**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №7

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35б |  | преподаватель каф. ИУ5 |
| Ширшов Артём |  |  |
| Подпись и дата: |  | Подпись и дата: |

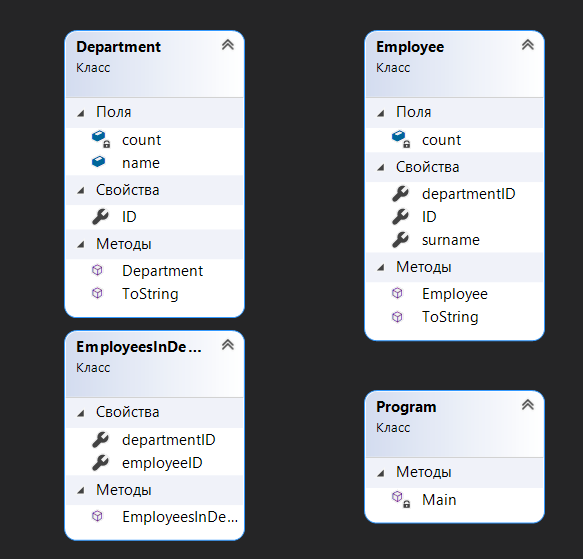
Москва, 2020 г.

# 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.

# Диаграмма классов



# Текст программы

Class1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7

{

class Employee

{

private static int count = 0;

public int ID { get; private set; }

public string surname { get; private set;}

public int departmentID { get; private set; }

public Employee(string surname, Department department)

{

this.surname = surname;

departmentID = department.ID;

count++;

ID = count;

}

public override string ToString()

{

return surname;

}

}

class Department

{

private static int count = 0;

public string name;

public int ID { get; private set; }

public Department(string name)

{

this.name = name;

count++;

ID = count;

}

public override string ToString()

{

return name;

}

}

class EmployeesInDepartments

{

public int departmentID { get; private set; }

public int employeeID { get; private set; }

public EmployeesInDepartments(int employeeID, int departmentID)

{

this.employeeID = employeeID;

this.departmentID = departmentID;

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab7

{

class Program

{

static void Main(string[] args)

{

var otdelSnab = new Department("Отдел снабжения");

var otdelKadr = new Department("Отдел кадров");

var otdelKach = new Department("Отдел качества");

var otdelDisp = new Department("Диспетчерский отдел");

var otdelUrid = new Department("Юридический отдел");

var otdelIT = new Department("Отдел Разработки");

List<Department> departments = new List<Department>() { otdelSnab, otdelKadr, otdelKach, otdelDisp, otdelUrid, otdelIT };

var p1 = new Employee("Ширшов", otdelIT);

var p2 = new Employee("Артемьев", otdelSnab);

var p3 = new Employee("Иванов", otdelKach);

var p4 = new Employee("Смыслов", otdelDisp);

var p5 = new Employee("Апельсинкина", otdelUrid);

var p6 = new Employee("Хижняков", otdelIT);

var p7 = new Employee("Савельев", otdelSnab);

var p8 = new Employee("Астафьев", otdelIT);

var p9 = new Employee("Полушкин", otdelDisp);

var p10 = new Employee("Рябкин", otdelKadr);

var p11 = new Employee("Абрикосова", otdelUrid);

List<Employee> employees = new List<Employee>() { p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11 };

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

var orderedEmployees = employees.OrderBy(i => i.departmentID);

foreach (var selectedEmployee in orderedEmployees)

{

Console.WriteLine(departments.Where(selectedDepartment => selectedDepartment.ID == selectedEmployee.departmentID).First() + " - " + selectedEmployee.surname);

}

Console.WriteLine("--------------------------------------------------------");

Console.WriteLine("Cписок всех сотрудников, у которых фамилия начинается с буквы А:");

var employeesWithFirstLetterA = from neededEmployee in employees

where neededEmployee.surname.StartsWith("А")

select neededEmployee;

foreach (var selectedEmployee in employeesWithFirstLetterA)

