**Теория по LINQ:**

**<https://professorweb.ru/my/LINQ/base/level1/linq_index.php>**

**LINQ**

Код основного примера

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Person

{

public string Name {set;get;}

public int Age {set;get;}

public Person()

{

Name = "Вася";

Age = 21;

}

public Person (string s, int n)

{

Name = s;

Age = n;

}

}

class Program

{

static void Main(string[] args)

{

#region[Пример 1. Базовый запрос. Отложенное выполнение]

//int[] numbers = { 1, 5, -7, 3, 5, -2, -8, -9, 0 };

//Console.WriteLine("Исходный набор чисел");

//foreach (int n in numbers)

// Console.Write(n + " ");

//var positive = from n in numbers

// where n > 0

// select n;

//Console.WriteLine("\n\nПоложительные числа");

//foreach (int p in positive)

// Console.Write(p + " ");

#endregion

#region[Пример 2. Методы расширения]

//int[] numbers = { 1, 2, 3, 4, 10, 34, 55, 66, 77, 88 };

// Console.WriteLine("Исходный набор чисел");

// foreach (int n in numbers)

// Console.Write(n + " ");

//IEnumerable<int> evens = from i in numbers

// where i % 2 == 0 && i > 10

// select i;

////Аналогично

////int[] numbers = { 1, 2, 3, 4, 10, 34, 55, 66, 77, 88 };

////IEnumerable<int> evens = numbers.Where(i => i % 2 == 0 && i > 10);

//Console.WriteLine("\n\nЧетные числа больше 10");

//foreach (int i in evens)

// Console.Write(i + " ");

#endregion

#region[Пример 3. Выборка из объектов]

//List<Person> People = new List<Person>{new Person(), new Person("Надя",26),

// new Person("Маша", 17), new Person("Андрей", 44),

// new Person("Инокентий", 29), new Person("Зина",23),

// new Person("Миша",7), new Person("Маша",13)};

//Console.WriteLine("Исходный набор людей\n");

//foreach (Person p in People)

//{

// Console.WriteLine("{0} - {1}", p.Name, p.Age);

//}

//var adults = from a in People

// where a.Age >= 18

// select a;

//Console.WriteLine("\nВсе взрослые (18+)\n");

//foreach (Person p in adults)

//{

// Console.WriteLine("{0} - {1}", p.Name, p.Age);

//}

#endregion

#region[Пример 4. Сортировка и множественная сортировка]

//List<Person> People = new List<Person>{new Person(), new Person("Надя",26),

// new Person("Маша", 17), new Person("Андрей", 44),

// new Person("Инокентий", 29), new Person("Зина",23),

// new Person("Миша",7), new Person("Маша",13)};

//Console.WriteLine("Исходный набор людей\n");

//foreach (Person p in People)

//{

// Console.WriteLine("{0} - {1}", p.Name, p.Age);

//}

//var sortPeople = from p in People

// orderby p.Name //, p.Age

// select p;

//Console.WriteLine("\nОтсортированные\n");

//foreach (Person p in sortPeople)

//{

// Console.WriteLine("{0} - {1}", p.Name, p.Age);

//}

#endregion

#region[Пример 5. Работа со множествами]

//string[] soft = { "Microsoft", "Google", "Apple" };

//string[] hard = { "Apple", "IBM", "Samsung" };

//// разность множеств

//var result = soft.Except(hard);//разность

////var result = soft.Intersect(hard);//пересечение

////var result = soft.Union(hard);//объединение

////var result = soft.Concat(hard);//объединение с повторением

//foreach (string s in result)

// Console.WriteLine(s);

#endregion

#region[Пример 6. Общие вычисления для выборки. Немедленное выполнение]

//List<Person> People = new List<Person>{new Person(), new Person("Надя",26),

// new Person("Маша", 17), new Person("Андрей", 44),

// new Person("Инокентий", 29), new Person("Зина",23),

// new Person("Миша",7), new Person("Маша",13)};

//Console.WriteLine("Исходный набор людей\n");

//foreach (Person p in People)

//{

// Console.WriteLine("{0} - {1}", p.Name, p.Age);

//}

//int totalAge = People.Sum(p => p.Age);

//int minAge = People.Min(p => p.Age);

//int maxAge = People.Max(p => p.Age);

//double averAge = People.Average(p => p.Age);

//Console.WriteLine("Общий возраст {0}, Минимальный возраст {1}, Максимальный возраст {2}, Средний возраст {3}", totalAge, minAge, maxAge, averAge);

#endregion

//методы Skip и Take пропускают/извлекают несколько элементов из выборки

#region[Пример 7. Группировка]

//List<Person> People = new List<Person>{new Person(), new Person("Игорь",20),

// new Person("Витя", 20), new Person("Настя", 18),

// new Person("Катя", 18), new Person("Женя",20),

// new Person("Оксана",10), new Person("Костя",10)};

//Console.WriteLine("Исходный набор людей\n");

//foreach (Person p in People)

//{

// Console.WriteLine("{0} - {1}", p.Name, p.Age);

//}

//var ageGroups = from p in People

// group p by p.Age;

//Console.WriteLine("\nГруппировка\n");

//foreach (var g in ageGroups)

//{

// Console.WriteLine(g.Key);

// foreach (var t in g)

// Console.WriteLine(t.Name);

// Console.WriteLine();

//}

#endregion

Console.ReadKey();

}

}

}