

# Списъци

## Обработка на поредици с променлива дължина

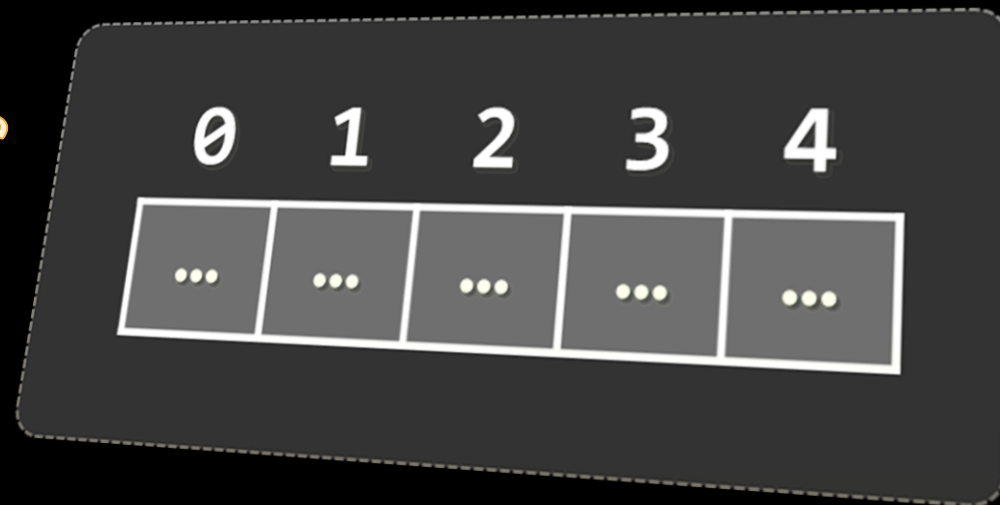
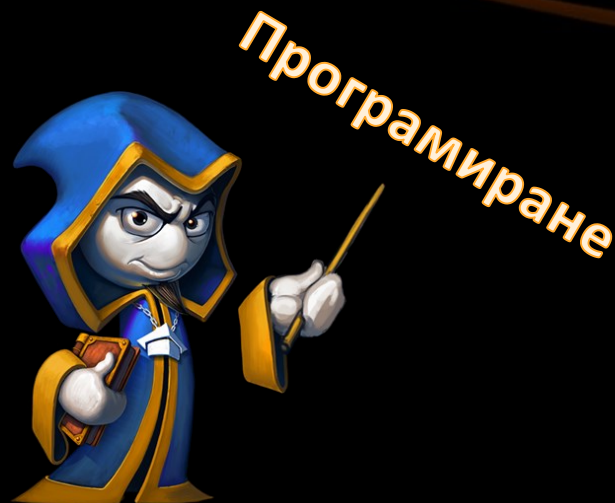


Учителски екип

Обучение за ИТ кариера

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

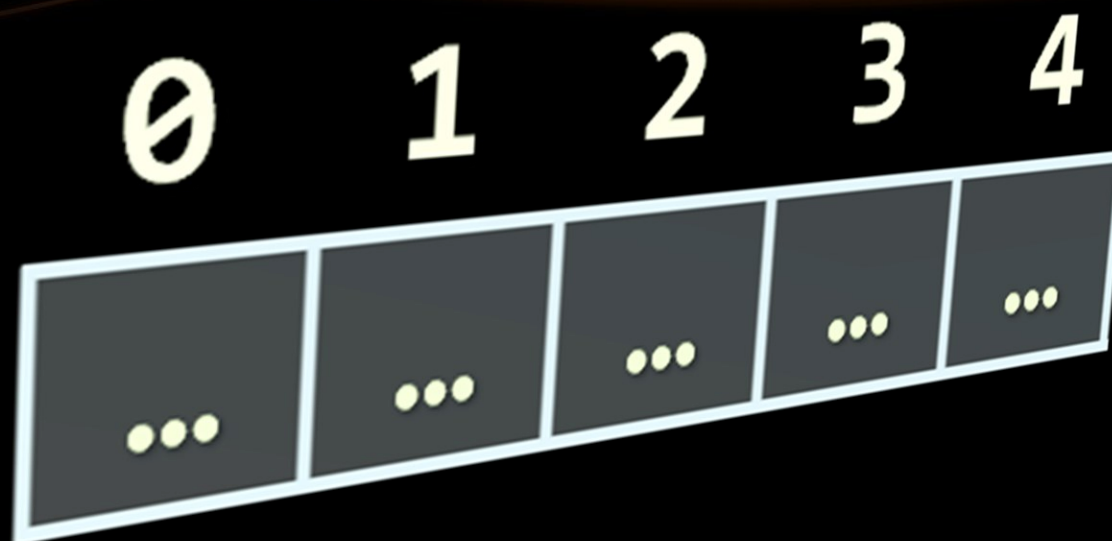
<https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Fundamentals>



