**Лабораторная работа № 7 по курсу**

**“Базовые компоненты интернет-технологий”**

Выполнил: Арнаут А.И.

Группа: РТ5-31

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.

**Исходный код:**

using System;

using System.Collections.Generic;

using System.Linq;

namespace LAB7

{

class Program

{

class Worker

{

public int WorkerID;

public string Surname;

public int IDDep;

public Worker(int a, string b, int c)

{

this.WorkerID = a;

this.Surname = b;

this.IDDep = c;

}

public override string ToString()

{

return this.WorkerID.ToString() + ": Фамилия:" + this.Surname + ": ID отделения сотрудника =" + this.IDDep;

}

}

class Department

{

public int IDDep;

public string DepartmentName;

public Department(int id, string name)

{

this.IDDep = id;

this.DepartmentName = name;

}

public override string ToString()

{

return "(ID отделения =" + this.IDDep.ToString() + "; Название отделения:" + this.DepartmentName + ")";

}

}

class DepartmentWorker

{

public int WorkerID;

public int IDDep;

public DepartmentWorker(int wid, int iddep)

{

this.WorkerID = wid;

this.IDDep = iddep;

}

}

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

{

new Worker(1,"Аристотель",1),

new Worker(2,"Алигьери",1),

new Worker(3,"Платон",3),

new Worker(4,"Толстой",3),

new Worker(5,"Кинг",2),

new Worker(6,"Хэмингуэй",2),

new Worker(7,"Андерсон",1),

new Worker(8,"Альдани",2),

new Worker(9,"Моэм",3),

new Worker(10,"Сэлинджер",3),

};

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

{

new Department(1,"Отдел продаж"),

new Department(2,"Отдел кадров"),

new Department(3,"Руководители"),

};

static List<DepartmentWorker> DWList = new List<DepartmentWorker>()

{

new DepartmentWorker(1,1),

new DepartmentWorker(2,1),

new DepartmentWorker(3,3),

new DepartmentWorker(4,3),

new DepartmentWorker(5,2),

new DepartmentWorker(6,2),

new DepartmentWorker(7,1),

new DepartmentWorker(8,2),

new DepartmentWorker(9,3),

new DepartmentWorker(10,3),

};

static void Main(string[] args)

{

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

var AllWorkers = from x in DepartmentList

join s in WorkerList on x.IDDep equals s.IDDep into temp

select new { Department = x.DepartmentName, Worker = temp };

foreach (var x in AllWorkers)

{

Console.WriteLine(" Тип отделения: " + x.Department);

foreach (var s in x.Worker)

Console.WriteLine(" " + s);

}

Console.WriteLine("Сотрудники на А");

var AWorkers = from x in WorkerList where x.Surname.StartsWith("А") select new { ID = x.WorkerID, Фамилия = x.Surname, };

foreach (var x in AWorkers)

{

Console.WriteLine("Фамилия: " + x.Фамилия);

}

Console.WriteLine("Список всех отделов и колличество сотрудников в них.");

var NumWorkers = from x in DepartmentList

join s in WorkerList on x.IDDep equals s.IDDep into temp

select new { Department = x.DepartmentName, Quantity = temp.Count() };

foreach (var x in NumWorkers)

{

Console.WriteLine(x.Department + ":" + x.Quantity);

}

Console.WriteLine("Список всех отделов,в которых хотя бы у одного сотруника фамилия начинается на А.");

var AnyWorkersA = from x in DepartmentList

join s in WorkerList on x.IDDep equals s.IDDep into temp

where temp.Any(x => x.Surname.StartsWith("А"))

select new { Department = x.DepartmentName };

foreach (var x in AnyWorkersA)

{

Console.WriteLine("Название отдела:" + x.Department);

}

Console.WriteLine("Список всех отделов,в которых у кажого сотруника фамилия начинается на А.");

var AllWorkersA = from x in DepartmentList

join s in WorkerList on x.IDDep equals s.IDDep into temp

where temp.All(x => x.Surname.StartsWith("А"))

select new { Department = x.DepartmentName };

foreach (var x in AllWorkersA)

{

Console.WriteLine("Название отдела:" + x.Department);

}

Console.WriteLine("Список всех отделов и список сотрудников в каждом отделе.");

var DWlink = from x in WorkerList

join y in DWList on x.WorkerID equals y.WorkerID into temp

from z1 in temp

join h in DepartmentList on z1.IDDep equals h.IDDep into temp2

from z2 in temp2

group z2 by z2.DepartmentName into g

select new

{ Department = g.Key, Count = g.Count() };

foreach (var d in DWlink)

{

Console.WriteLine(d.Department + ":" + d.Count);

}

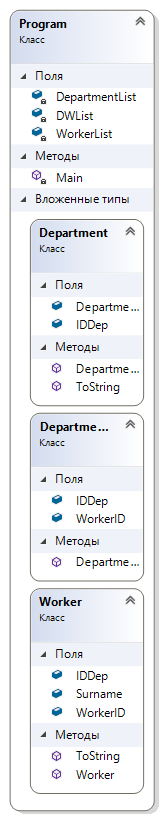
Console.ReadKey();

}

}

}

**Диаграмма классов:**



**Пример выполнения программы:**

