

# Речници, ламбда изрази и LINQ

Колекции и заявки

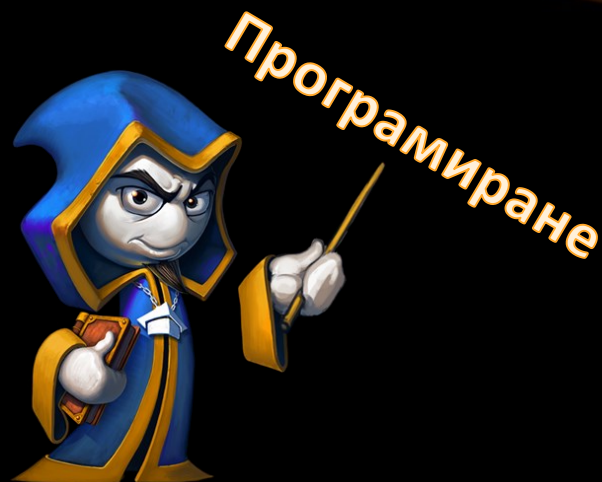


Учителски екип

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>



# SortedDictionary<K, V>

- Сортирани речници
  - Използват балансирано дърво за претърстване
  - SortedDictionary<K, V>
  - Пазят ключовете си сортирани в техния естествен ред

```
var sortedDict = new SortedDictionary<int,int>();
```

# Речници: Функционалност

- **Count** – пази броя на двойките от ключ-стойност
- **Keys** – съдържа уникалните ключове

```
var dict = new SortedDictionary<string, int>();  
foreach(var key in dict.Keys)  
    Console.WriteLine(key);
```

- **Values** – съдържа всички стойности

```
Console.WriteLine(String.Join(", ", dict.Values));
```

- Основни операции: **Add()**, **[], Remove()**, **Clear()**


## Речници: Функционалност (2)

- Намиране на ключ / стойност:
  - **ContainsKey()** – проверяваме дали даден **ключ** съществува в речника (бърза операция)
  - **ContainsValue()** – проверяваме дали дадена **стойност** съществува в речника (бавна операция)
  - **TryGetValue()** – проверяваме дали даден **ключ** съществува в речника и отпечатва стойността му

# SortedDictionary<K, V> – Пример

Palco	+388 892 55802
-------	----------------

Comparator  
Function



SortedDictionary <string, string>	

Ключ

Стойность



# Пример: SortedDictionary – Събития

```
var events = new SortedDictionary<DateTime, string>();
events[new DateTime(1998, 9, 4)] = "Google's birth date";
events[new DateTime(2013, 11, 5)] = "SoftUni's birth date";
events[new DateTime(1975, 4, 4)] = "Microsoft's birth date";
events[new DateTime(2004, 2, 4)] = "Facebook's birth date";
events[new DateTime(2013, 11, 5)] = "SoftUni was founded";
foreach (var entry in events)
{
    Console.WriteLine("{0:dd-MMM-yyyy}: {1}",
        entry.Key, entry.Value);
}
```

# Задача: Пребройте реалните числа

- Въведете списък от реални числа и ги изведете в нарастващ ред заедно с техния брой срещания

8 2.5 2.5 8 2.5



2.5 -> 3 times  
8 -> 2 times

1.5 5 1.5 3



1.5 -> 2 times  
3 -> 1 times  
5 -> 1 times

-2 0.33 0.33 2



-2 -> 1 times  
0.33 -> 2 times  
2 -> 1 times

Тествайте в Judge: <https://judge.softuni.bg/Contests/2670>

# Решение: Пребройте реалните числа

```
double[] nums = Console.ReadLine().Split(' ')
    .Select(double.Parse).ToArray();

var counts = new SortedDictionary<double, int>();
foreach (var num in nums)
    if (counts.ContainsKey(num))
        counts[num]++;
    else
        counts[num] = 1;

foreach (var num in counts.Keys)
    Console.WriteLine($"{num} -> {counts[num]}");
```

