Отчет по лабораторной работе 7 по курсу

«Базовые компоненты интернет технологий (БКИТ)»

Выполнил

Студент группы ИУ5-33

Зинченко Ярослав Игоревич

Москва 2017

**Требования:**

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 на языке C#:**

Файл Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab7

{

class Program

{

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

{

new Worker(1, "Kent", 23),

new Worker(2, "Oswald", 27),

new Worker(3, "Allen", 24),

new Worker(4, "Song", 25),

new Worker(5, "Lang", 23),

new Worker(6, "Manson", 26),

new Worker(7, "Kingston", 27),

new Worker(8, "Tennant", 25),

new Worker(9, "Jones", 24),

new Worker(10, "Tyler", 23),

new Worker(11, "Noble", 27),

new Worker(12, "Pond", 26),

new Worker(13, "Williams", 25),

new Worker(14, "Green", 24),

new Worker(15, "Carter", 23),

new Worker(16, "Jackson", 27),

new Worker(17, "Cameron", 26),

new Worker(18, "House", 27),

new Worker(19, "Alto", 24),

new Worker(20, "O'Neill", 23)

};

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

{

new Department(23, "Literature"),

new Department(24, "Programming"),

new Department(25, "Music"),

new Department(26, "Painting"),

new Department(27, "Science"),

};

static List<workers\_department> w\_d = new List<workers\_department>()

{

new workers\_department(1 , 23),

new workers\_department(2 , 27),

new workers\_department(3 , 24),

new workers\_department(4 , 25),

new workers\_department(5 , 23),

new workers\_department(6 , 26),

new workers\_department(7 , 27),

new workers\_department(8 , 25),

new workers\_department(9 , 24),

new workers\_department(10 , 23),

new workers\_department(11 , 27),

new workers\_department(12 , 26),

new workers\_department(13 , 25),

new workers\_department(14 , 24),

new workers\_department(15 , 23),

new workers\_department(16 , 27),

new workers\_department(17 , 26),

new workers\_department(18 , 27),

new workers\_department(19 , 24),

new workers\_department(20 , 23)

};

static void Main(string[] args)

{

Console.WriteLine("\n Here you can see results for all querries with link 1:M\n");

Console.WriteLine("\nList of all workers and departments, sorted by department: ");

var que1 = from x in workers

orderby x.ID\_Dep descending, x.ID\_W descending

select x;

foreach (var x in que1) Console.WriteLine(x);

Console.WriteLine("\nList of all workers with first 'A' in the surname: ");

var que2 = from x in workers

where x.Surname[0] is 'A'

orderby x.Surname ascending, x.ID\_W descending

select x;

foreach (var x in que2)

Console.WriteLine(x);

Console.WriteLine("\nInformation of all departments and about number of workers in each one: ");

var que3 = from x in deps

join y in workers on x.ID\_Dep equals y.ID\_Dep into temp

from t in temp

select new { Department = x.Name\_Dep, ID = x.ID\_Dep, count = temp.Count() };

que3 = que3.Distinct();

foreach (var x in que3)

Console.WriteLine(x);

Console.WriteLine("\nA list of departments in which all workers surnames start with an 'A': ");

var que4 = from x in workers

join y in que2 on x.ID\_Dep equals y.ID\_Dep into temp

from t in temp

select new { ID\_of\_Department = x.ID\_Dep, count = temp.Count() };

que4 = que4.Distinct();

var que4\_2 = from x in que3

from y in que4

where (x.count == y.count) && (x.ID == y.ID\_of\_Department)

select new { ID\_of\_Department = x.ID };

que4\_2 = que4\_2.Distinct();

foreach (var x in que4\_2)

Console.WriteLine(x);

Console.WriteLine("\n\nList of departments in which at least one worker " +

"has a surname beginning with 'A': ");

var que5 = from x in workers

where x.Surname[0] is 'A'

select new { v1 = x.ID\_Dep };

que5 = que5.Distinct();

var que5\_2 = from x in deps

from y in que5

where x.ID\_Dep == y.v1

select new { v1 = x.Name\_Dep };

que5\_2 = que5\_2.Distinct();

foreach (var x in que5\_2)

Console.WriteLine(x);

Console.WriteLine("\n Here you can see results for all querries with link M:M\n");

Console.WriteLine("All the departments and workers in each department: ");

var que6\_1 = from y in deps

join l in w\_d on y.ID\_Dep equals l.ID\_Dep into temp2

from t2 in temp2

select new { id = y.ID\_Dep, name = y.Name\_Dep };

que6\_1 = que6\_1.Distinct();

foreach (var y in que6\_1)

{

Console.WriteLine("\n\n" + y+ "\n");

var que6\_2 = from x in w\_d

where (y.id == x.ID\_Dep)

select new { id\_w = x.ID\_W };

que6\_2 = que6\_2.Distinct();

var que6\_3 = from t in workers

from t2 in que6\_2

where t2.id\_w == t.ID\_W

select new { id = t.ID\_W, surname = t.Surname};

que6\_3 = que6\_3.Distinct();

foreach (var t in que6\_3)