# Съдържание

1. Що е то списък?
2. Операции върху списъци
3. Вход/Изход на списък





# Списъци

Масиви с променлива дължина

# List<T> – Списък

- **List<T>** - списък на елементи от какъвто е тип **T**

```
var names = new List<string>(); // създава списък от низове
```

```
names.Add("Peter");
```

```
names.Add("Maria");
```

```
names.Add("George");
```

```
foreach (var name in names)
```

```
    Console.WriteLine(name);
```

```
names.Remove("Maria");
```

```
Console.WriteLine(  
    String.Join(", ", names));
```

```
var nums = new List<int> {  
    10, 20, 30, 40, 50, 60};
```

```
nums.RemoveAt(2);
```

```
nums.Add(100);
```

```
nums.Insert(0, -100);
```

```
Console.WriteLine(  
    String.Join(", ", nums));
```



# Структурата от данни List<T>

- List<T> съдържа списък от елементи (като масив, но разширяващ се)
- Позволява добавяне / вмъкване / премахване / търсене на елементи:
  - Add(element) – добавя елемент към List<T>
  - Count – връща броя на елементите в List<T>
  - Remove(element) – премахва първото срещане на елемент (връща true / false)
  - RemoveAt(index) – премахва елемент по неговия индекс
  - Insert(index, element) – вмъква елемент на зададената позиция
  - Contains(element) – определя дали елемента се съдържа в списъка
  - Sort() – сортира във възходящ ред
  - Reverse() – обръща списъка наобратно

# Add() – Залепя елемента за края

100

List<int>

Брой:

0

# Remove() – Изтрива елемент

10

List<int>

2

10

5

Брой:

