

Московский государственный технический
университет имени Н. Э. Баумана.

Факультет “Информатика и системы управления”
Кафедра ИУ5 “Системы обработки информации и управления”

Курс “Парадигмы и конструкции языков
программирования”
Отчет по лабораторной работе №7.

Выполнила:

Студент группы ИУ5-31Б

Савельева Д.А

Подпись и дата:

Проверил:

Преподаватель кафедры ИУ5

Нардид А.Н.

Подпись и дата:

Москва, 2024 г.

Постановка 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.

Текст программы:

```
using System.ComponentModel.Design;
using System.Data.Common;
using System.Dynamic;

List<Worker> workers =
[
    new Worker(1, "Alice A", 3),
    new Worker(2, "Andrey M", 3),
    new Worker(3, "Mike A", 1),
    new Worker(4, "Stepan S", 2),
    new Worker(5, "Marina K", 1),
];

List<Department> deps =
[
    new Department(1, "management"),
```

```

        new Department(2, "it"),
        new Department(3, "design"),
        new Department(4, "marketing"),
    ];

List<Dep_worker> d_w = new List<Dep_worker>(workers.Count());
foreach (var wor in workers){
    d_w.Add(new Dep_worker(wor.Dep_id, wor.Work_id));
}

void Num1(){
    Console.WriteLine("\nNum1\n");
    var num1 = from w in workers from d in deps
                where w.Dep_id == d.Id
                orderby d.Name
                select new {w.Fio, d.Name};

    foreach (var item in num1){
        Console.WriteLine($"{item.Fio} works in {item.Name}");
    }
}

void Num2(){
    Console.WriteLine("\nNum2\n");
    var num2 = from w in workers
                where w.Fio.Split()[1].StartsWith('A')
                select new {w.Work_id, w.Fio};

    foreach (var item in num2){
        Console.WriteLine($"Worker number {item.Work_id}: {item.Fio}");
    }
}

void Num3(){ // через методы
    Console.WriteLine("\nNum3\n");
    var num3 = deps.GroupJoin(workers, // второй набор
                              d => d.Id, // свойство-селектор объекта из первого набора
                              w => w.Dep_id, // свойство-селектор объекта из второго набора
                              (d, w) => new // результат
                              {
                                  Name = d.Name,
                                  Workers = w
                              });

    foreach (var dep in num3)
    {
        Console.WriteLine($"In {dep.Name} - {dep.Workers.Count()} workers");
    }
}

```

```

}

void Num4(){
    Console.WriteLine("\nNum4\n");
    var num4 = deps.Where(d => workers.All(w => w.Dep_id == d.Id &&
w.Fio.StartsWith('A')))
        .Select(d => d.Name);

    foreach (var item in num4){
        Console.WriteLine($"Department {item}");
    }
}

void Num5(){ // имя на А
    Console.WriteLine("\nNum5\n");
    var num5 = deps.Where(d => workers.Any(w => w.Dep_id == d.Id &&
w.Fio.StartsWith('A')))
        .Select(d => d.Name);

    foreach (var item in num5){
        Console.WriteLine($"Department {item}");
    }
}

void Num6(){
    Console.WriteLine("\nNum6\n");
    var num6 = d_w.GroupBy(d => d.Dep_id);
    foreach(var dep in num6)
    {
        Console.WriteLine($"Department:{dep.Key}\nWorkers:");
        foreach(var wor in dep)
        {
            Console.WriteLine(wor.Work_id);
        }
    }
}

void Num7(){
    Console.WriteLine("\nNum7\n");
    var num7 = d_w.GroupBy(d => d.Dep_id);
    foreach(var dep in num7)
    {
        Console.WriteLine($"Department:{dep.Key}\nAmount of workers:");
        int c = 0;
        foreach(var wor in dep)
        {
            c++;
        }
    }
}

```

```

        Console.WriteLine(c);
    }
}

Num1();
Num2();
Num3();
Num4();
Num5();
Num6();
Num7();

class Worker
{
    public int Work_id {get;set;}
    public string Fio {get;set;}
    public int Dep_id {get;set;}

    public Worker(int id1, string fio, int id2)
    {
        Work_id = id1;
        Fio = fio;
        Dep_id = id2;
    }
}

class Department
{
    public int Id {get;set;}
    public string Name{get;set;}
    public Department(int id, string name){
        Id = id;
        Name = name;
    }
}

class Dep_worker
{
    public int Dep_id {get;set;}
    public int Work_id{get;set;}

    public Dep_worker(int id1,int id2){
        Dep_id = id1;
        Work_id = id2;
    }
}

```

Результат:

Num1	Num6
Alice A works in design	Department:3
Andrey M works in design	Workers:
Stepan S works in it	1
Mike A works in management	2
Marina K works in management	Department:1
Num2	Workers:
	3
Worker number 1: Alice A	5
Worker number 3: Mike A	Department:2
	Workers:
Num3	4
In management - 2 workers	Num7
In it - 1 workers	
In design - 2 workers	Department:3
In marketing - 0 workers	Amount of workers:
	2
Num4	Department:1
	Amount of workers:
	2
Num5	Department:2
	Amount of workers:
Department design	1