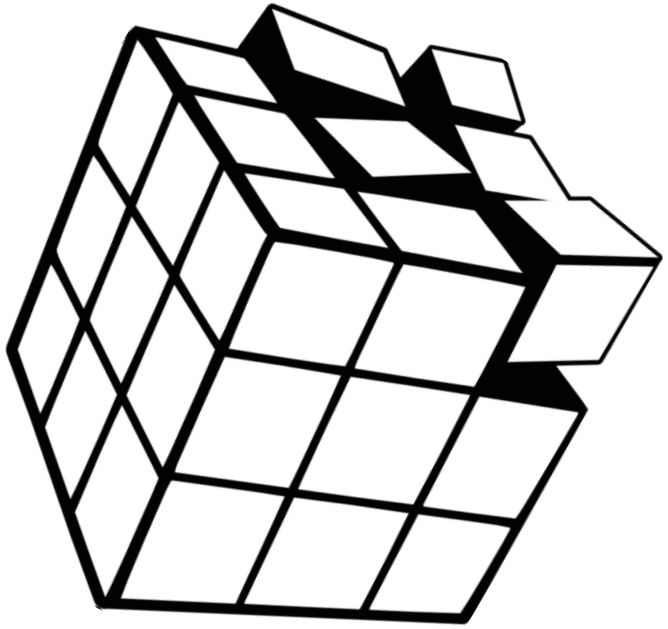
Многомерни масиви

Обработка на матрици и на назъбени масиви



SoftUni Foundation

Проект "Отворено учебно съдържание по програмиране и ИТ", СофтУни Фондация

https://github.com/BG-IT-Edu

Курс "Структури от данни и алгоритми"

Софтуерни и хардуерни науки

Съдържание

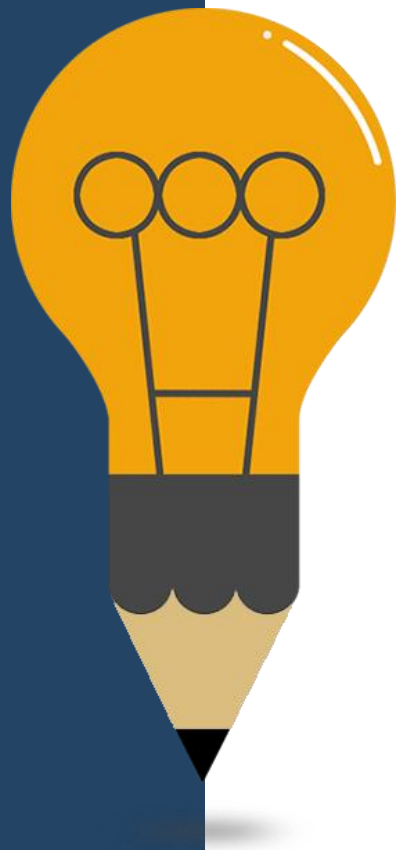
SoftUni Foundation

1. **Многомерни масиви**
   - Създаване
   - Достъп до елементи
   - Четене и отпечатване
2. **Назъбени масиви** (масив от масиви)
   - Създаване
   - Достъп до елементи
   - Четене и отпечатване

Многомерни масиви

Определение и използване

Какво е многомерен масив?



- Масивът е систематично подреждане на **подобни елементи**
- **Многомерните масиви** имат повече от едно измерение
  - Най-често използваните многомерни масиви са с **две измерения**

Ред	Колона				
	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

Индекс на реда

Индекс на колоната

Създаване на многомерен масив (1)

- Използваме ключовата дума **new**
- Трябва да се определи **размера** на всяко измерение

```
int[,] intMatrix = new int[3, 4];
float[,] floatMatrix = new float[8, 2];
string[,,] stringCube = new string[5, 5, 5];
```

- Този синтаксис се използва **само в C#**

Създаване на многомерен масив (2)

SoftUni Foundation

- Създаване със стойности:

```
int[,] matrix = {
    {1, 2, 3, 4}, // ред 0 стойности
    {5, 6, 7, 8} // ред 1 стойности
};
```

- Многомерните масиви представляват **редове със стойности**
- **Редовете** са **първото** измерение, а **колоните** са **второто**