3

# Insert() – Добавя елемент на позиция

-5

List<int>

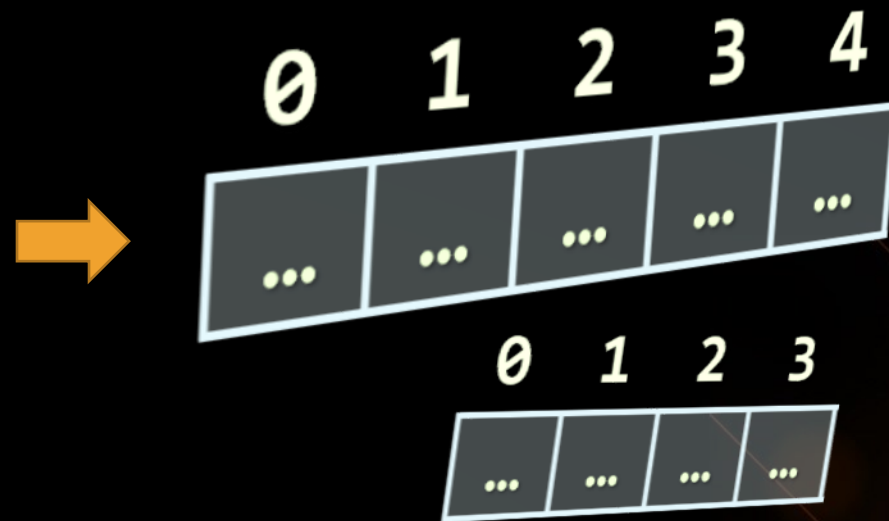
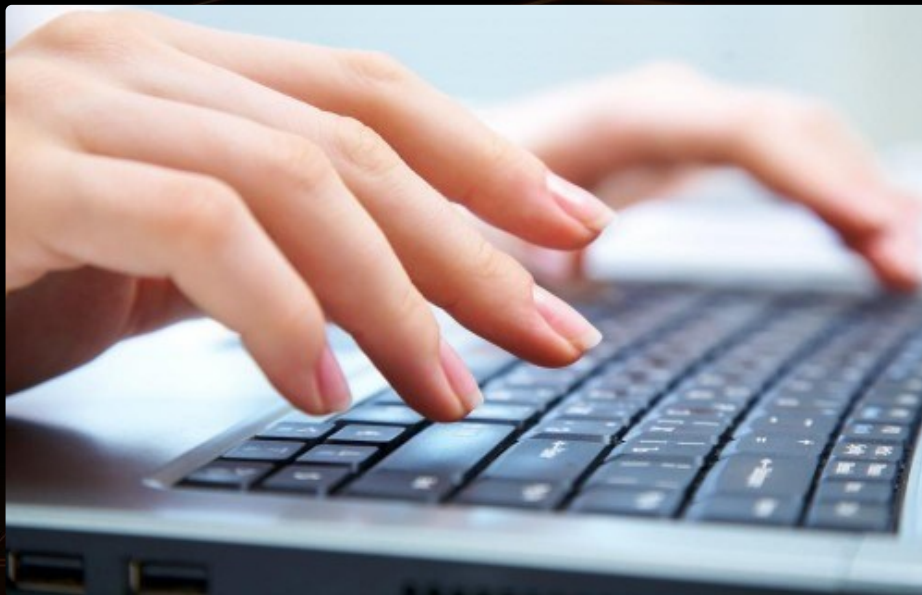
2

5

Брой:

3





**Въвеждане на списъци от конзолата**  
Използваме `for` или `String.Split()`

# Въвеждане на списъци от конзолата

- Първо, въвеждаме от конзолата **дължината** на списъка:

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

- После, създаваме списък с размер **n** и въвеждаме **елементи**:

```
List<int> list = new List<int>();  
  
for (int i = 0; i < n; i++)  
{  
    list.Add(int.Parse(Console.ReadLine()));  
}
```

Тествайте в Judge: <https://judge.softuni.bg/Contests/2656>

# Въвеждане на данни от един ред

- Стойностите могат да се въвеждат от **един ред с интервали**:

2 8 30 25 40 72 -2 44 56

```
string values = Console.ReadLine();  
List<string> items = values.Split(' ').ToList();  
List<int> nums = new List<int>();  
for (int i = 0; i < items.Count; i++)  
    nums.Add(int.Parse(items[i]));
```

`string.Split(' ')`  
отделя **НИЗА** с интервали  
и прави колекция

Превръща  
колекцията в **List**

```
var items = Console.ReadLine().Split(' ')  
    .Select(int.Parse).ToList();
```

Всичко  
наведнъж

# Изпечатване на списъци на конзолата

- Изпчатване на списъка чрез **for**-цикъл:

```
List<string> list = new List<string>() {  
    "one", "two", "three", "four", "five", "six"};  
for (int index = 0; index < list.Count; index++)  
    Console.WriteLine("arr[{0}] = {1}", index, list[index]);
```

- Изпечатване на списъка чрез **String.Join(...)**:

```
List<string> list = new List<string>() {  
    "one", "two", "three", "four", "five", "six"};  
Console.WriteLine(String.Join("; ", list));
```



# Какво научихме този час?

- **Списъкът** съдържа поредица от елементи (като масив, но с променлива дължина)
  - Може да добавяме / трием / вмъкваме елементи по време на работата на програмата

- Създаване на списък:

```
List<int> numbers = new List<int>();  
var nums = new List<int>() { 1, 2, 3 };
```

