

# Речници, ламбда изрази и LINQ

Колекции и заявки



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>

<https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Fundamentals>





# LINQ

$f(x) \rightarrow y$

## Ламбда функции и LINQ

LINQ в действие: филтриране,  
разпределяне, подреждане

# Обработка на поредици с LINQ

- **Min()** – намира **най-малкия** елемент в колекция

```
new List<int>() { 1, 2, 3, 4, -1, -5, 0, 50 }.Min() → -5
```

- **Max()** – намира **най-големия** елемент в колекция

```
new int[] { 1, 2, 3, 40, -1, -5, 0, 5 }.Max() → 40
```

- **Sum()** – намира **сумата** на всички елементи в колекция

```
new long[] {1, 2, 3, 4, -1, -5, 0, 50}.Sum() → 54
```

- **Average()** – намира **средноаритметичното** на всички елементи

```
new int[] {1, 2, 3, 4, -1, -5, 0, 50}.Average() → 6.75
```

# Задача: Sum, Min, Max, Average

- Напишете програма, която въвежда **n** цели числа и извежда техните **сума**, **минимум**, **максимум** и **средноаритметично**:

5  
12  
20  
-5  
37  
8



Sum = 72  
Min = -5  
Max = 37  
Average = 14.4

4  
50  
20  
25  
40



Sum = 135  
Min = 20  
Max = 50  
Average = 33.75



# Решение: Sum, Min, Max, Average

```
using System.Linq;
```

```
...
```

```
int n = int.Parse(Console.ReadLine());
```

```
int[] nums = new int[n];
```

```
for (int i = 0; i < n; i++)
```

```
    nums[i] = int.Parse(Console.ReadLine());
```

```
Console.WriteLine("Sum = {0}", nums.Sum());
```

```
Console.WriteLine("Min = {0}", nums.Min());
```

```
// TODO: изведете максимум и средноаритметично
```

Добавете **System.Linq**, за да може да използваме LINQ функциите като **.Max()** и **.Sum()**

# Въвеждане на колекции от един ред

- Използвайте **Select()** за въвеждане на колекции:

```
var nums = Console.ReadLine()  
    .Split()  
    .Select(number => double.Parse(number));  
// .Select(double.Parse); // къса версия
```

```
var nums = Console.ReadLine()  
    .Split()  
    .Select(int.Parse);  
// .Select(number => int.Parse(number)); // дълга  
версия
```

# Преобразуване на колекции

- Използвайте **ToArray()**, **ToList()** за преобразуване:

```
int[] nums = Console.ReadLine()  
    .Split()  
    .Select(number => int.Parse(number))  
    .ToArray();
```

```
List<double> nums = Console.ReadLine()  
    .Split()  
    .Select(double.Parse)  
    .ToList();
```

# Сортиране на колекции

- Чрез **OrderBy()** сортираме в нарастващ ред:

```
List<int> nums = { 1, 5, 2, 4, 3 };  
nums = nums  
    .OrderBy(num => num)  
    .ToList();
```

- Чрез **OrderByDescending()** сортираме в намалящ ред:

```
List<int> nums = { 1, 5, 2, 4, 3 };  
nums = nums.OrderByDescending(num => num).ToList();  
Console.WriteLine(String.Join(", ", nums));
```



# Сортиране на колекции по няколко признака

- Чрез **ThenBy()** можем да сортираме по няколко признака:

```
Dictionary<int, string> products =  
    new Dictionary<int, string>();  
  
Dictionary<int, string> sortedDict = products  
    .OrderBy(pair => pair.Value)  
    .ThenBy(pair => pair.Key)  
    .ToDictionary(pair => pair.Key, pair => pair.Value);
```

# Вземане / Пропускане на елементи от колекция

- Чрез **Take()** можем да вземем определен брой елементи:

```
var nums = new List<int>() { 10, 20, 30, 40, 50, 60}  
    .Take(3)  
    .ToArray();  
// nums = [10, 20, 30]
```

- Чрез **Skip()** можем да пропуснем определен брой:

```
var nums = new List<int>() { 10, 20, 30, 40, 50, 60}  
    .Skip(3).Take(2)  
    .ToArray();  
// nums = [40, 50]
```

# Задача: Най-големите 3 числа

- Въведете списък от реални числа и изведете най-големите 3 от тях

10 30 15 20 50 5 → 50 30 20

20 30 → 30 20

0 -5 -1 -3 -2 → 0 -1 -2

# Задача: Най-големите 3 числа

```
List<int> nums = Console.ReadLine().Split()  
    .Select(int.Parse)  
    .ToList();  
  
var sortedNums = nums.OrderByDescending(x => x);  
  
var largest3Nums = sortedNums.Take(3);  
  
Console.WriteLine(string.Join(" ", largest3Nums));
```



