Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Язык LINQ и методы-расширения

тема

Вариант 11 (8+)

|  |  |  |
| --- | --- | --- |
| Преподаватель |  | А. А. Чикизов |
|  | подпись, дата | инициалы, фамилия |
| Студент КИ18-16б 031831229 |  | В. А. Прекель |
| номер группы, зачетной книжки | подпись, дата | инициалы, фамилия |

Красноярск 2020

# 1 Задание

Варианты 8+

Взять из Лабораторной работы 4, переделать задание по аналогии из варианта 1-7

Внутри класса обязательно должен быть массив, должен быть метод-расширения который обрабатывает элементы данного массива, сам метод должен быть использован в одном из запросов. Так же должен быть второй класс, у которого есть общее поле с первым. После заполнения данных показать, как вы научились использовать LINQ при работе с перечисляемыми типами. Собственные доработки и демонстрация более глубоких знаний приветствуются.

# 2 Исходный код основного алгоритма

Листинг 1 – CSharpLabs.Lab05\CSharpLabs.Lab05.Console\Program.cs

п»їusing System;  
using System.Linq;  
using System.Text;  
  
using CSharpLabs.Lab05.Core;  
  
Console.OutputEncoding = Encoding.UTF8;  
var random = new Random();  
  
var services = Enumerable.Range(1, 10)  
 .Select(id => new Service($"РЎРµСЂРІРёСЃ в„–{id}", id))  
 .ToList();  
  
var cars = new[] {"Volkswagen", "BMV", "Audi", "Mercedes-Benz", "Porsche", "Toyota", "Nissan", "Renault"}  
 .Select(manufacturer => new Car(  
 manufacturer,  
 random.Next(1, services.Count + 1),  
 Enumerable.Range(0, 5)  
 .Select(\_ => random.Next(1000000000) / 100.0)  
 .ToArray()  
 )  
 ).ToList();  
Console.WriteLine(String.Join("\n", services));  
Console.WriteLine();  
  
Console.WriteLine(String.Join("\n", cars));  
Console.WriteLine();  
  
var q1 = cars  
 .Join(services, car => car.ServiceId, service => service.ServiceId,  
 (car, service) => new {car, service})  
 .OrderBy(carservice => carservice.car.Manufacturer);  
Console.WriteLine(String.Join("\n", q1));  
Console.WriteLine();  
  
var q2 = services  
 .GroupJoin(cars, service => service.ServiceId, car => car.ServiceId,  
 (service, carsInService) => new  
 {Service = service, Cars = carsInService.OrderByDescending(car => car.AveragePrice())})  
 .Select(serviceWithCars =>  
 $"{serviceWithCars.Service} Cars = [{String.Join("; ", serviceWithCars.Cars.Select(car => $"{car.Manufacturer} {car.AveragePrice()}"))}]");  
Console.WriteLine(String.Join("\n", q2));  
Console.WriteLine();

Листинг 2 – CSharpLabs.Lab05\CSharpLabs.Lab05.Core\Car.cs

using System;  
using System.Collections.Generic;  
  
namespace CSharpLabs.Lab05.Core  
{  
 public record Car(string Manufacturer, int ServiceId, double[] Prices)  
 {  
 public override string ToString() =>  
 $"Car {{ Manufacturer = {Manufacturer}, ServiceId = {ServiceId}, Prices = [{String.Join("; ", Prices)}] }}";  
 }  
}

Листинг 3 – CSharpLabs.Lab05\CSharpLabs.Lab05.Core\CarExtension.cs

using System;  
using System.Linq;  
  
namespace CSharpLabs.Lab05.Core  
{  
 public static class CarExtension  
 {  
 public static double AveragePrice(this Car car) => Math.Round(car.Prices.Average() \* 100) / 100;  
 }  
}

Листинг 4 – CSharpLabs.Lab05\CSharpLabs.Lab05.Core\Service.cs

namespace CSharpLabs.Lab05.Core  
{  
 public record Service(string Name, int ServiceId);  
}