- Достъп до елементите: `numbers[5] = 10;`

- Отпечатване на списък: `Console.WriteLine(string.Join(" ", list));`

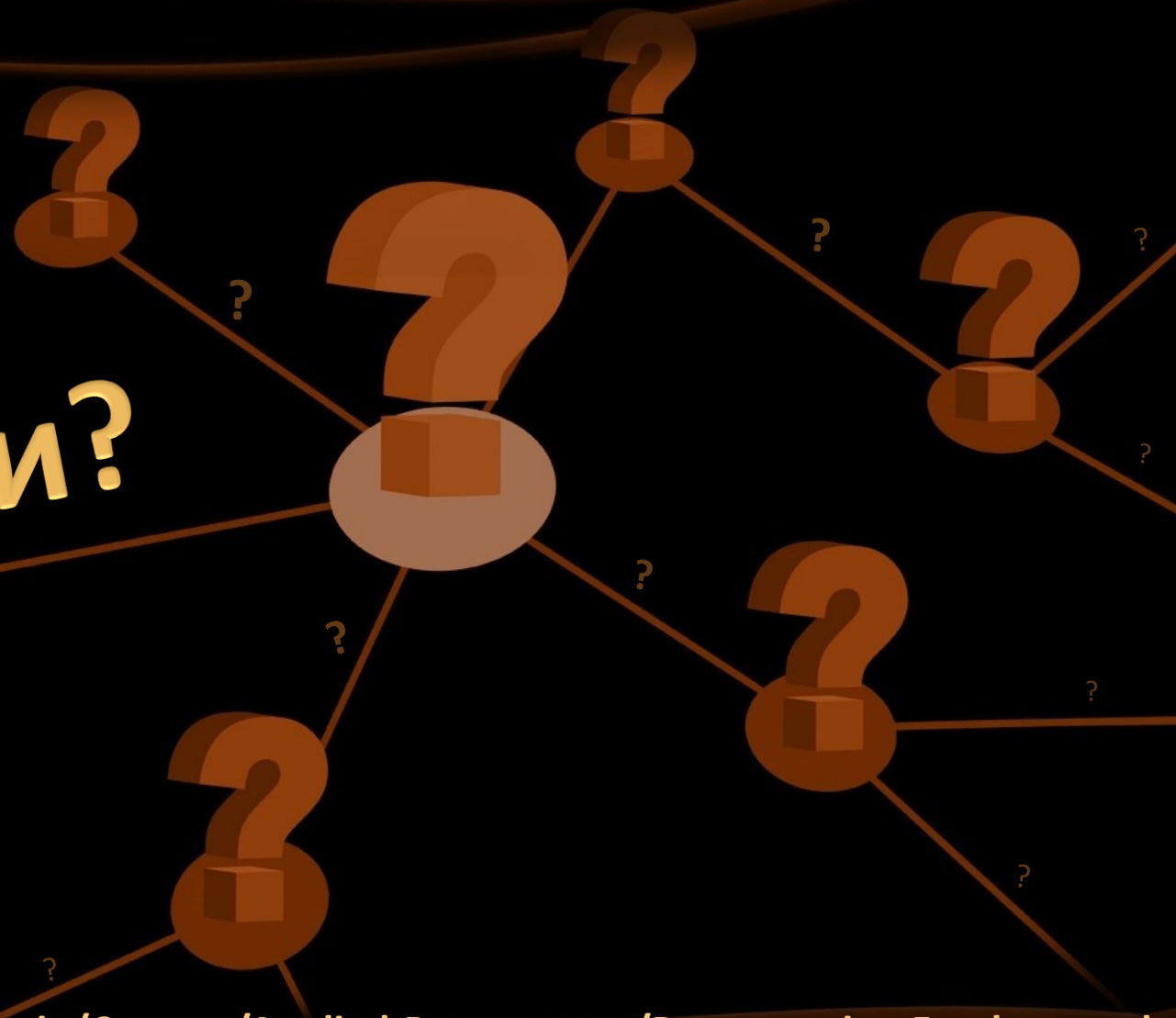




# Списъци



## 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

