отдел=" + this.ID\_записи\_об\_отделе.ToString() + "; Наименование отдела=" + this.Наименование\_отдела + ")";

}

}

public class Сотрудник\_отдела

{

public int ID\_записи\_о\_сотруднике;

public int ID\_записи\_об\_отделе;

public Сотрудник\_отдела(int i, int g)

{

this.ID\_записи\_о\_сотруднике = i;

this.ID\_записи\_об\_отделе = g;

}

public override string ToString()

{

return "(id Отдел=" + this.ID\_записи\_об\_отделе.ToString() + "; Наименование отдела=" + this.ID\_записи\_о\_сотруднике + ")";

}

}

//Связь между списками

//Пример данных

static List<Сотрудник> d1 = new List<Сотрудник>()

{

new Сотрудник(1, "Рыбина", 3),

new Сотрудник(3, "Афанасьева", 8),

new Сотрудник(5, "Мишура", 5),

new Сотрудник(8, "Алантьев", 3),

new Сотрудник(6, "Алин", 8),

new Сотрудник(6, "Аллигрова", 8)

};

static List<Отдел> d2 = new List<Отдел>()

{

new Отдел(3, "Бухгалтерия"),

new Отдел(8, "Администрация"),

new Отдел(5, "Отдел кадров")

};

static List<Сотрудник\_отдела> lnk = new List<Сотрудник\_отдела>()

{

new Сотрудник\_отдела(1,3),

new Сотрудник\_отдела(1,8),

new Сотрудник\_отдела(1,5),

new Сотрудник\_отдела(3,1),

new Сотрудник\_отдела(3,3),

new Сотрудник\_отдела(3,5),

new Сотрудник\_отдела(5,3),

new Сотрудник\_отдела(5,8),

new Сотрудник\_отдела(5,5),

new Сотрудник\_отдела(6,3),

new Сотрудник\_отдела(6,8),

new Сотрудник\_отдела(6,5),

new Сотрудник\_отдела(8,3),

new Сотрудник\_отдела(8,8),

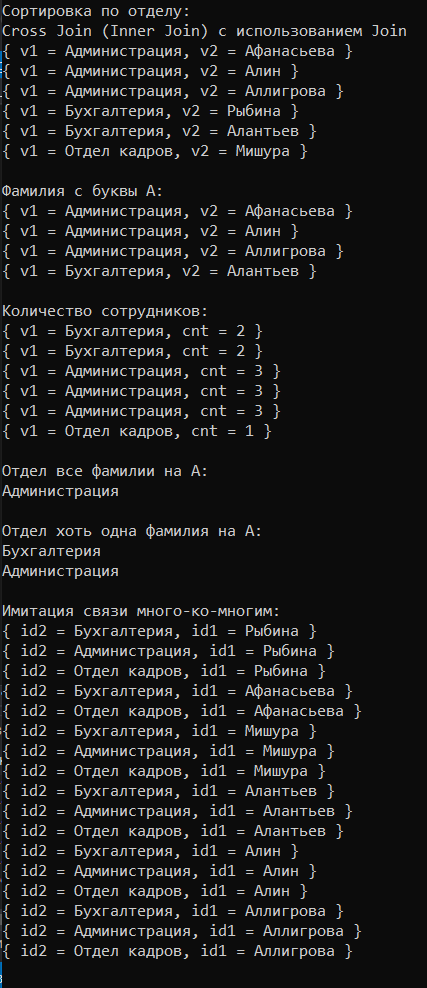
new Сотрудник\_отдела(8,5),

};

}

}

**Экранная формы с примерами выполнения программы**

****