Console.WriteLine(t);

}

Console.WriteLine("\n\nList of all departments and number of worker in each department: ");

var que7\_1 = from x in deps

join l in w\_d on x.ID\_Dep equals l.ID\_Dep into temp2

from t2 in temp2

select new { id = x.ID\_Dep, name = x.Name\_Dep };

que7\_1 = que7\_1.Distinct();

foreach (var x in que7\_1)

{

Console.WriteLine("\n" + x);

int N=0;

var que7\_2 = from y in w\_d

where (x.id == y.ID\_Dep)

select new { id\_w = y.ID\_W };

que7\_2 = que7\_2.Distinct();

foreach (var t2 in que7\_2)

foreach (var t in workers)

if (t2.id\_w == t.ID\_W)

N++;

Console.WriteLine("Number of workers " + N);

}

Console.WriteLine("\nPlease press any key to continue");

Console.Read();

}

}

}

Файл Department.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab7

{

public class Department

{

int id\_dep;

String name\_dep;

public Department(int i, string s)

{

id\_dep = i;

name\_dep = s;

}

public int ID\_Dep

{

get

{

return id\_dep;

}

set

{

id\_dep = value;

}

}

public String Name\_Dep

{

get

{

return name\_dep;

}

set

{

name\_dep = value;

}

}

public override string ToString()

{

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

}

}

}

Файл Workers.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab7

{

public class Worker

{

int id\_w;

String surname;

int id\_dep;

public Worker(int i, string s, int i\_d)

{

id\_w = i;

surname = s;

id\_dep = i\_d;

}

public int ID\_W

{

get

{

return id\_w;

}

set

{

id\_w = value;

}

}

public int ID\_Dep

{

get

{

return id\_dep;

}

set

{

id\_dep = value;

}

}

public String Surname

{

get

{

return surname;

}

set

{

surname = value;

}

}

public override string ToString()

{

return "(id=" + this.id\_w.ToString() + "; surname=" + this.surname + "; Department id=" + this.id\_dep + ")";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab7

{

public class workers\_department

{

int id\_w;

int id\_dep;

public workers\_department(int i, int i\_d)

{

id\_w = i;

id\_dep = i\_d;

}

public int ID\_W

{

get

{

return id\_w;

}

set

{

id\_w = value;

}

}

public int ID\_Dep

{

get

{

return id\_dep;

}

set

{

id\_dep = value;

}

}

public override string ToString()

{

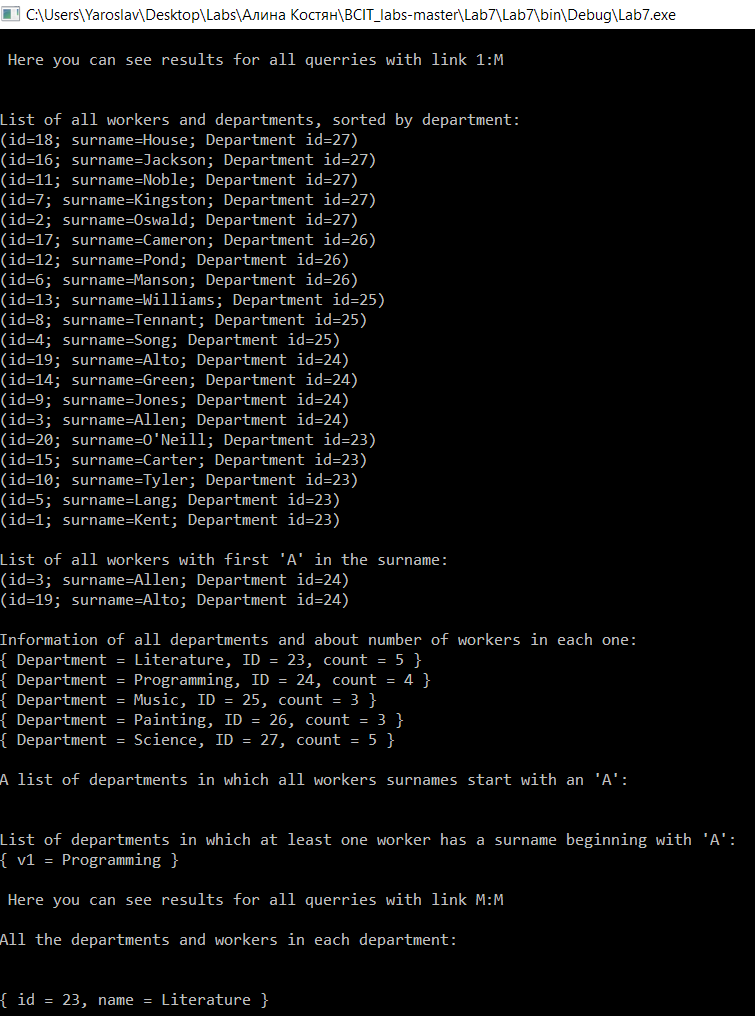
return "(Departmant=" + this.id\_dep.ToString() + "; Worker=" + this.id\_w.ToString() + ")";

