**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

«Работа с LINQ»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Перова Анна Евгеньевна |  | Гапанюк Юрий  Евгеньевич |
| Подпись и дата: |  | Подпись и дата:  10.12.20 |

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.Linq;

using System.Collections.Generic;

namespace dzBKIT

{

class Program

{

public class Employee

{

public int ID;

public string LastName;

public int Department\_ID;

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

{

this.ID = a;

this.LastName = b;

this.Department\_ID = c;

}

public override string ToString()

{

return "(ID=" + this.ID.ToString() + "; LastName=" + this.LastName + "; Department ID" + this.Department\_ID.ToString() + ")";

}

}

public class Department

{

public int ID;

public string Name;

public Department(int a, string b)

{

this.ID = a;

this.Name = b;

}

public override string ToString()

{

return "(ID=" + this.ID.ToString() + "; Name=" + this.Name + ")";

}

}

public class EmployeeDepartment

{

public int ID\_Employee;

public int ID\_Department;

public EmployeeDepartment(int a, int b)

{

this.ID\_Employee = a;

this.ID\_Department = b;

}

public override string ToString()

{

return "(ID\_Employee=" + this.ID\_Department.ToString() + "; id\_branch=" + this.ID\_Department.ToString() + ")";

}

}

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

{

new Employee(1, "Аваров", 1),

new Employee(2, "Абакумов", 1),

new Employee(3, "Агашин", 2),

new Employee(4, "Айвазян", 2),

new Employee(5, "Иванова", 3),

new Employee(6, "Петрова", 3),

new Employee(7, "Козлов", 4),

new Employee(8, "Перова", 4),

new Employee(9, "Андреева", 5),

new Employee(10, "Белюкин", 5),

new Employee(11, "Осипова", 5)

};

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

{

new Department(1, "Финансовый отдел"),

new Department(2, "Отдел кадров"),

new Department(3, "Коммерческий отдел"),

new Department(4, "Отдел логистики"),

new Department(5, "Отдел рекламы")

};

static List<EmployeeDepartment> emloyees\_departments = new List<EmployeeDepartment>()

{

new EmployeeDepartment(1, 1),

new EmployeeDepartment(1, 2),

new EmployeeDepartment(2, 1),

new EmployeeDepartment(3, 3),

new EmployeeDepartment(4, 3),

new EmployeeDepartment(5, 4),

new EmployeeDepartment(5, 3),

};

static void Main(string[] args)

{

Console.WriteLine("Связь 1:М");

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

var z1 = from x in employees

from y in departments

where x.Department\_ID == y.ID

orderby y.Name

select new { department = y.Name, employee = x.LastName };

foreach (var x in z1) Console.WriteLine(x);

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

var z2 = from x in employees

where x.LastName[0] == 'А'

select x.LastName;

foreach (var x in z2) Console.WriteLine(x);

Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе:");

var z3 = (from x in departments

join y in employees on x.ID equals y.Department\_ID into temp

from t in temp

select new { department= x.Name, employees\_number = temp.Count() }).Distinct();

foreach (var x in z3) Console.WriteLine(x);

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

var z4 = (from x in departments

join y in employees on x.ID equals y.Department\_ID into temp

from t in temp

where temp.Count(t => t.LastName[0] == 'А') == temp.Count()

select x.Name).Distinct();

foreach (var x in z4) Console.WriteLine(x);

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

var z5 = (from x in departments

join y in employees on x.ID equals y.Department\_ID into temp

from t in temp

where t.LastName[0] == 'А'

select x.Name).Distinct();

foreach (var x in z5) Console.WriteLine(x);

Console.WriteLine("\nСвязь М:М");

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

var z6 = from x in departments

from y in employees

join z in emloyees\_departments on x.ID equals z.ID\_Department into temp1

from t1 in temp1

where t1.ID\_Employee == y.ID

select new { department = x.Name, employee = y.LastName };

string d = "";

foreach (var x in z6)

{

if (x.department != d)

{

d = x.department;

Console.WriteLine(d + ":");

Console.WriteLine(x.employee);

}

else

{

Console.WriteLine(x.employee);

}

}

Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе:");

var z7 = (from x in departments

join y in emloyees\_departments on x.ID equals y.ID\_Department into temp

from t in temp

select new { department = x.Name, employee\_number = temp.Count() }).Distinct();

foreach (var x in z7) Console.WriteLine(x);

Console.ReadLine();

}

}

}

Скриншот с примером работы программы

