**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

«Работа с LINQ»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студентка группы ИУ5-33 |  | преподаватель каф. ИУ5 |
| Беспалова У.А. |  | Гапанюк Ю.Е. |
| Подпись и дата:  12.12.2018 |  | Подпись и дата: |

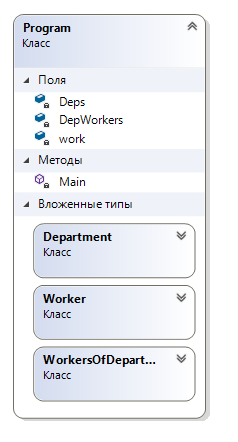
Москва, 2018 г.

1. Содержание 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:
   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.
   2. 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 лаб7LinQ

{

class Program

{

public class Worker

{

public int workerID;

public string surname;

public int depID;

public Worker (int idd, string Sur, int dep)

{

workerID = idd;

surname = Sur;

depID = dep;

}

public override string ToString() //превращение результата в строку

{

return "(ID Сотрудника = " + this.workerID.ToString() + "; Фамилия = " + this.surname + "; Отдел " + this.depID + ")";

}

}

public class Department

{

public string depName;

public int depID;

public Department(int idd, string name)

{

depName = name;

depID = idd;

}

}

public class WorkersOfDepartment

{

public int workweID;

public int depID;

public WorkersOfDepartment(int i1, int i2)

{

workweID = i1;

depID = i2;

}

}

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

{

new Department (1, "Отдел Линейного Счастья"),

new Department (2, "Отдел Недоступных Проблем"),

new Department (3, "Отдел Смысла Жизни"),

new Department (4, "Отдел Философской Физики")

};

static List<Worker> work = new List<Worker>()

{

new Worker (1, "Киврин", 1),

new Worker (2, "Амперян", 1),

new Worker (3, "Корнеев", 1),

new Worker (4, "Ойра-Ойра", 2),

new Worker (5, "Привалов", 2),

new Worker (6, "Хунта", 3),

new Worker (7, "Выбегалло", 3),

new Worker (8, "Дрозд", 3),

new Worker (9, "Король", 4),

new Worker (10, "Кукушкин", 4),

new Worker (11, "Капель", 4),

};

static List<WorkersOfDepartment> DepWorkers = new List<WorkersOfDepartment> {

new WorkersOfDepartment(1, 1),

new WorkersOfDepartment(1, 2),

new WorkersOfDepartment(2, 1),

new WorkersOfDepartment(3, 1),

new WorkersOfDepartment(3, 3),

new WorkersOfDepartment(4, 2),

new WorkersOfDepartment(4, 3),

new WorkersOfDepartment(5, 1),

new WorkersOfDepartment(5, 3),

new WorkersOfDepartment(5, 4),

new WorkersOfDepartment(6, 2),

new WorkersOfDepartment(7, 2),

new WorkersOfDepartment(7, 3),

new WorkersOfDepartment(7, 4),

new WorkersOfDepartment(8, 3),

new WorkersOfDepartment(8, 4),

new WorkersOfDepartment(9, 3),

new WorkersOfDepartment(10, 4),

new WorkersOfDepartment(11, 1),

new WorkersOfDepartment(11, 2),

new WorkersOfDepartment(11, 3),

new WorkersOfDepartment(11, 4) };

static void Main(string[] args)

{

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

var qd = from x in Deps

join y in work on x.depID equals y.depID into g

orderby x.depName

select new { Dep = x.depName, wor = g.OrderBy(g => g.surname) };

foreach (var x in qd)

{

Console.WriteLine(x.Dep );

foreach (var y in x.wor)

{

Console.WriteLine(" - " + y.surname);

}

}

Console.WriteLine("\nСписок сотрудников c фамилиями на букву 'К' ");

var q1 = from ivan in work

where ivan.surname.StartsWith("К")

orderby ivan.surname

select ivan.surname;

foreach (var x in q1) Console.WriteLine(x);

Console.WriteLine("\nЧисло сотрудников в отделах:");

foreach (var x in qd)

{

Console.WriteLine(x.Dep + " - " + x.wor.Count());

}

Console.WriteLine("\nОтделы, в которых фамилии всех сотрудников начинаются на букву 'К'");

var q2 = from x in Deps

join y in work on x.depID equals y.depID into g

where g.All(g => g.surname.StartsWith("К"))

select new { Dep = x.depName, wor = g };

foreach (var x in q2)

{

Console.WriteLine(x.Dep);

foreach (var y in x.wor)

{

Console.WriteLine(" - " + y.surname);

}

}

Console.WriteLine("\nОтделы, в которых есть сотрудники с фамилией на букву 'K'");

var q3 = from x in Deps

join y in work on x.depID equals y.depID into g

where g.Any(g => g.surname.StartsWith("К"))

select new { Dep = x.depName, wor = g };

foreach (var x in q3)

{

Console.WriteLine(x.Dep);

foreach (var y in x.wor)

{

Console.WriteLine(" - " + y.surname);

}

}

Console.WriteLine("\n\n\nОтделы и сотрудники\n");

var q4 = from x in Deps

join y in DepWorkers on x.depID equals y.depID into lst1

from l1 in lst1

join z in work on l1.workweID equals z.workerID into lst2

from l2 in lst2

orderby l1.depID, l2.surname

group l2 by x.depName into g

select g;

foreach (var x in q4)

{

Console.WriteLine(x.Key);

foreach (var y in x)

{

Console.WriteLine(" - "+y.surname);

}

}

Console.WriteLine("\nКоличество сотрудников в отделах:\n");

var q5 = from x in Deps

join y in DepWorkers on x.depID equals y.depID into lst1

from l1 in lst1

join z in work on l1.workweID equals z.workerID into lst2

from l2 in lst2

orderby l1.depID, l2.surname

group l2 by x.depName into g

select new { Dep = g.Key, wor = g.Count() };

foreach (var x in q5)

{

Console.WriteLine(x.Dep + " - " + x.wor);

}

Console.ReadLine();

}

}

}

Экранные формы с примерами работы программы

