Московский государственный технический университет и Н.Э. Баумана

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-5 «Системы обработки информации и управления»

ОТЧЕТ

по лабораторной работе №7

Выполнил:

Студент ИУ5-32Б

Нырков Илья Алексеевич

Проверил:

Доцент Гапанюк Ю. Е.

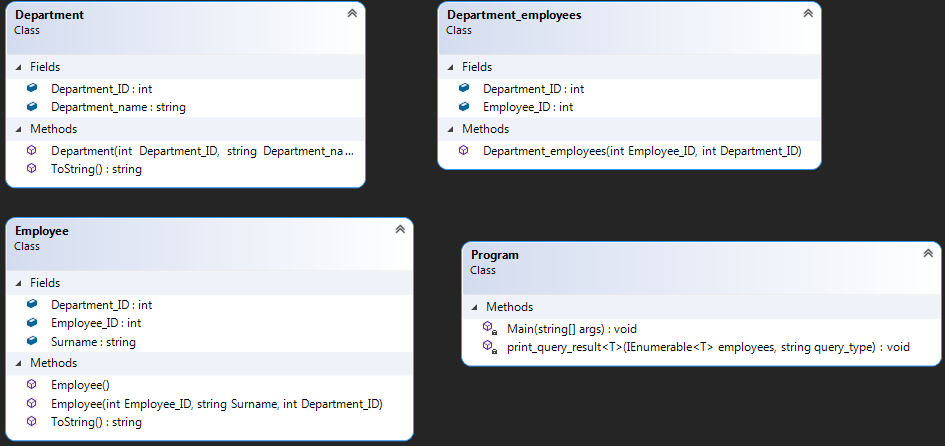
МОСКВА 2020

**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.

**Диаграмма классов:**

****

**Листинг кода программы:**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Threading;

namespace Linq

{

class Employee

{

public int Employee\_ID;

public string Surname;

public int Department\_ID;

public Employee() { }

public Employee(int Employee\_ID, string Surname, int Department\_ID)

{

this.Employee\_ID = Employee\_ID;

this.Surname = Surname;

this.Department\_ID = Department\_ID;

}

public override string ToString()

{

return ( "Department\_ID= " + Department\_ID.ToString() +

" ID= " + Employee\_ID.ToString() + " " + Surname);

}

}

class Department

{

public int Department\_ID;

public string Department\_name;

public Department(int Department\_ID, string Department\_name)

{

this.Department\_ID = Department\_ID;

this.Department\_name = Department\_name;

}

public override string ToString()

{

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

" name= " + Department\_name);

}

}

//5

class Department\_employees

{

public int Employee\_ID;

public int Department\_ID;

public Department\_employees(int Employee\_ID, int Department\_ID)

{

this.Employee\_ID = Employee\_ID;

this.Department\_ID = Department\_ID;

}

}

class Program

{

static void print\_query\_result<T>(IEnumerable<T> employees, string query\_type)

{

Console.WriteLine("+++++++++++++++++++++++++++++++++++++++++\n" + query\_type);

foreach (var x in employees)

{

Console.WriteLine(x.ToString());

}

}

static void Main(string[] args)

{

//data

List<Employee> employees = new List<Employee>

{

new Employee(1,"Antonov",5),

new Employee(2,"Lebedev",4),

new Employee(3,"Abramovich",5),

new Employee(4,"Viktorovich",4),

new Employee(5,"Robertovich",4),

new Employee(6,"Alibasov",5),

new Employee(7,"Alexeev",5),

new Employee(8,"Nikitina",4),

new Employee(9,"Lapenko",4),

new Employee(10,"Albertov",4),

new Employee(11,"Leonov", 3),

new Employee(12,"Radionov", 3),

new Employee(13,"Nixon", 3)

};

List<Department> departments = new List<Department>

{

new Department(5, "Human resourses"),

new Department(4, "IT development"),

new Department(3, "IT testing")

};

//4

var departments\_employees = from x in employees

orderby x.Department\_ID

select x;

print\_query\_result(departments\_employees, nameof(departments\_employees));

var begin\_with\_a = from x in employees

where x.Surname.StartsWith("A")

select x;

print\_query\_result(begin\_with\_a, nameof(begin\_with\_a));

var deps\_emp\_cnt = from x in departments

select new { dep\_name = x.Department\_name,

value = employees.Count(p => p.Department\_ID == x.Department\_ID) };

Console.WriteLine("+++++++++++++++++++++++++++++++++++++++++\ndeps\_emp\_cnt");

foreach(var item in deps\_emp\_cnt)

{

Console.WriteLine(item.dep\_name + " has " + item.value + " employees");

}

var deps\_with\_All\_A\_employees = from x in departments

where employees.Count(p => p.Department\_ID == x.Department\_ID) ==

employees.Count(p => p.Surname.StartsWith("A") && p.Department\_ID == x.Department\_ID)

select x;

print\_query\_result(deps\_with\_All\_A\_employees, nameof(deps\_with\_All\_A\_employees));

var deps\_with\_A\_employees = from x in departments

where employees.Count(p => p.Surname.StartsWith("A") &&

p.Department\_ID == x.Department\_ID) > 0

select x;

print\_query\_result(deps\_with\_A\_employees, nameof(deps\_with\_A\_employees));

//6

List<Department\_employees> dep\_employees = new List<Department\_employees>();

foreach(var x in employees)

{

Department\_employees tmp = new Department\_employees(x.Employee\_ID, x.Department\_ID);

dep\_employees.Add(tmp);

if (tmp.Department\_ID == 4)

{

Department\_employees tmp2 = new Department\_employees(x.Employee\_ID, 3);

dep\_employees.Add(tmp2);

}

}

var query\_empl\_in\_deps = from d in dep\_employees

join e in employees on d.Employee\_ID equals e.Employee\_ID

orderby d.Department\_ID

select new { Department\_ID = d.Department\_ID, Surname = e.Surname };

Console.WriteLine("+++++++++++++++++++++++++++++++++++++++++\nquery\_empl\_in\_deps");

foreach (var item in query\_empl\_in\_deps)

{

Console.WriteLine("Dep\_ID = " + item.Department\_ID + " " + item.Surname);

}

var query\_empl\_in\_deps\_cnt = from x in departments

select new { dep\_name = x.Department\_name,

empl\_count = dep\_employees.Count(p => p.Department\_ID == x.Department\_ID) };

Console.WriteLine("+++++++++++++++++++++++++++++++++++++++++\nquery\_empl\_in\_deps\_cnt");

foreach (var item in query\_empl\_in\_deps\_cnt)

{

Console.WriteLine(item.dep\_name + " has " + item.empl\_count + " employees");

}

}

}

}

**Скриншоты выполнения программы:**

