

Речници, ламбда изрази и LINQ

Колекции и заявки

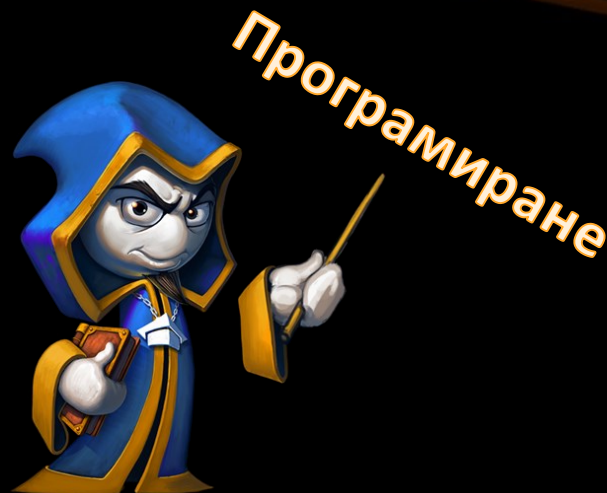


Учителски екип

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>



Ламбда Изрази

- Ламбда изразите са **анонимна функция** съдържаща изрази и твърдения

```
var lambda = (a => a > 5);
```

- Ламбда изразите
 - Използват ламбда оператор **=>**
 - Чете се като „**води към**“
 - **Лявата страна** описва **входните** параметри
 - **Дясната страна** описва **израза** или **твърдението**

Ламбда функции

- Ламбда функциите са внедрени методи (функции) които вземат входни параметри и връщат стойности:

$x \Rightarrow x / 2$



```
static int Func(int x) { return x / 2; }
```

$x \Rightarrow x \neq 0$



```
static bool Func(int x) { return x != 0; }
```

$() \Rightarrow 42$



```
static int Func() { return 42; }
```

$(x, y) \Rightarrow x+y$



```
static int Func(int x, int y)  
{ return x+y; }
```

Филтриране на колекции

- Чрез **Where()**, **Count()**:

```
int[] nums = { 1, 2, 3, 4, 5, 6};  
nums = nums  
    .Where(num => num % 2 == 0)  
    .ToArray();  
// nums = [2, 4, 6]
```

```
int[] nums = { 1, 2, 3, 4, 5, 6};  
int count = nums.Count(num => num % 2 == 0);  
// count = 3
```

Филтриране и сортиране с Ламбда функции

```
int[] nums = { 11, 99, 33, 55, 77, 44, 66, 22, 88 };
```

```
nums.OrderBy(x => x).Take(3);
```

```
// 11 22 33
```

```
nums.Where(x => x < 50);
```

```
// 11 33 44 22
```

```
nums.Count(x => x % 2 == 1);
```

```
// 5
```

```
nums.Select(x => x * 2).Take(5);
```

```
// 22 198 66 110 154
```


Извличане на уникални елементи от колекция

- **Distinct()** извлича **уникалните** елементи от колекция:

```
int[] nums = { 1, 2, 2, 3, 4, 5, 6, -2, 2, 0,  
15, 3, 1, 0, 6 };
```

```
nums = nums  
    .Distinct()  
    .ToArray();
```

```
// nums = [1, 2, 3, 4, 5, 6, -2, 0, 15]
```

Задача: Сортиране на кратки думи

- Въведете **текст**, извличете неговите **думи**, намерете всички **кратки думи** (с по-малко от 5 знака) и ги изведете в **азбучен ред**, с **малки букви**
 - Използвайте следните разделители: **. , : ; () [] " ' / \ ! ?**
(интервал)
 - Засичайте без значение от големината на буквите; премахнете дублиранията

In SoftUni you can study Java, C#, PHP and JavaScript.
JAVA and c# developers graduate in 2-3 years. Go in!

2-3, and, c#, can, go, in, java, php, you

Тествайте в Judge: <https://judge.softuni.bg/Contests/2671>

Решение: Сортиране на кратки думи

```
char[] separators = new char[]
{'.', ',', ':', ';', '(', ')', '[', ']', '\\', '\"', '\'', '/', '!', '?', ' '};
string sentence = Console.ReadLine().ToLower();
string[] words = sentence.Split(separators);
var result = words
    .Where(w => w != "")
    // TODO: филтрирайте по дължина < 5
    .OrderBy(w => w).Distinct();
Console.WriteLine(string.Join(", ", result));
```


Извличане на един елемент от колекция

- Чрез **First()**, **Last()**, **Single()**:

```
int[] nums = { 1, 2, 3, 4, 5, 6 };
```

```
int firstNum = nums.First(x => x % 2 == 0); // 1
```

```
int lastNum = nums.Last(x => x % 2 == 1); // 6
```

```
int singleNum = nums.Single(x => x == 4); // 4
```

Други операции на колекции

- Обръщане на колекцията чрез **Reverse()**

```
int[] nums = { 1, 2, 3, 4, 5, 6};  
nums = nums.Reverse();  
// nums = 6, 5, 4, 3, 2, 1
```

- Сляпане чрез **Concat()**:

```
int[] nums = { 1, 2, 3, 4, 5, 6 };  
int[] otherNums = { 7, 8, 9, 0 };  
nums = nums.Concat(otherNums);  
// nums = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0
```

