|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю.Е.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |  | Демонстрация:  Гапанюк Ю.Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

**Отчет по лабораторной работе № 7 по курсу**

**Базовые компоненты интернет-технологий**

**ГУИМЦ**

#### Тема работы: " LINQ to Objects "

9

(количество листов)

Вариант № **4**

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Чиварзин А.Е. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

Москва, МГТУ - 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[1. Описание задания 3](#_Toc59094492)

[2. Диаграмма классов 3](#_Toc59094493)

[3. Текст программы 3](#_Toc59094494)

[4. Результаты выполнения программы 8](#_Toc59094495)

# 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.

# Диаграмма классов



# Текст программы

using System;

using System.Linq;

using System.Collections.Generic;

namespace HMW

{

public class Worker

{

public int ID;

public string LastName;

public int OtdelID;

public Worker(int id, string lastName, int otdelID)

{

this.ID = id;

this.LastName = lastName;

this.OtdelID = otdelID;

}

public override string ToString()

{

return $"(ID = {ID}; LastName = {LastName}; OtdelID = {OtdelID})";

}

}

public class Otdel

{

public int ID;

public string Name;

public Otdel(int id, string name)

{

this.ID = id;

this.Name = name;

}

public override string ToString()

{

return $"(ID = {ID}; Name = {Name})";

}

}

public class MkM\_Link

{

public int WorkerID;

public int OtdelID;

public MkM\_Link(int workerID, int OtdelID) => (WorkerID, OtdelID) = (workerID, OtdelID);

}

public class Program

{

static List<Worker> list\_worker = new List<Worker>()

{

new Worker(0, "Аникеев", 0),

new Worker(1, "Архангельский", 0),

new Worker(2, "Иванов", 0),

new Worker(3, "Петров", 1),

new Worker(4, "Сидоров", 1)

};

static List<Otdel> list\_otdel = new List<Otdel>()

{

new Otdel(0, "Главного механника"),

new Otdel(1, "Художественный"),

new Otdel(2, "Законодательный")

};

static readonly List<MkM\_Link> list\_MkM = new List<MkM\_Link>()

{

new MkM\_Link(0, 1),

new MkM\_Link(0, 0),

new MkM\_Link(1, 1),

new MkM\_Link(2, 0),

new MkM\_Link(2, 1),

new MkM\_Link(3, 0),

new MkM\_Link(4, 1)

};

static void sort\_by\_otdel()

{

Console.WriteLine("Cписок всех сотрудников и отделов, отсортированный по отделам:");

var res\_otdel = from otdel in list\_otdel orderby otdel.ID ascending select otdel;

foreach (var otdel in res\_otdel)

{

Console.WriteLine("Отдел " + otdel);

var workers = from worker in list\_worker where worker.OtdelID == otdel.ID select worker;

foreach (var worker in workers)

Console.WriteLine("\t" + worker);

}

}

static void Worker\_LastName\_FirstChar(char firstChar)

{

Console.WriteLine($"Список всех сотрудников, фамилия которых начинается на: {firstChar}");

var workers = from worker in list\_worker where worker.LastName[0] == firstChar select worker;

foreach (var worker in workers)

Console.WriteLine("\t" + worker);

}

static void Otdels\_workersCount()

{

Console.WriteLine("Список всех отделов и количество сотрудников в каждом отделе");

foreach (var otdel in list\_otdel)

{

Console.Write("\t" + otdel + " : ");

var workers = from worker in list\_worker where worker.OtdelID == otdel.ID select worker;

Console.WriteLine(workers.Count());

}

}

static void Otdels\_allWorkers\_beginChar(char firstChar)

{

Console.WriteLine($"Cписок отделов, в которых у всех сотрудников фамилия начинается с буквы {firstChar}");

foreach (var otdel in list\_otdel)

{

var workers = from worker in list\_worker where worker.OtdelID == otdel.ID select worker;

if (workers.All(worker => worker.LastName[0] == firstChar))

Console.WriteLine("\t" + otdel);

}

}

static void Otdels\_OnePlus\_FirstChar(char firstChar)

{

Console.WriteLine($"Cписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы {firstChar}");

foreach (var otdel in list\_otdel)

{

var workers = from worker in list\_worker where worker.OtdelID == otdel.ID select worker;

if (workers.Any(worker => worker.LastName[0] == firstChar))

Console.WriteLine($"\t{otdel}");

}

}

static void FullListMkM()

{

Console.WriteLine("Список всех отделов и список сотрудников в каждом отделе (связь много-ко-многим)");

foreach (var otdel in list\_otdel)

{

Console.WriteLine($"\t{otdel}:");

var workers = from worker in list\_MkM

from worker\_MkM in list\_worker

where worker.OtdelID == otdel.ID && worker.WorkerID == worker\_MkM.ID

select worker\_MkM;

foreach (var worker in workers)

Console.WriteLine($"\t\t{worker}");

}

}

private static void Otdel\_workersCount\_MkM()

{

Console.WriteLine("Список всех отделов и количество сотрудников в каждом отделе (связь много-ко-многим)");

foreach (var otdel in list\_otdel)

