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

8

(количество листов)

Студент группы ИУ5-32:

Платонов Артемий

Подпись Дата 06.12.2017

Руководитель: Гапанюк Ю.Е

Подпись

Дата

Москва, 2017

***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:
   * 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

{

/// <summary>

/// Класс данных

/// </summary>

public class Staff

{

/// <summary>

/// Ключ

/// </summary>

public int id;

/// <summary>

/// Для группировки

/// </summary>

public string sur;

/// <summary>

/// Значение

/// </summary>

public int id\_d;

/// <summary>

/// Конструктор

/// </summary>

public Staff(int i, string s, int d)

{

this.id = i;

this.sur = s;

this.id\_d = d;

}

/// <summary>

/// Приведение к строке

/// </summary>

public override string ToString()

{

return "сотрудник (id=" + this.id.ToString() + "; sur=" + this.sur + "; id\_d=" + this.id\_d + ")";

}

}

/// <summary>

/// Класс для сравнения данных

/// </summary>

public class StaffEqualityComparer : IEqualityComparer<Staff>

{

public bool Equals(Staff x, Staff y)

{

bool Result = false;

if (x.id == y.id && x.sur == y.sur && x.id\_d == y.id\_d) Result = true;

return Result;

}

public int GetHashCode(Staff obj)

{

return obj.id;

}

}

/// <summary>

/// Связь между списками

/// </summary>

public class DataLink

{

public int id;

public string name;

public DataLink(int i, string n)

{

this.id = i;

this.name = n;

}

public override string ToString()

{

return "отдел (id=" + this.id.ToString() + "; name=" + this.name + ")";

}

}

public class LinkStaff

{

public int sid;

public int did;

public LinkStaff(int s, int d)

{

sid = s;

did = d;

}

}

//Пример данных

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

{

new Staff(1, "Афанасьев", 11),

new Staff(2, "Кузьменко", 12),

new Staff(3, "Абрикосов", 13),

new Staff(4, "Баранов", 15),

new Staff(5, "Нестеров", 15),

new Staff(6, "Анархов", 11),

new Staff(7, "Касаткина", 12),

new Staff(8, "Архипов", 12),

new Staff(9, "Иванов", 12),

new Staff(10, "Сидоров", 13)

};

static List<DataLink> lnk = new List<DataLink>()

{

new DataLink(11,"1"),

new DataLink(12,"4"),

new DataLink(13,"1"),

new DataLink(14,"4"),

new DataLink(15,"2")

};

static List<LinkStaff> linkstaff = new List<LinkStaff>

{

new LinkStaff(1,11),

new LinkStaff(1,12),

new LinkStaff(1,13),

new LinkStaff(2,11),

new LinkStaff(3,15),

new LinkStaff(3,13),

new LinkStaff(4,13),

new LinkStaff(5,12),

new LinkStaff(2,15),

new LinkStaff(3,12),

new LinkStaff(4,13)

};

static void Main(string[] args)

{

Console.WriteLine("Сортировка");

var q1 = from x in d1

orderby x.id\_d ascending

select x;

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

var q2 = from x in lnk

orderby x.id ascending, x.name ascending

select x;

foreach (var x in q2) Console.WriteLine(x);

Console.WriteLine("\nФамилия на А");

var q3 = from x in d1

where x.sur[0]=='А'

select x;

foreach (var x in q3) Console.WriteLine(x);

Console.WriteLine("\nкол-во сотрудников в отделах");

var q4 = from x in lnk

select x;

int count = 0;

foreach (var x in q4)

{

count = d1.Count(y => y.id\_d == x.id);

Console.WriteLine(x + " кол-во сотруников: " + count);

}

Console.WriteLine("\nвсе на А в отделе");

var q5 = from x in lnk

where (d1.Count(y => y.sur[0] == 'А' && y.id\_d == x.id) == d1.Count(y => y.id\_d == x.id)) && d1.Count(y => y.id\_d == x.id)!=0

select x;

foreach (var x in q5) Console.WriteLine(x);

Console.WriteLine("\nхотя бы одна фамилия на А в отделе");

var q6 =( from x in d1

from y in lnk

where x.id\_d == y.id && x.sur[0]=='А'

select y).Distinct();

foreach (var x in q6) Console.WriteLine(x);

Console.WriteLine("\nсотрудники в каждом отделе");

foreach (var x in lnk)

{

var q7 = from y in d1

from z in linkstaff

where z.sid == y.id && z.did == x.id

select y;

Console.WriteLine(x);

foreach (var z in q7) Console.WriteLine(z);

}

Console.WriteLine("\nкол-во сотрудников в отделах");

foreach (var x in lnk)

{

var q7 = from y in linkstaff

where (y.did == x.id)

select y;

Console.WriteLine(x + ": " + q7.Count());

}

}

}

}



