Отчёт по лабораторной работе №7 по курсу БКИТ

«Работа с LINQ»

Подготовил

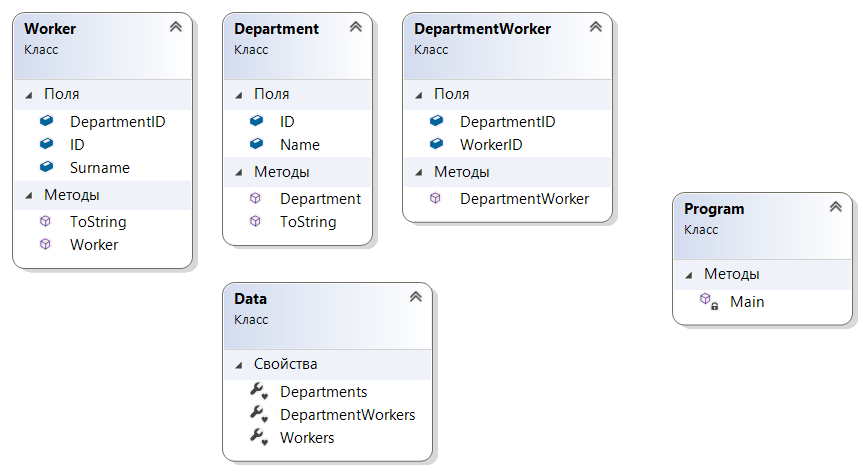
Ионов С.А.

ИУ5-31Б

1. 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.
7. Диаграмма классов



1. Текст программы

1) Program.cs

using System;

using System.Linq;

namespace Lab7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("\t\t1)Связь 1:М от Отдел к Сотрудник");

Console.WriteLine("\t1.1. Список всех сотрудников и отделов, отсортированный по отделам");

foreach (var d in Data.Departments)

{

//Перебор (укороченный синтаксис цикла) по каждому сотруднику

var q1 = from x in Data.Workers

where (d.ID == x.DepartmentID)

select x;

//Проверка на соответствие сотрудника его отделу (вывод типа: отдел - сотрудники)

Console.WriteLine(d);

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

}

Console.WriteLine("\n\t1.2. Список всех сотрудников, у которых фамилия начинается с буквы А");

string lit = "А";

var q2 = from x in Data.Workers

where (x.Surname.Substring(0, 1) == lit)

select x;

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

if (q2.Count() == 0)

{

Console.WriteLine("Такие сотрудники отсутствуют");

}

Console.WriteLine("\n\t1.3. Cписок всех отделов и количество сотрудников в каждом отделе");

foreach (var x in Data.Departments)

{

int num = Data.Workers.Count(y => y.DepartmentID == x.ID);

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

}

Console.WriteLine("\n\t1.4. Cписок отделов, в которых у всех сотрудников фамилия начинается с буквы А");

var q3 = from x in Data.Departments

where (Data.Workers.Count(y => y.Surname.Substring(0, 1) == "А" && y.DepartmentID == x.ID) ==

Data.Workers.Count(y => y.DepartmentID == x.ID))

select x;

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

if (q3.Count() == 0)

{

Console.WriteLine("Такие отделы отсутствуют");

}

Console.WriteLine("\n\t1.5. Cписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы А");

var q4 = from x in Data.Departments

where (Data.Workers.Count(y => y.Surname.Substring(0, 1) == "А" && y.DepartmentID == x.ID) > 0)

select x;

foreach (var x in q4) Console.WriteLine(x);

if (q4.Count() == 0)

{

Console.WriteLine("Такие отделы отсутствуют");

}

Console.WriteLine("\n\t\t2)Связь М:М от Отдел к Сотрудник");

Console.WriteLine("\t2.1. Cписок всех отделов и список сотрудников в каждом отделе");

foreach (var x in Data.Departments)

{

var q5 = from y in Data.DepartmentWorkers

where (y.DepartmentID == x.ID)

select y;

var q6 = from y in Data.Workers

from z in q5

where (z.WorkerID == y.ID)

select y;

Console.WriteLine(x);

foreach (var y in q6) Console.WriteLine(" ID: " + y.ID + "; Фамилия: " + y.Surname);

}

Console.WriteLine("\n\t2.2. Список всех отделов и количество сотрудников в каждом отделе");

foreach (var x in Data.Departments)

{

var q5 = from y in Data.DepartmentWorkers

where (y.DepartmentID == x.ID)

select y;

Console.WriteLine(x + "; Количество: " + q5.Count());

}

Console.ReadLine();

}

}

}

2)BD.cs

namespace Lab7

{

class Worker//Сотрудник

{

public int ID;//Первичный ключ

public string Surname;//Фамилия

public int DepartmentID;//ID отдела

public Worker(int id, string sn, int dep)

{

ID = id;

Surname = sn;

DepartmentID = dep;

}

public override string ToString()

{

return "ID: " + ID + "; Фамилия: " + Surname + "; ID\_Отдела: " + DepartmentID;

}

}

class Department //Отдел

{

public int ID;//ID отдела

public string Name;//Название

public Department(int id, string nm)

{

ID = id;

Name = nm;

}

public override string ToString()

{

return "ID: " + ID + "; Название отдела: " + Name;

}

}

class DepartmentWorker//Класс сотрудники отдела

{

public int WorkerID;//ID работника

public int DepartmentID;//ID отдела

public DepartmentWorker(int idW, int idD)

{

WorkerID = idW;

DepartmentID = idD;

}

}

}

3)Data.cs

using System.Collections.Generic;

namespace Lab7

{

class Data

{

internal static List<Worker> Workers { get; } = new List<Worker>()//Список сотрудников

{

new Worker(1, "Петров", 1),

new Worker(2, "Иванов", 2),

new Worker(3, "Алтуфьев", 1),

new Worker(4, "Игнатьева", 3),

new Worker(5, "Прокофьев", 2),

new Worker(6, "Александрова", 1),

new Worker(7, "Алексеев", 3),

};

internal static List<DepartmentWorker> DepartmentWorkers { get; } = new List<DepartmentWorker>//Список сотрудников отдела

{

new DepartmentWorker(1,1),

new DepartmentWorker(1,2),

new DepartmentWorker(1,3),

new DepartmentWorker(2,1),

new DepartmentWorker(3,1),

new DepartmentWorker(3,3),

new DepartmentWorker(4,3),

new DepartmentWorker(5,2),

new DepartmentWorker(6,1),

new DepartmentWorker(7,2),

new DepartmentWorker(7,3)

};

internal static List<Department> Departments { get; } = new List<Department>()//Список отделов

{

new Department(1, "Отдел продаж"),

new Department(2, "Отдел закупок"),

new Department(3, "Отдел кадров"),

};

}

}

1. Экранные формы с примерами выполнения программы

