Московский Государственный Университет им. Н.Э. Баумана

Отчет по лабораторной работе №7 по курсу БКИТ

Выполнил:

Саадуев Артур Сагид-Султанович

ИУ5-31

Условие задачи

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.

Код

Файл: Program.cs

```csharp
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),
```

```csharp
            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)
```

```csharp
                    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

```csharp
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

```csharp
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 + ")";
        }
    }
}
```

Файл: workers_department.cs

```csharp
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() + ")";
        }
    }
}
```

Примеры работающей программы


```
C:\Users\Lina\source\repos\BCIT_labs\Lab7\Lab7\bin\Debug\Lab7.exe

Here youe can see results for all querries with link 1:M


List of all workers and departments, sorted by department:
(id=18; surname=House; Department id=27)
(id=16; surname=Jackson; Department id=27)
(id=11; surname=Noble; Department id=27)
(id=7; surname=Kingston; Department id=27)
(id=2; surname=Oswald; Department id=27)
(id=17; surname=Cameron; Department id=26)
(id=12; surname=Pond; Department id=26)
(id=6; surname=Manson; Department id=26)
(id=13; surname=Williams; Department id=25)
(id=8; surname=Tennant; Department id=25)
(id=4; surname=Song; Department id=25)
(id=19; surname=Alto; Department id=24)
(id=14; surname=Green; Department id=24)
(id=9; surname=Jones; Department id=24)
(id=3; surname=Allen; Department id=24)
(id=20; surname=O'Neill; Department id=23)
(id=15; surname=Carter; Department id=23)
(id=10; surname=Tyler; Department id=23)
(id=5; surname=Lang; Department id=23)
(id=1; surname=Kent; Department id=23)

List of all workers with first 'A' in the surname:
(id=3; surname=Allen; Department id=24)
(id=19; surname=Alto; Department id=24)

Information of all departments and about number of workers in each one:
{ Department = Literature, ID = 23, count = 5 }
{ Department = Programming, ID = 24, count = 4 }
{ Department = Music, ID = 25, count = 3 }
{ Department = Painting, ID = 26, count = 3 }
{ Department = Science, ID = 27, count = 5 }

A list of departments in which all workers surnames start with an 'A':


List of departments in which at least one worker has a surname beginning with 'A':
{ v1 = Programming }
```

```
Here youe can see results for all querries with link M:M

All the departments and workers in each department:


{ id = 23, name = Literature }

{ id = 1, surname = Kent }
{ id = 5, surname = Lang }
{ id = 10, surname = Tyler }
{ id = 15, surname = Carter }
{ id = 20, surname = O'Neill }


{ id = 24, name = Programming }

{ id = 3, surname = Allen }
{ id = 9, surname = Jones }
{ id = 14, surname = Green }
{ id = 19, surname = Alto }


{ id = 25, name = Music }

{ id = 4, surname = Song }
{ id = 8, surname = Tennant }
{ id = 13, surname = Williams }


{ id = 26, name = Painting }

{ id = 6, surname = Manson }
{ id = 12, surname = Pond }
{ id = 17, surname = Cameron }


{ id = 27, name = Science }

{ id = 2, surname = Oswald }
{ id = 7, surname = Kingston }
{ id = 11, surname = Noble }
{ id = 16, surname = Jackson }
{ id = 18, surname = House }
```

```
List of all departments and number of worker in each department:

{ id = 23, name = Literature }
Number of workers 5

{ id = 24, name = Programming }
Number of workers 4

{ id = 25, name = Music }
Number of workers 3

{ id = 26, name = Painting }
Number of workers 3

{ id = 27, name = Science }
Number of workers 5

Please press any key to continue
```