**counts[num]** пази  
колко пъти **num** се  
среща в **nums**

Тествайте в Judge: <https://judge.softuni.bg/Contests/2670>



# Какво научихме този час?

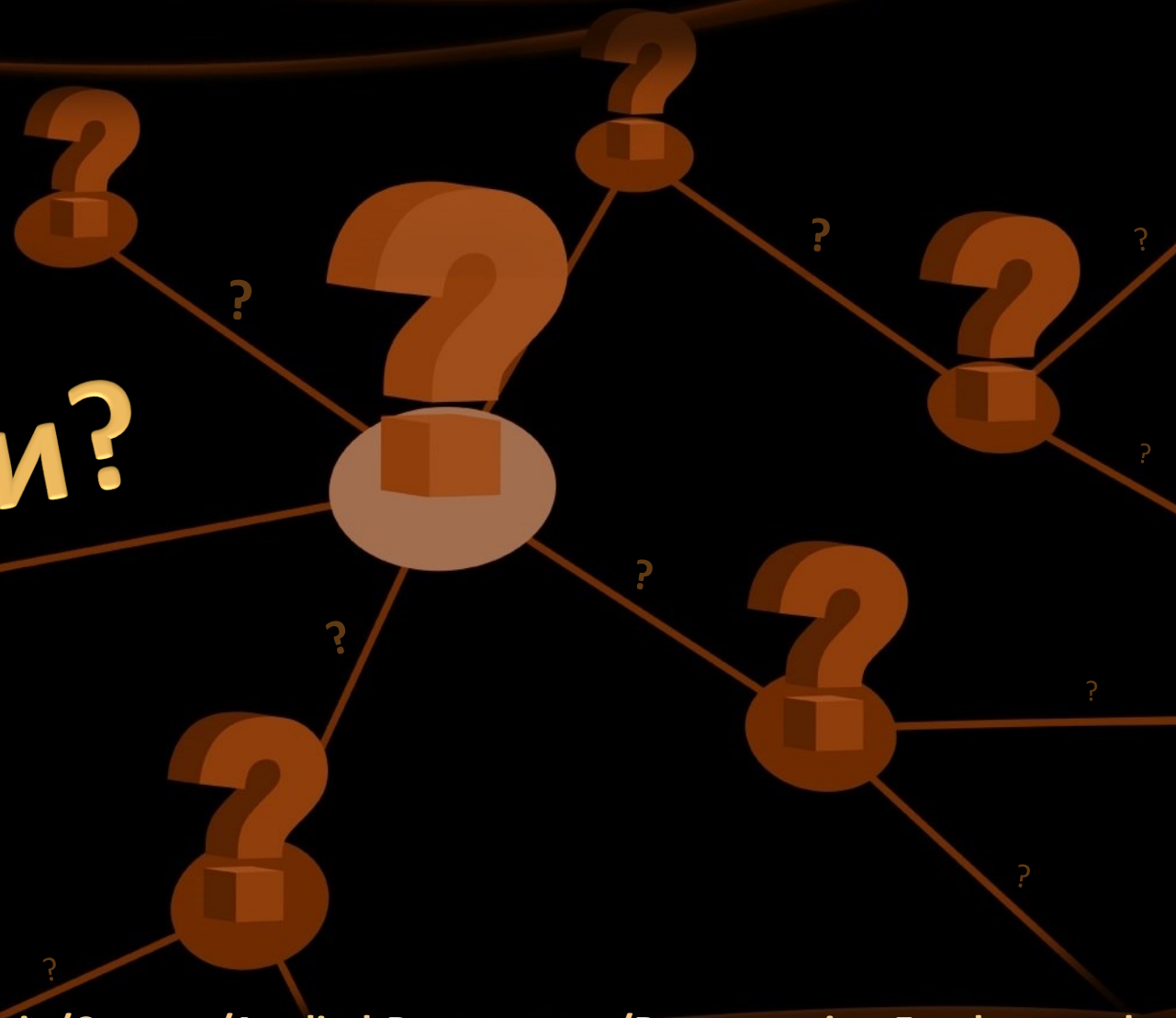
- `Dictionary<K, V>` с/у  
`SortedDictionary<K, V>` - елементите в обикновения речник се пазят по ред на добавяне, докато в сортирания – редът на добавяне няма значение
- Начинът на работа със `SortedDictionary<K, V>` е аналогичен на обикновения речник



# 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