{

Console.Write("\t" + otdel + ": ");

var workers = from worker\_MkM in list\_MkM

from worker in list\_worker

where worker\_MkM.OtdelID == otdel.ID && worker\_MkM.WorkerID == worker.ID

select worker;

Console.WriteLine(workers.Count());

}

}

public static void Main()

{

Console.WriteLine("Чиварзин А. Е. ИУ5Ц-52Б\n");

sort\_by\_otdel();

Console.WriteLine();

Worker\_LastName\_FirstChar('А');

Console.WriteLine();

Otdels\_workersCount();

Console.WriteLine();

Otdels\_allWorkers\_beginChar('А');

Console.WriteLine();

Otdels\_OnePlus\_FirstChar('А');

Console.WriteLine("---------------------------------------------------------------------------------------");

FullListMkM();

Console.WriteLine();

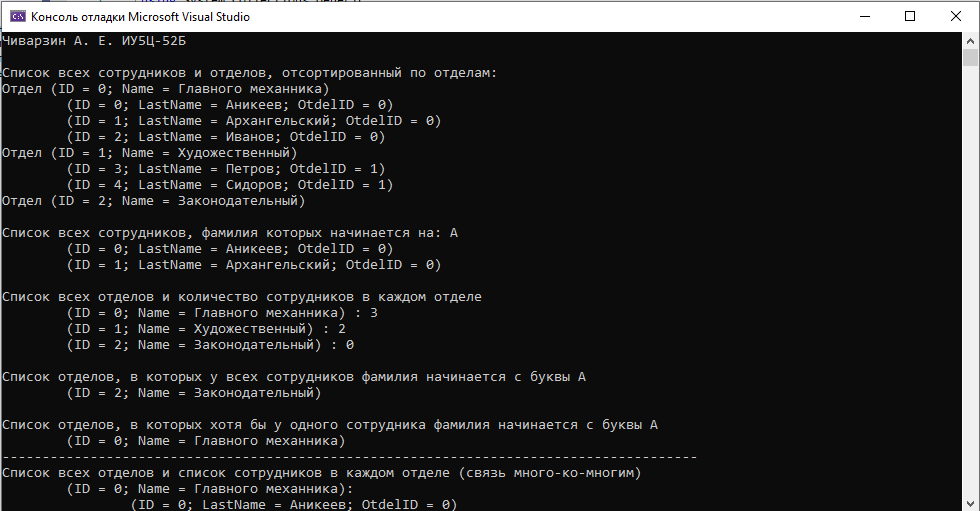
Otdel\_workersCount\_MkM();

}

}

}

# Результаты выполнения программы

****

Чиварзин А. Е. ИУ5Ц-52Б

Cписок всех сотрудников и отделов, отсортированный по отделам:

Отдел (ID = 0; Name = Главного механника)

(ID = 0; LastName = Аникеев; OtdelID = 0)

(ID = 1; LastName = Архангельский; OtdelID = 0)

(ID = 2; LastName = Иванов; OtdelID = 0)

Отдел (ID = 1; Name = Художественный)

(ID = 3; LastName = Петров; OtdelID = 1)

(ID = 4; LastName = Сидоров; OtdelID = 1)

Отдел (ID = 2; Name = Законодательный)

Список всех сотрудников, фамилия которых начинается на: А

(ID = 0; LastName = Аникеев; OtdelID = 0)

(ID = 1; LastName = Архангельский; OtdelID = 0)

Список всех отделов и количество сотрудников в каждом отделе

(ID = 0; Name = Главного механника) : 3

(ID = 1; Name = Художественный) : 2

(ID = 2; Name = Законодательный) : 0

Cписок отделов, в которых у всех сотрудников фамилия начинается с буквы А

(ID = 2; Name = Законодательный)

Cписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы А

(ID = 0; Name = Главного механника)

---------------------------------------------------------------------------------------

Список всех отделов и список сотрудников в каждом отделе (связь много-ко-многим)

(ID = 0; Name = Главного механника):

(ID = 0; LastName = Аникеев; OtdelID = 0)

(ID = 0; LastName = Аникеев; OtdelID = 0)

(ID = 1; LastName = Архангельский; OtdelID = 0)

(ID = 2; LastName = Иванов; OtdelID = 0)

(ID = 2; LastName = Иванов; OtdelID = 0)

(ID = 3; LastName = Петров; OtdelID = 1)

(ID = 4; LastName = Сидоров; OtdelID = 1)

(ID = 1; Name = Художественный):

(ID = 2; Name = Законодательный):

Список всех отделов и количество сотрудников в каждом отделе (связь много-ко-многим)

(ID = 0; Name = Главного механника): 7

(ID = 1; Name = Художественный): 0

(ID = 2; Name = Законодательный): 0

A:\sasha\YandexDisk\YandexDisk\МГТУ\5-й семестр\Основные компоненты интернет-технологий\ЛР\VS\LAB\_7\_asDZ\LAB\_7asDZ\bin\Debug\netcoreapp3.1\LAB\_7asDZ.exe (процесс 7324) завершил работу с кодом 0.

Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".

Нажмите любую клавишу, чтобы закрыть это окно…