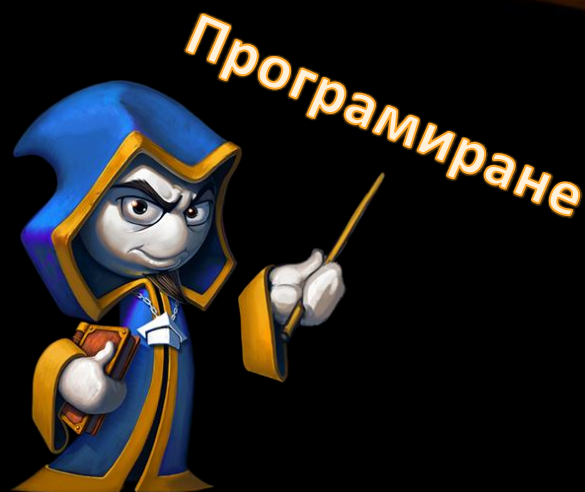


# Многомерни масиви

## Масиви с повече размерности



```
int[,] intMatrix =  
{  
    {2, 8, 3, 5},  
    {7, 9, 0, 3},  
};
```



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. Какво е многомерен масив?
2. Двумерни масиви – матрици
3. Многомерни масиви
  - размерности
  - индексирание



# Какво е многомерен масив?

- **Двумерен масив** = таблица (или матрица)
- Всеки елемент се определя чрез:
  - номер на реда
  - номер на колоната
- **Многомерен масив** = масив от масиви:
  - Двумерен масив = масив от едномерни масиви
  - Тримерен масив = масив от двумерни масиви
  - Четиримерен масив = масив от тримерни масиви
- **Основните правила** от едномерни масиви **важат и за многомерни**

```
int[,] intMatrix =  
{  
    {2, 8, 3, 5},  
    {7, 9, 0, 3},  
};
```

# Двумерен масив – таблица или матрица

- Има **rows** x **columns** на брой елементи, където:
  - **rows** е броят на редовете, а
  - **columns** – броят на колоните
- Размера на масива е **постоянен**
- Елементите са от **един и същ** тип
- Елементите са номерирани с два индекса:
  - **Ред** – от 0 до **rows-1**
  - **Колона** – от 0 до **columns-1**

	0	1	2	3
0	2	8	3	5
1	7	9	0	3

# Деклариране на многомерен масив

- Едномерен масив от цели числа декларирахме чрез:

```
int[] simpleArray;
```

- Двумерен масив декларираме така:

```
int[,] twoDimensionalArray;
```

- Аналогично е и за тримерен масив:

```
int[, ,] threeDimensionalArray;
```

- Няма теоретично ограничение за броя на размерностите



# Деклариране и заделяне

- Отбелязването на променливата като многомерен масив не заделя памет за неговите елементи. За целта използваме **new**:

```
int[,] intMatrix = new int[3, 4];
```

Двумерен масив от цели числа с 3 реда и 4 колони на ред

```
float[, ,] floatCube = new float[5, 5, 5];
```

Тримерен масив от 5 елемента, като всеки е двумерен масив с 5 реда и 5 колони

# Инициализация на двумерен масив

- Както при едномерените масиви, можем да зададем стойности на многомерния масив при деклариране:

```
// декларираме двумерен масив 2 x 4 (2 реда, 4 колони)
int[,] intMatrix =
{
    {2, 8, 3, 5},
    {7, 9, 0, 3},
};
```

	0	1	2	3
0	2	8	3	5
1	7	9	0	3

# Достъп до елементите на многомерен масив

- Както при едномерните масиви, така и при многомерните **всички индекси започват от 0**. Разликата е, че тук индексите са повече от 1.
- Ето как достъпваме елементите на масива от предния слайд:

`intMatrix[0, 0]`

`intMatrix[0, 1]`

`intMatrix[0, 2]`

`intMatrix[0, 3]`

`intMatrix[1, 0]`

`intMatrix[1, 1]`

`intMatrix[1, 2]`

`intMatrix[1, 3]`

- Индексите се отделят със запетаи!

	0	1	2	3
0	2	8	3	5
1	7	9	0	3



# Дължина на многомерен масив

- Всяка размерност на многомерен масив може да има различна дължина спрямо останалите.
- Всяка размерност се номерира по сходен начин с индекси.
- За да разберем колко реда има двумерния масив:

```
intMatrix.GetLength(0);
```

- А за да разберем колко колони има:

```
intMatrix.GetLength(1);
```

# Пример: Отпечатване на матрица

```
int[,] intMatrix =  
{  
    {2, 8, 3, 5},  
    {7, 9, 0, 3},  
};
```

```
for(int row = 0; row < intMatrix.GetLength(0); row++)  
{  
    for(int col = 0; col < intMatrix.GetLength(1); col++)  
    {  
        Console.Write(intMatrix[row, col]+" ");  
    }  
    Console.WriteLine();  
}
```

Брой редове

Брой  
колони

Отпечатване  
на елемента

Прехвърляме  
се на нов ред

# Пример: Вход / изход на матрица

```
int rows = int.Parse(Console.ReadLine()); //въвеждаме брой редове
int cols = int.Parse(Console.ReadLine()); //въвеждаме брой колони
int[,] matrix = new int[rows, cols]; //заделяме памет за масива

for(int row = 0; row < rows; row++)
{
    for(int col = 0; col < cols; col++)
    {
        Console.Write("matrix[{0},{1}] = ", row, col);
        matrix[row, col] = int.Parse(Console.ReadLine());
    }
}
//TODO: отпечатваме елементите на масива като в предния пример
```