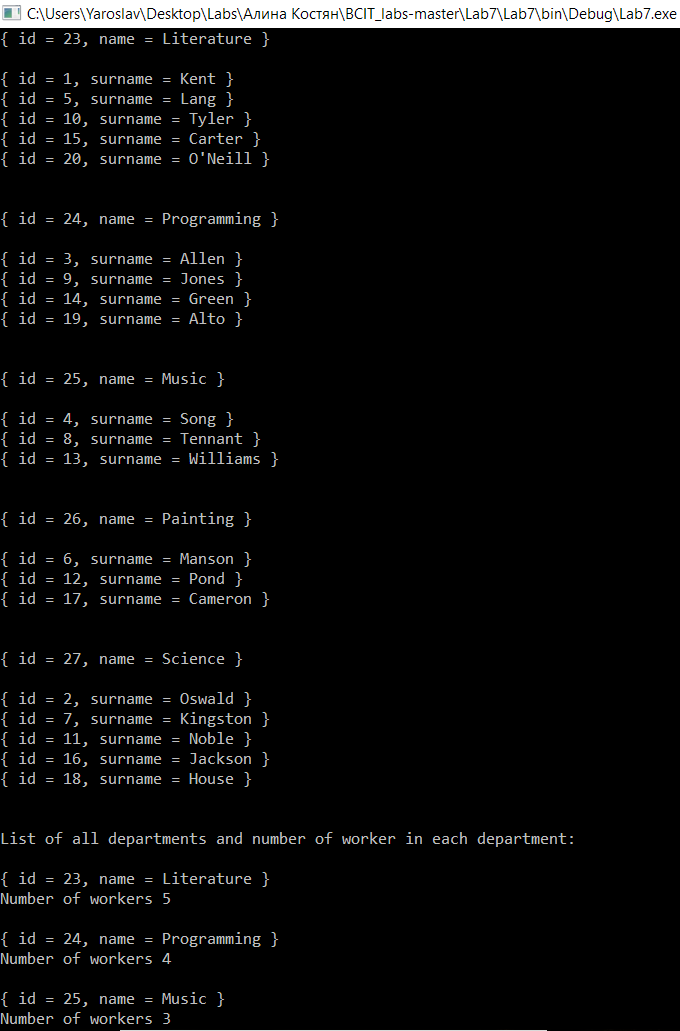
}

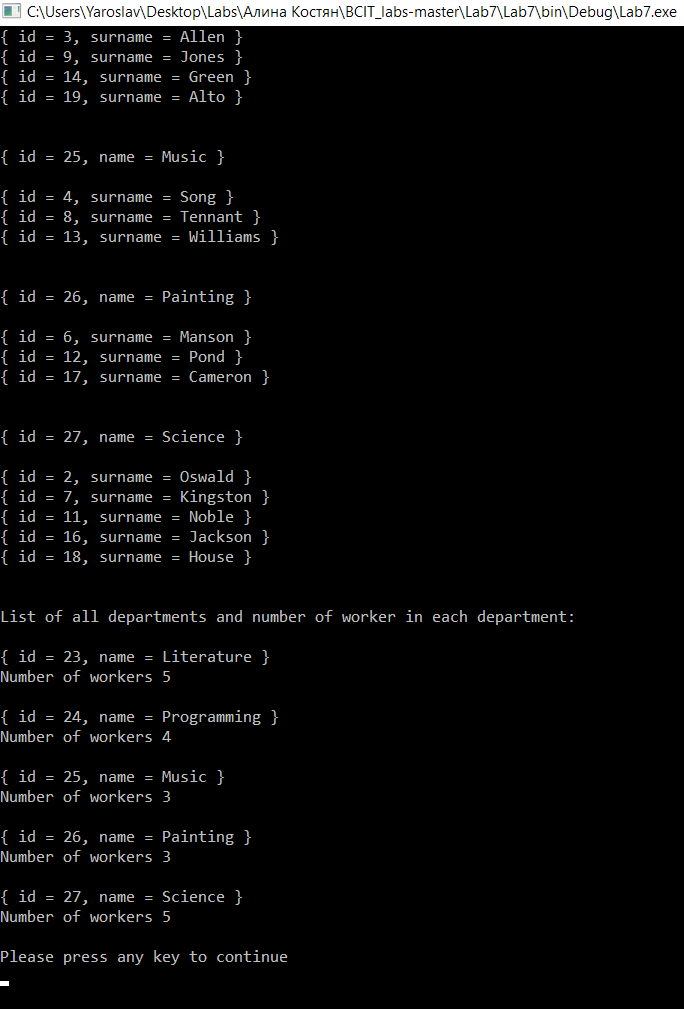
}

}

**Пример:**

****

****

****