{

Console.WriteLine(selectedEmployee.surname);

}

Console.WriteLine("--------------------------------------------------------");

foreach (var departmentWithEmployees in departments)

{

var numberOfEmployeesInDepartment = employees.Where(neededEmployee => neededEmployee.departmentID == departmentWithEmployees.ID).Count();

Console.WriteLine(departmentWithEmployees.name + " Количество сотрудников:" + numberOfEmployeesInDepartment);

}

Console.WriteLine("--------------------------------------------------------");

foreach (var selectedDepartment in departments)

{

var departmentWithEmployees = from neededEmployees in employees

where selectedDepartment.ID == neededEmployees.departmentID

select neededEmployees;

if (departmentWithEmployees.All(selectedEmployee => selectedEmployee.surname.StartsWith("А")))

{

Console.WriteLine(selectedDepartment.name);

}

}

Console.WriteLine("--------------------------------------------------------");

foreach (var selectedDepartment in departments)

{

var departmentWithEmployees = from neededEmployees in employees

where selectedDepartment.ID == neededEmployees.departmentID

select neededEmployees;

if (departmentWithEmployees.Any(selectedEmployee => selectedEmployee.surname.StartsWith("А")))

{

Console.WriteLine(selectedDepartment.name);

}

}

Console.WriteLine("--------------------------------------------------------");

Console.WriteLine("MANY TO MANY PART");

Console.WriteLine("--------------------------------------------------------");

List<EmployeesInDepartments> manyToManyList = new List<EmployeesInDepartments>()

{

new EmployeesInDepartments(p1.ID, otdelDisp.ID),

new EmployeesInDepartments(p1.ID, otdelIT.ID),

new EmployeesInDepartments(p2.ID, otdelKach.ID),

new EmployeesInDepartments(p2.ID, otdelKadr.ID),

new EmployeesInDepartments(p3.ID, otdelSnab.ID),

new EmployeesInDepartments(p3.ID, otdelUrid.ID),

new EmployeesInDepartments(p4.ID, otdelDisp.ID),

new EmployeesInDepartments(p4.ID, otdelUrid.ID),

new EmployeesInDepartments(p5.ID, otdelIT.ID),

new EmployeesInDepartments(p5.ID, otdelSnab.ID),

new EmployeesInDepartments(p6.ID, otdelKach.ID),

new EmployeesInDepartments(p6.ID, otdelKadr.ID),

new EmployeesInDepartments(p7.ID, otdelUrid.ID),

new EmployeesInDepartments(p8.ID, otdelIT.ID),

new EmployeesInDepartments(p9.ID, otdelIT.ID),

new EmployeesInDepartments(p10.ID, otdelSnab.ID),

};

var employeesWithConnection = manyToManyList.Join(

employees,

connection => connection.employeeID,

employee => employee.ID,

(connection, employee) => new { Surname = employee.surname, DepID = connection.departmentID });

var resultCollection = departments.GroupJoin(

employeesWithConnection,

department => department.ID,

connectionCollection => connectionCollection.DepID,

(dep, conCol) => new {depName = dep.name, Surnames = conCol.Select(collection => collection.Surname) });

foreach (var departmentCollection in resultCollection)

{

Console.WriteLine("-");

Console.WriteLine(departmentCollection.depName);

foreach(var empSurname in departmentCollection.Surnames)

{

Console.WriteLine(empSurname);

}

}

Console.WriteLine("--------------------------------------------------------");

foreach (var departmentCollection in resultCollection)

{

Console.WriteLine("-");

var numberOfEmployeesInDepartment = (departmentCollection.Surnames.Count());

Console.WriteLine(departmentCollection.depName +" Количество сотрудников:"+ numberOfEmployeesInDepartment);

}

Console.ReadKey();

}

}

}

# Результаты

