|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет имени Н.Э. Баумана**  **(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ                              Информатика и системы управления

КАФЕДРА                  Системы обработки информации и управления

**Лабораторная работа №7**

**По курсу «Базовые компоненты интернет-технологий»**

Подготовила:

Студентка группы ИУ5-35Б.

Очеретная С.В.

18.11.2020

Проверил:

Преподаватель кафедры ИУ5  
Гапанюк Ю.Е.

Москва, 2020 г.

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.

Текст программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace LABA7\_BKIT

{

class Program

{

// Класс Сотрудник (2)

public class Employee

{

// Первичный ключ - ID запись о сотруднике

public int id;

// Фамилия сотрудника

public string surname;

// ID записи об отделе.

public int department;

// Конструктор

public Employee(int i, string srnm, int d)

{

this.id = i;

this.surname = srnm;

this.department = d;

}

// Приведение к строке

public override string ToString()

{

return "|ID сотрудника: " + this.id.ToString() + " Фамилия сотрудника: " + this.surname + " ID отдела: " + this.department.ToString() + "|";

}

public bool StartsWith(char letter) // (для 4 задания - возвращает первую букву)

{

return surname[0] == letter;

}

}

// класс отдел (3)

public class Department

{

// Первичный ключ - ID запись об отделе

public int id;

// Наименование отдела

public string name;

public Department(int i, string name)

{

this.id = i;

this.name = name;

}

// Приведение к строке

public override string ToString()

{

return "|ID отдела: " + this.id.ToString() + " Наименовение отдела:" + this.name + "|";

}

public bool StartsWith(char letter) // (4)

{

return name[0] == letter;

}

}

// Связь между списками, 1 ко многим

public class DepatmentEmpls // Сотрудники отдела (5)

{

public int eID;

public int dID;

public DepatmentEmpls(int dID, int eID) // ID отдела и сотрудника

{

this.dID = dID;

this.eID = eID;

}

}

// сотрудники

static List<Employee> employeesTable = new List<Employee>()

{

new Employee(1, "Аверьянов", 1),

new Employee(2, "Петров", 2),

new Employee(3, "Очеретная", 3),

new Employee(4, "Герасимов", 3),

new Employee(5, "Турчин", 3),

new Employee(6, "Финк", 2),

new Employee(7, "Андреев", 1)

};

// отделы

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

{

new Department(1, "1 Отдел"),

new Department(2, "2 Отдел"),

new Department(3, "3 Отдел"),

new Department(4, "4 Отдел")

};

// Таблица связей много ко многим

static List<DepatmentEmpls> departmentsEmplsTable = new List<DepatmentEmpls>()

{

new DepatmentEmpls(1,1), // (ID отдела из списка List<Employee>, ID сотрудника из списка List<Department>)

new DepatmentEmpls(1,4),

new DepatmentEmpls(2,5),

new DepatmentEmpls(2,2),

new DepatmentEmpls(2,3),

new DepatmentEmpls(3,6),

new DepatmentEmpls(3,7)

};

static void Main(string[] args)

{

Console.WriteLine("Cписок всех сотрудников и отделов, отсортированный по отделам (Задание 4.1)");

var q1 = from e in employeesTable

join d in departmentsTable on e.department equals d.id // соединяем с условием соединения - одинаковая ID запись отдела

orderby d.name // cортировка по наименованию отдела типа "№ОТДЕЛА отдел", без descending, значит по возрастанию

select new { empl = e.surname, dep = d.name }; // генерация sql запроса с данными об фамилии сотрудника и названии отдела

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

Console.WriteLine("\nCписок всех сотрудников, у которых фамилия начинается с буквы «А» (Задание 4.2)");

var q2 = from e in employeesTable

where e.StartsWith('А') // условие отбора - фамилия с буквы А

select e; // возвращается список сотрудников

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

Console.WriteLine("\nCписок всех отделов и количество сотрудников в каждом отделе (Задание 4.3)");

var q3 = from e in employeesTable

group e by e.department into g // Группировка, ключ группировки - ID отдела

select new { id = g.Key, amount = g.Count() };

// т.е. проходим по сотрудникам и раскидываем их по отделам в соответствии с ID отделом, которому они принадлежат

// в итоге получаем запрос со списком групп, где id - ключ группировки, amount - колличество сотрудников в группе

// причем пустые отделы не отображаются, т.к. там нет сотрудников

foreach (var x in q3) // проходимся по сформированным группам

{

var q23 = from d in departmentsTable // ищем отдел с нужным ID группы

where d.id == x.id // условие выборки - где ID отдела = ID отдела сотрудника, используется связь один ко многим

select d.name; // возвращаем название отдела

foreach (var d in q23) Console.WriteLine(d + ": " + x.amount + " сотрудника");

}

Console.WriteLine("\nCписок отделов, в которых у всех сотрудников фамилия начинается с буквы «А»(Задание 4.4)");

var q4 = from e in employeesTable

group e by e.department into g

where g.All(e => e.StartsWith('А')) // здесь уже условие для всех групп

select new { id = g.Key, vals = g }; // возвращает запрос с нужными группами (отделами)

foreach (var x in q4) // тоже самое, как для запроса 23

{

var q24 = from d in departmentsTable

where d.id == x.id

select d.name;

foreach (var d in q24) Console.WriteLine(d);

}

Console.WriteLine("\nCписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А»(Задание 4.5)");

var q5 = from e in employeesTable

group e by e.department into g

where g.Any(e => e.StartsWith('А')) // здесь уже хотя бы у одного сотрудника

select new { dep = g.Key, vals = g };

foreach (var x in q5) // все тоже самое

{

var q25 = from d in departmentsTable

where d.id == x.dep

select d.name;

foreach (var d in q25) Console.WriteLine(d);

}

Console.WriteLine("\nСписок всех отделов и список сотрудников в каждом отделе (Задание 6.1)");

var q36 = from d in departmentsTable

join de in departmentsEmplsTable on d.id equals de.dID into Dtemp // для каждого ID отдела с общей таблицы формируем список с отделами

from dd in Dtemp

join e in employeesTable on dd.eID equals e.id into Etemp // из каждого сформированного списка выбираем сотрудников с ID нужного отдела

group new { depName = d.name, empls = Etemp } by d.name into g // формируем группы для каждого отдела с именем и списком сотрудников

select new { depName = g.Key, deps = g }; // формируем общий запрос с названием отдела и группой отделов

foreach (var x in q36)

{

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

foreach (var d in x.deps)

foreach (var ee in d.empls)

Console.WriteLine(" " + ee.surname);

}

Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе (Задание 6.2)"); // то же самое, только с колличеством

var q66 = from d in departmentsTable

join de in departmentsEmplsTable on d.id equals de.dID into Dtemp

from dd in Dtemp

join e in employeesTable on dd.eID equals e.id into Etemp

group new { depName = d.name, empls = Etemp } by d.name into g

select new { depName = g.Key, depsCount = g.Count() };

foreach (var d in q66)

{

Console.WriteLine(d.depName + " колличество сотрудников:" + d.depsCount);

}

Console.ReadLine();

}

}

}

Результаты

