Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

Московский государственный технический университет имени Н.Э.Баумана

(МГТУ им. Н.Э.Баумана)

**Отчет по лабораторной работе №7**

**по курсу БКИТ**

Выполнил: Труфанов Вадим, ИУ 5-33

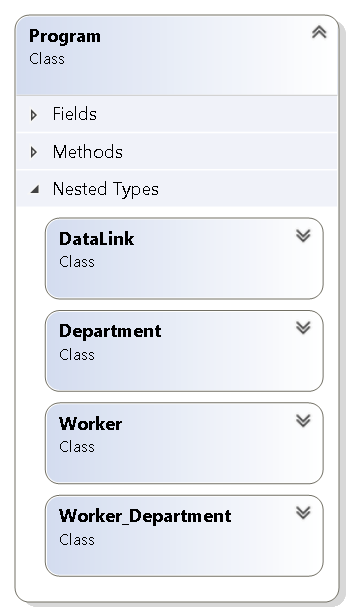
Преподаватель: Гапанюк Ю.Е.

Москва, 2018

1. **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.
7. **Диаграмма классов**



1. **Текст программы**

*Класс Program*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab7

{

class Program

{

public class Worker

{

public int id;

public string surname;

public int department\_id;

public Worker(int i, string s, int d)

{

this.id = i;

this.surname = s;

this.department\_id = d;

}

public override string ToString()

{

return "(id=" + this.id.ToString() + "; surname=" + this.surname + "; dep\_id=" + this.department\_id + ")";

}

}

public class Department

{

public int id;

public string name;

public Department(int i, string s)

{

this.id = i;

this.name = s;

}

public override string ToString()

{

return "(id=" + this.id.ToString() + " name=" + this.name + ")";

}

}

public class Worker\_Department

{

public int work\_id;

public int dep\_id;

public Worker\_Department(int w, int d)

{

this.work\_id = w;

this.dep\_id = d;

}

public override string ToString()

{

return "(work\_id=" + this.work\_id.ToString() + " dep\_id = " + this.dep\_id.ToString() + ")";

}

}

public class DataLink

{

public int d1;

public int d2;

public DataLink(int i1, int i2)

{

this.d1 = i1;

this.d2 = i2;

}

}

//Пример данных

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

{

new Worker(1, "Аносов", 1),

new Worker(2, "Андронов", 1),

new Worker(3, "Бессонов", 3),

new Worker(4, "Лобачев", 2),

new Worker(5, "Амелин", 2),

new Worker(6, "Попов", 3),

new Worker(7, "Иванов", 2),

new Worker(8, "Туполев", 3)

};

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

{

new Department(1, "Руководство"),

new Department(2, "Маркетинг"),

new Department(3, "Разработчики")

};

static List<Worker\_Department> work\_deps = new List<Worker\_Department>()

{

new Worker\_Department(1,1),

new Worker\_Department(1,3),

new Worker\_Department(2,1),

new Worker\_Department(3,2),

new Worker\_Department(3,1),

new Worker\_Department(4,2),

new Worker\_Department(5,3),

new Worker\_Department(5,1),

new Worker\_Department(6,2),

new Worker\_Department(6,3)

};

static void Main(string[] args)

{

Console.WriteLine("Сотрудники");

var q00 = from x in workers select x;

foreach (var x in q00) Console.WriteLine(x);

Console.WriteLine("Отделы");

var q01 = from x in deps select x;

foreach (var x in q01) Console.WriteLine(x);

Console.WriteLine("\n\nСписок всех сотрудников и отделов, отсортированный по отделам");

var q1 = from x in deps

join y in workers on x.id equals y.department\_id

orderby x.name

group y by x.id into g

select new { Key = g.Key, Values = g };

foreach (var x in q1)

{

var q2 = from z in deps

where z.id == x.Key

select z.name;

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

foreach (var y in x.Values)

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

}

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

var q3 = from x in workers

where x.surname[0] == 'А'

select x;

foreach (var x in q3) Console.WriteLine(x.surname);

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

var q4 = from x in deps

join y in workers on x.id equals y.department\_id

group y by x.id into g

select new { Key = g.Key, Values = g };

foreach (var x in q1)

{

string depName = deps.Where(z => z.id == x.Key).First().name;

Console.WriteLine(depName);

Console.WriteLine("Количество сотрудников: {0}", x.Values.Count());

}

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

var q5 = from x in deps

join y in workers on x.id equals y.department\_id

where y.surname[0] == 'А'

group y by x.id into g

select new { Key = g.Key, Values = g };

foreach (var x in q5)

{

int workCounts = workers.Where(z => z.department\_id == x.Key).Count();

if (x.Values.Count() == workCounts)

{

string depName = deps.Where(z => z.id == x.Key).First().name;

Console.WriteLine(depName);

foreach (var y in x.Values)

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

}

}

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

var q6 = from x in deps

join y in workers on x.id equals y.department\_id

where y.surname[0] == 'А'

group y by x.id into g

select new { Key = g.Key, Values = g };

foreach (var x in q6)

{

{

string depName = deps.Where(z => z.id == x.Key).First().name;

Console.WriteLine(depName);

foreach (var y in x.Values)

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

}

}

Console.WriteLine("\n\nМногие-к-многим");

var lnk0 = from x in workers

join l in work\_deps on x.id equals l.work\_id into temp

from t1 in temp

join y in deps on t1.dep\_id equals y.id into temp2

from t2 in temp2

select new { x.surname, t2.name };

foreach (var x in lnk0)

{

Console.WriteLine(x);

}

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

var lnk1 = from x in workers

join l in work\_deps on x.id equals l.work\_id into temp

from t1 in temp

join y in deps on t1.dep\_id equals y.id into temp2

from t2 in temp2

group x by t2.id into g

select new { Key = g.Key, Values = g };

foreach (var x in lnk1)

{

string depName = deps.Where(z => z.id == x.Key).First().name;

Console.WriteLine(depName);

foreach (var y in x.Values)

Console.WriteLine("\t" + y.surname);

}

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

var lnk2 = from x in workers

join l in work\_deps on x.id equals l.work\_id into temp

from t1 in temp

join y in deps on t1.dep\_id equals y.id into temp2

from t2 in temp2

group x by t2.id into g

select new { Key = g.Key, Values = g };

foreach (var x in lnk2)

{

string depName = deps.Where(z => z.id == x.Key).First().name;

Console.WriteLine(depName);

Console.WriteLine("Количество сотрудников: {0}", x.Values.Count());

}

Console.ReadLine();

}

}

}

1. **Экранные формы с примерами выполнения программы**

