**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №7

«Трек курса «Задания на основе языка C#»»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  | преподаватель каф. ИУ5 |
| Драгун И.А. |  | Ю.Е. Гапанюк. |
| Подпись и дата:  28.12.2020 |  | Подпись и дата:  28.12.2020 |

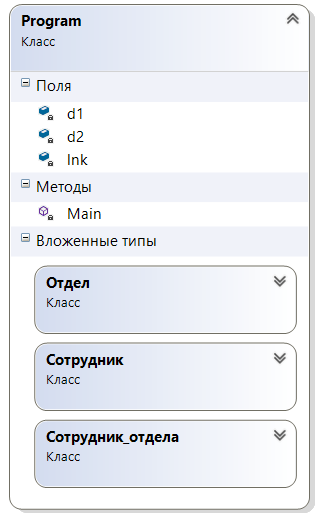
Москва, 2020 г.

**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\_0

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Cортировка по отделу");

var q8 = from x in d2

join y in d1 on x.idDepartmentRecord equals y.idDepartmentRecord

orderby x.departmentName ascending

select new { v1 = x.departmentName, v2 = y.EmployeeSurname };

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

//Фамилия с буквы "А"

Console.WriteLine("Фамилия 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("Количество сотрудников");

var q11 = from x in d2

join y in d1 on x.idDepartmentRecord equals y.idDepartmentRecord into temp

from t in temp

select new { v1 = x.departmentName, cnt = temp.Count() };

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

//отдел все фамилии на А

Console.WriteLine("отдел все фамилии на А");

var q12 = from x in d2

join y in d1 on x.idDepartmentRecord equals y.idDepartmentRecord into temp

select new { v1 = x.departmentName, d2Group = temp };

foreach (var x in q12)

{

bool flag = true;

int k = 0;

foreach (var y in x.d2Group)

{

k++;

//string отд = y.EmployeeSurname;

string z = y.EmployeeSurname;

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("отдел хоть одна фамилия на А");

foreach (var x in q12)

{

int k = 0;

foreach (var y in x.d2Group)

{

string отд = y.EmployeeSurname;

string z = y.EmployeeSurname;

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("Cписок всех отделов и список сотрудников в каждом отделе через много-ко-многим");

var lnk1 = from x in d1

join l in lnk on x.idEmployeeRecord equals l.idEmployeeRecord into temp

from t1 in temp

join y in d2 on t1.idDepartmentRecord equals y.idDepartmentRecord into temp2

from t2 in temp2

select new { id2 = t2.departmentName, id1 = x.EmployeeSurname };

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

Console.ReadKey();

}

public class Сотрудник

{

public int idEmployeeRecord;

public string EmployeeSurname;

public int idDepartmentRecord;

public Сотрудник(int id, string surname, int idDR)

{

this.idEmployeeRecord = id;

this.EmployeeSurname = surname;

this.idDepartmentRecord = idDR;

}

public override string ToString()

{

return "(id=" + this.idEmployeeRecord.ToString() + "; Фамилия=" + this.EmployeeSurname + "; ID записи об отделе=" + this.idDepartmentRecord + ")";

}

}

public class Отдел

{

public int idDepartmentRecord;

public string departmentName;

public Отдел(int id, string dN)

{

this.idDepartmentRecord = id;

this.departmentName = dN;

}

public override string ToString()

{

return "(id Отдел=" + this.idDepartmentRecord.ToString() + "; Наименование отдела=" + this.departmentName + ")";

}

}

public class Сотрудник\_отдела

{

public int idEmployeeRecord;

public int idDepartmentRecord;

public Сотрудник\_отдела(int idER, int idDR)

{

this.idEmployeeRecord = idER;

this.idDepartmentRecord = idDR;

}

public override string ToString()

{

return "(id Отдел=" + this.idDepartmentRecord.ToString() + "; id Сотрудника=" + this.idEmployeeRecord + ")";

}

}

//Связь между списками

//Пример данных

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

{

new Сотрудник(1, "Иванов", 1),

new Сотрудник(2, "Петров", 1),

new Сотрудник(3, "Кузьмин", 2),

new Сотрудник(4, "Киреев", 2),

new Сотрудник(5, "Кузнецов", 2),

new Сотрудник(6, "Андреев", 3),

new Сотрудник(7, "Антонов", 3),

};

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

{

new Отдел(1, "Программисты"),

new Отдел(2, "Инженеры"),

new Отдел(3, "Проверяющие")

};

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

{

new Сотрудник\_отдела(1,1),

new Сотрудник\_отдела(2,1),

new Сотрудник\_отдела(3,2),

new Сотрудник\_отдела(4,1),

new Сотрудник\_отдела(5,2),

new Сотрудник\_отдела(6,3),

new Сотрудник\_отдела(7,3),

};

}

}

**Экранные формы с примерами выполнения программы**