# Какво научихме днес?

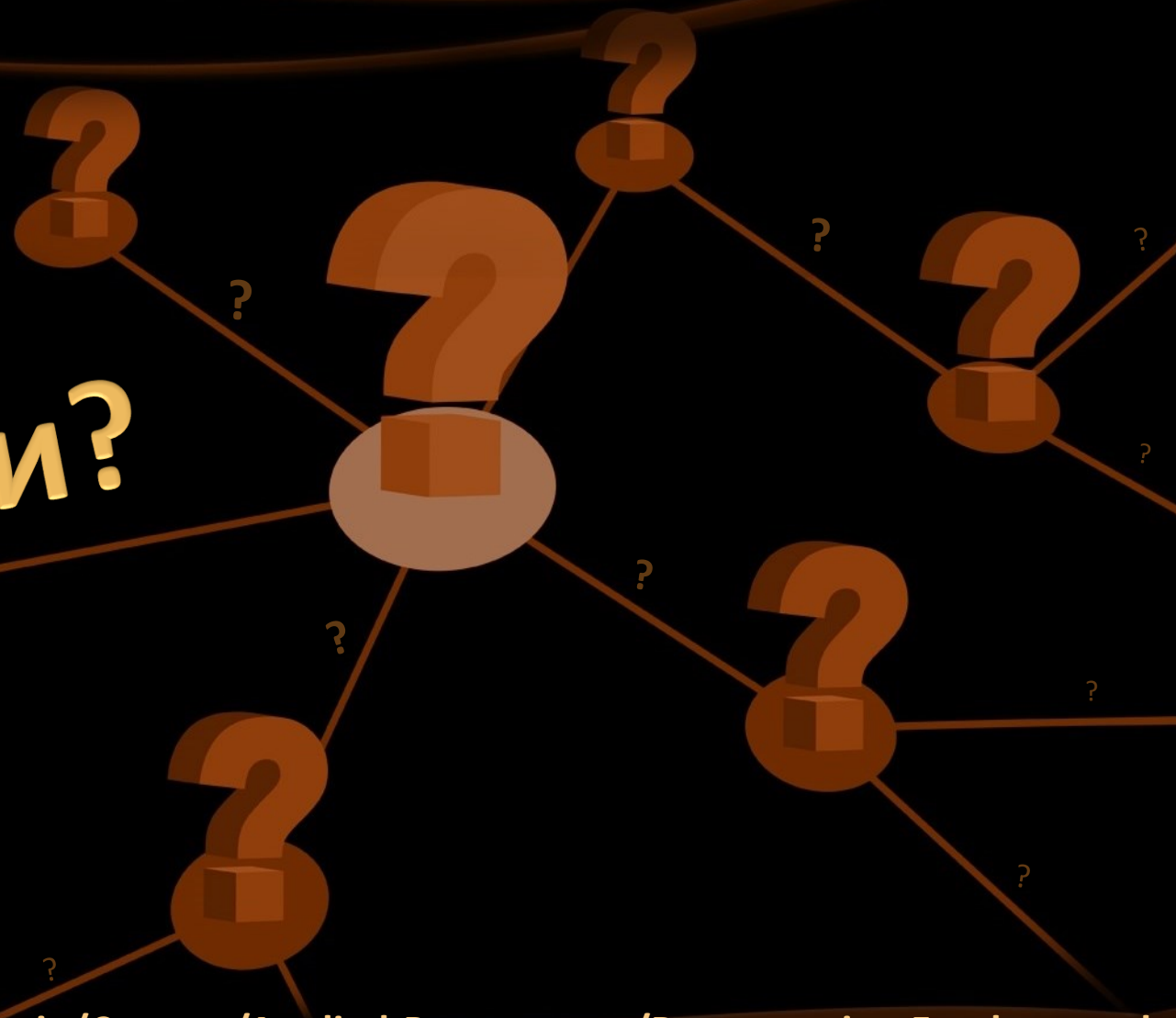
- LINQ сериозно опростява обработката на данни
- Част от поддържаните операции са:
  - Намиране на минимум/максимум
  - Сумиране
  - Средноаритметично
  - Сортиране
  - Извличане на няколко елемента и др.



# Речници, ламбда изрази и LINQ



Въпроси?



# Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма **"Обучение за ИТ кариера"** на МОН за подготовка по професия "Приложен програмист"



Министерство  
на образованието  
и науката



Национална  
програма  
„Обучение за  
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni  
Foundation