Задача: Сгъни и сумирай

- Въведете масив от $4*k$ цели числа, сгънете го както е показано и изведете сумата от горните и долните редове ($2*k$ цели числа):

1 2 3 4 5 6 7 8

2 1 8 7
3 4 5 6

5 2 3 6 → 5 6
2 3 → 7 9

1 2 3 4 5 6 7 8 → 2 1 8 7
3 4 5 6 → 5 5 13 13

4 3 -1 2 5 0 1 9 8 6 7 -2 → -1 3 4 -2 7 6
2 5 0 1 9 8 → 1 8 4 -1 16 14

Тествайте в Judge: <https://judge.softuni.bg/Contests/2671>

Решение: Сгъни и сумирай

```
int[] arr = Console.ReadLine()
    .Split(' ').Select(int.Parse).ToArray();
int k = arr.Length / 4;

int[] row1left = arr.Take(k).Reverse().ToArray();
int[] row1right = arr.Reverse().Take(k).ToArray();
int[] row1 = row1left.Concat(row1right).ToArray();
int[] row2 = arr.Skip(k).Take(2 * k).ToArray();

var sumArr =
    row1.Select((x, index) => x + row2[index]);
Console.WriteLine(string.Join(" ", sumArr));
```

Какво научихме този час?

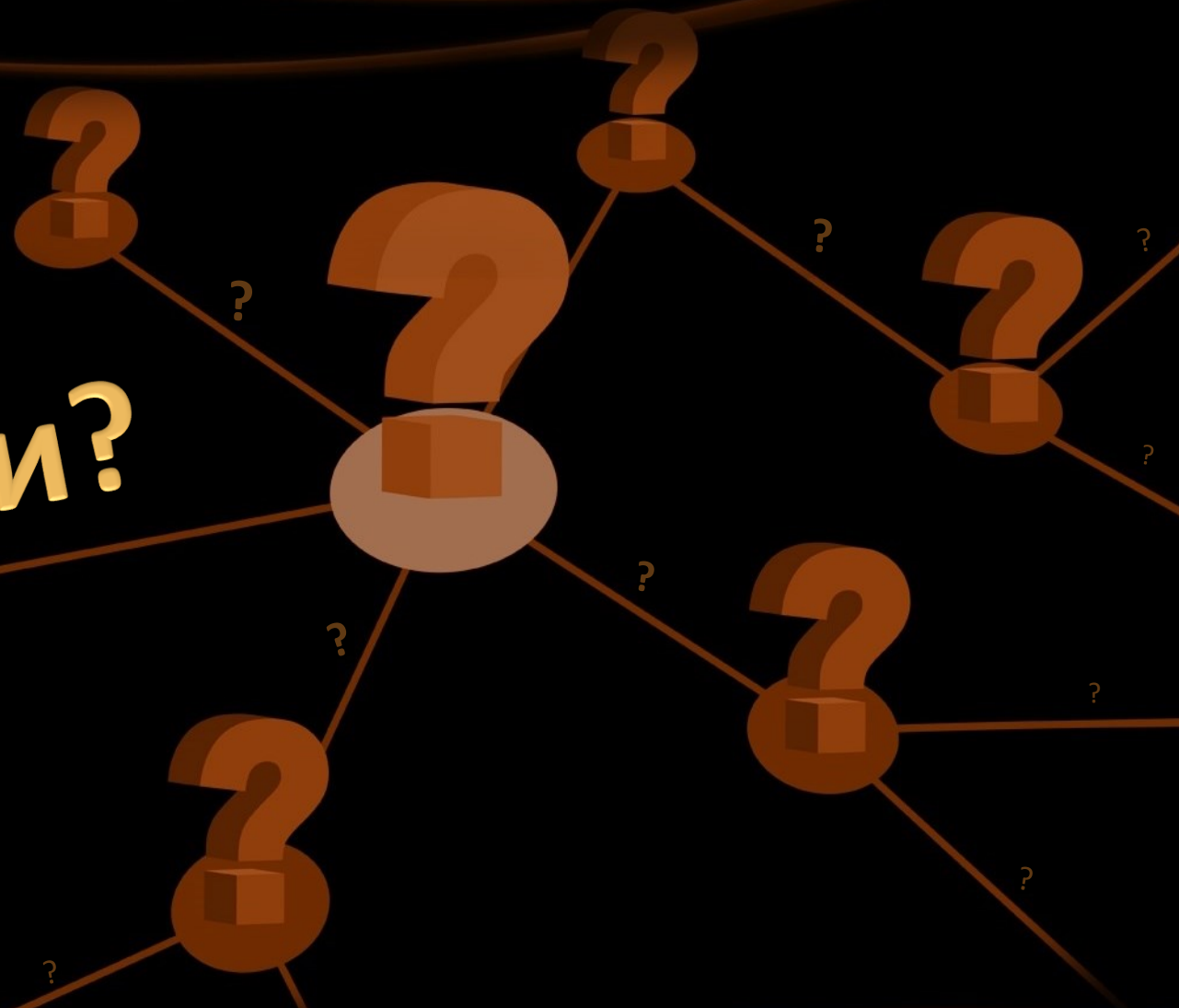
- Ламбда изразите напомнят функциите: лявата страна описва параметрите, а дясната – израза или твърдението
- Полезни са за компактно задаване на критерии при извличане на данни, сортировка и др.



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

