

Защищено:
Гапанюк Ю.Е.

"__" _____ 2023 г.

Демонстрация:
Казакова В.В.

"__" _____ 2023 г.

**Отчет по лабораторной работе № 4 по курсу
Парадигмы и конструкции языков программирования**

Тема работы: " Работа с LINQ to Objects"

6

(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-51Б

Казакова В.В.

(подпись)

"__" _____ 2023 г.

1. Описание задания

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.

2. Текст программы

```
using System;
using System.Collections.Generic;
using System.Linq;

class Program
{
    static void Main()
    {
        // Создаем коллекции сотрудников и отделов
        List<Отдел> отделы = new List<Отдел>
        {
            new Отдел { ID = 1, Наименование = "ИТ" },
            new Отдел { ID = 2, Наименование = "Маркетинг" },
            new Отдел { ID = 3, Наименование = "Финансы" }
        };

        List<Сотрудник> сотрудники = new List<Сотрудник>
        {
            new Сотрудник { ID = 1, Фамилия = "Александров", IDОтдела = 1 },
            new Сотрудник { ID = 2, Фамилия = "Белов", IDОтдела = 2 },
            new Сотрудник { ID = 3, Фамилия = "Андреев", IDОтдела = 1 },
            new Сотрудник { ID = 4, Фамилия = "Сидоров", IDОтдела = 3 },
            new Сотрудник { ID = 5, Фамилия = "Алексеев", IDОтдела = 2 }
        };

        // Запросы
        Console.WriteLine("Список всех сотрудников и отделов, отсортированный
по отделам:");
        var query1 = from сотрудник in сотрудники
                     join отдел in отделы on сотрудник.IDОтдела equals
отдел.ID
                     orderby отдел.ID
                     select new { Сотрудник = сотрудник, Отдел = отдел };
        foreach (var item in query1)
        {
            Console.WriteLine($"{item.Сотрудник.Фамилия} работает в отделе
{item.Отдел.Наименование}");
        }

        Console.WriteLine("\nСписок всех сотрудников, у которых фамилия
начинается с буквы 'А':");
        var query2 = from сотрудник in сотрудники
                     where сотрудник.Фамилия.StartsWith("А")
                     select сотрудник;
        foreach (var сотрудник in query2)
        {
            Console.WriteLine($"{сотрудник.Фамилия}");
        }

        Console.WriteLine("\nСписок всех отделов и количество сотрудников в
каждом отделе:");
        var query3 = from отдел in отделы
                     join сотрудник in сотрудники on отдел.ID equals
сотрудник.IDОтдела into grp
                     select new { Отдел = отдел, Количество = grp.Count() };
        foreach (var item in query3)
        {
            Console.WriteLine($"{item.Отдел.Наименование}: {item.Количество}
сотрудник(ов)");
        }
    }
}
```

```

    }

    Console.WriteLine("\nСписок отделов, в которых у всех сотрудников
фамилия начинается с буквы 'А':");
    var query4 = from отдел in отделы
                 join сотрудник in сотрудники on отдел.ID equals
сотрудник.IDОтдела
                 where сотрудники.All(s => s.Фамилия.StartsWith("А"))
                 select отдел;
    foreach (var отдел in query4)
    {
        Console.WriteLine($"{отдел.Наименование}");
    }

    Console.WriteLine("\nСписок отделов, в которых хотя бы у одного
сотрудника фамилия начинается с буквы 'А':");
    var query5 = from отдел in отделы
                 join сотрудник in сотрудники on отдел.ID equals
сотрудник.IDОтдела
                 where сотрудники.Any(s => s.Фамилия.StartsWith("А"))
                 select отдел;
    foreach (var отдел in query5)
    {
        Console.WriteLine($"{отдел.Наименование}");
    }

    // Много-ко-многим
    List<СотрудникиОтдела> сотрудникиОтдела = new List<СотрудникиОтдела>
    {
        new СотрудникиОтдела { IDCотрудника = 1, IDОтдела = 1 },
        new СотрудникиОтдела { IDCотрудника = 2, IDОтдела = 2 },
        new СотрудникиОтдела { IDCотрудника = 3, IDОтдела = 1 },
        new СотрудникиОтдела { IDCотрудника = 4, IDОтдела = 3 },
        new СотрудникиОтдела { IDCотрудника = 5, IDОтдела = 2 }
    };

    Console.WriteLine("\nСписок всех отделов и список сотрудников в каждом
отделе:");
    var query6 = from отдел in отделы
                 join сотрудникОтдела in сотрудникиОтдела on отдел.ID
equals сотрудникОтдела.IDОтдела into grp
                 select new { Отдел = отдел, Сотрудники = grp.Select(s =>
сотрудники.First(emp => emp.ID == s.IDСотрудника)) };
    foreach (var item in query6)
    {
        Console.WriteLine($"{item.Отдел.Наименование}: {string.Join(", ",
item.Сотрудники.Select(emp => emp.Фамилия))}");
    }

    Console.WriteLine("\nСписок всех отделов и количество сотрудников в
каждом отделе:");
    var query7 = from отдел in отделы
                 join сотрудникОтдела in сотрудникиОтдела on отдел.ID
equals сотрудникОтдела.IDОтдела into grp
                 select new { Отдел = отдел, Количество = grp.Count() };
    foreach (var item in query7)
    {
        Console.WriteLine($"{item.Отдел.Наименование}: {item.Количество}
сотрудник(ов)");
        Console.ReadKey();
    }
}

class Сотрудник
{

```

```
        public int ID { get; set; }
        public string Фамилия { get; set; }
        public int IDОтдела { get; set; }
    }

    class Отдел
    {
        public int ID { get; set; }
        public string Наименование { get; set; }
    }

    class СотрудникиОтдела
    {
        public int IDCотрудника { get; set; }
        public int IDОтдела { get; set; }
    }
```

3. Экранные формы

C:\Новая папка\Рабочий стол\Учебные материалы\ПиКЯП\LR4_Kazakova\bin\Debug\net6.0\LR4_Kazakova.exe

Список всех сотрудников и отделов, отсортированный по отделам:

Александров работает в отделе IT
Андреев работает в отделе IT
Белов работает в отделе Маркетинг
Алексеев работает в отделе Маркетинг
Сидоров работает в отделе Финансы

Список всех сотрудников, у которых фамилия начинается с буквы 'А':

Александров
Андреев
Алексеев

Список всех отделов и количество сотрудников в каждом отделе:

IT: 2 сотрудник(ов)
Маркетинг: 2 сотрудник(ов)
Финансы: 1 сотрудник(ов)

Список отделов, в которых у всех сотрудников фамилия начинается с буквы 'А':

Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы 'А':

IT
IT
Маркетинг
Маркетинг
Финансы

Список всех отделов и список сотрудников в каждом отделе:

IT: Александров, Андреев
Маркетинг: Белов, Алексеев
Финансы: Сидоров

Список всех отделов и количество сотрудников в каждом отделе:

IT: 2 сотрудник(ов)