# Задача: Средноаритметично по редове

- Напишете програма, която обработва двумерен масив.
  - Изведете го като в края на всеки ред добавите средноаритметичното от реда.
  - Всеки елемент да заема 8 позиции

Тествайте в Judge: <https://judge.softuni.bg/Contests/2667>

# Пример: Средноаритметично по редове

2  
4  
1  
2  
3  
5  
8  
6  
9  
4



1	2	3	5	2.75
8	6	9	4	6.75

Тествайте в Judge: <https://judge.softuni.bg/Contests/2667>



# Решение: Средноаритметично по редове

```
// TODO: Въведете матрицата
for (int row = 0; row < rows; row++) {
    double avg = 0;
    for (int col = 0; col < cols; col++) {
        Console.Write("{0, 8}", matrix[row, col]);
        avg += matrix[row, col];
    }
    avg = avg / cols;
    Console.WriteLine("{0, 8}", avg);
}
```

В началото на всеки ред, зануляваме

Извеждаме в 8 позиции

Изчисляваме средноаритметично и го извеждаме

Тествайте в Judge: <https://judge.softuni.bg/Contests/2667>

# Какво научихме този час?

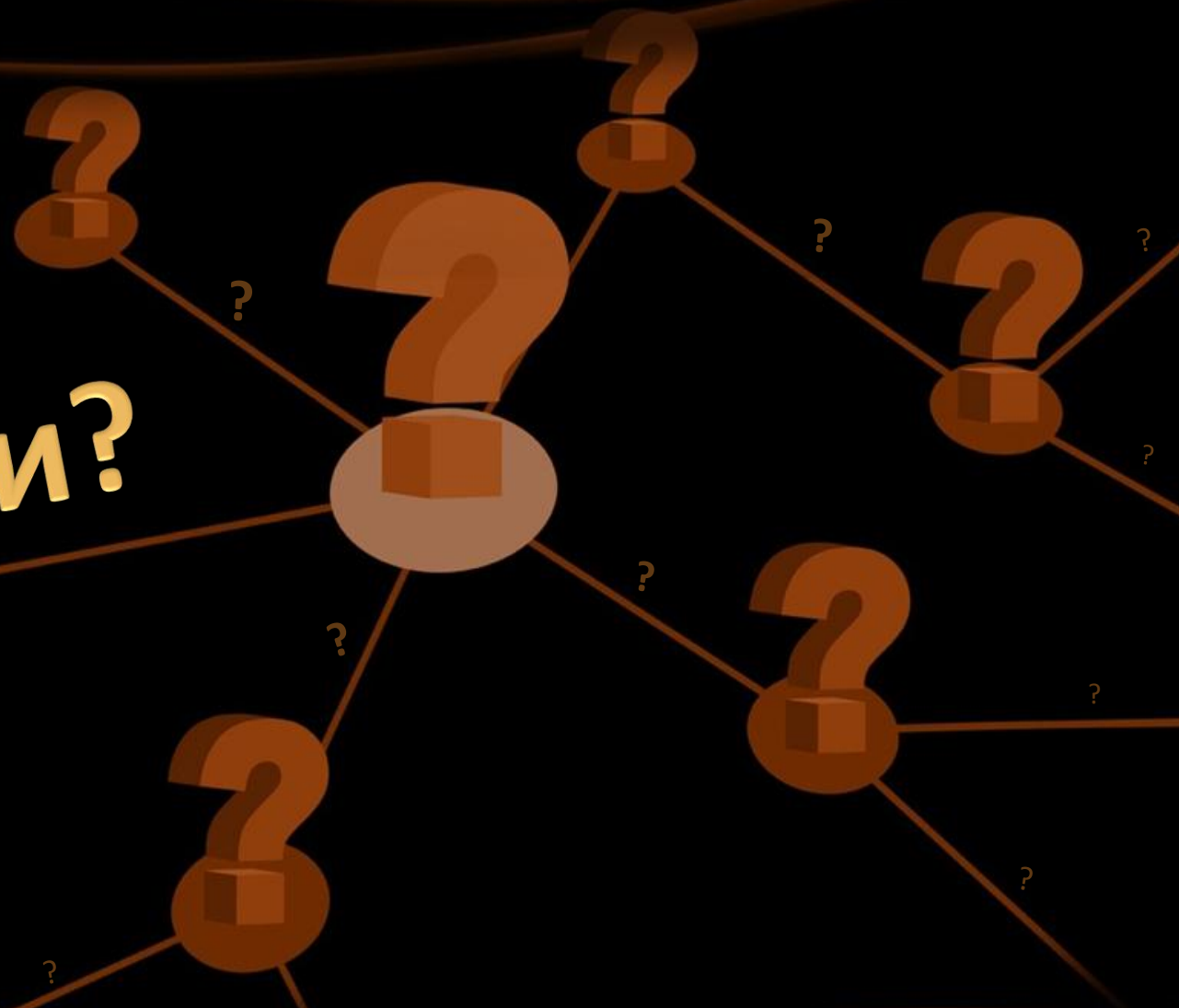
- Многомерните масиви са масиви, чиито елементи са също (многомерни) масиви
- Многомерните масиви разполагат с повече от една размерност и с толкова индекси, колкото размерности имат
- Памет за многомерните масиви се заделя чрез `new` като посочим за всяка размерност колко елемента трябва да има



# Многомерни масиви



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