Достъп до елементи

- **Взимане** на стойността на елемента:

```
int[,] array = {{1, 2}, {3, 4}}
int element11 = array[1, 1]; // element11 = 4
```

- **Задаване** на стойност на елемента:

```
int[,] array = new int[3, 4];
for (int row = 0; row < array.GetLength(0); row++)
  for (int col = 0; col < array.GetLength(1); col++)
    array[row, col] = row + col;
```

Връща дължината на измерението

Отпечатване на матрица – Пример (1)

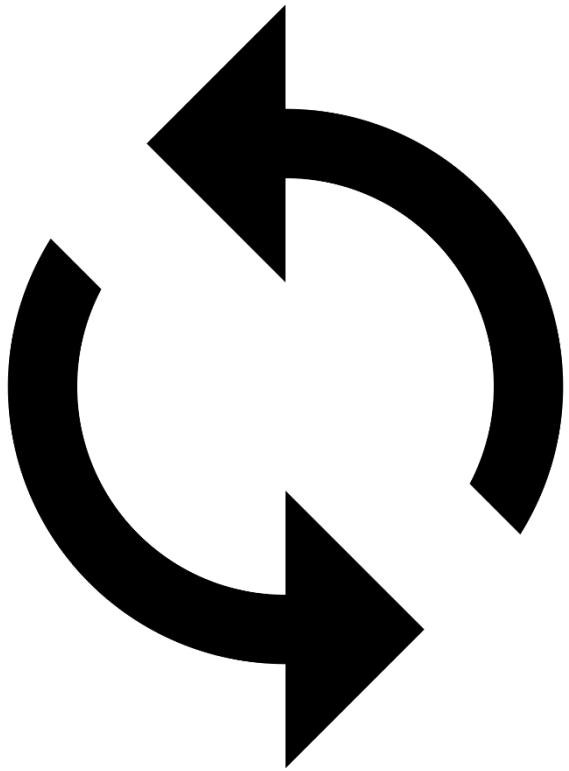
SoftUni Foundation

```
int[,] matrix =
                  {  { 5, 2, 3, 1 },
                     { 1, 9, 2, 4 },
                     { 9, 8, 6, 11 }  };
for (int row = 0; row < matrix.GetLength(0); row++)
{
  for (int col = 0; col < matrix.GetLength(1); col++)
  {
    Console.Write("{0} ", matrix[row, col]);
  }

  Console.WriteLine();
}
```

Отпечатване на матрица – Пример (2)

- Чрез **foreach-цикъл** минаваме през всички елементи на матрицата

```
int[,] matrix = {
  { 5, 2, 3, 1 },
  { 1, 9, 2, 4 },
  { 9, 8, 6, 9 }
};

foreach (int element in matrix)
{
  Console.WriteLine(element);
}
```



Задача: Сума на елементите в матрица

- Прочетете матрицата от конзолата
- Отпечатайте броя на редовете
- Отпечатайте броя на колоните
- Отпечатайте **сумата на всички елементи** в матрицата

```
3, 6
7, 1, 3, 3, 2, 1
1, 3, 9, 8, 5, 6
4, 6, 7, 9, 1, 0
```

```
3
6
76
```

```
3, 4
1, 2, 3, 1
1, 2, 2, 4
2, 2, 2, 2
```

```
3
4
24
```

Решение: Сума на елементите в матрица (1)

```
int[] sizes = Console.ReadLine().Split(", ")
    .Select(int.Parse).ToArray();

int[,] matrix = new int[sizes[0], sizes[1]];

for (int row = 0; row < matrix.GetLength(0); row++)
{
  int[] colElements = Console.ReadLine()
        .Split(", ")
        .Select(int.Parse)
        .ToArray();
  for (int col = 0; col < matrix.GetLength(1); col++)
    matrix[row, col] = colElements[col];
}
```

Взимаме дължината на нулевото измерение (редове)

Взимаме дължината на първото измрение (колони)

Решение: Сума на елементите в матрица (2)

SoftUni Foundation

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

Тествайте решението в Judge: https://judge.softuni.org/Contests/Practice/Index/4156#2

Задача: Сума на колоните на матрица

- Прочетете размерите на матрицата
- Прочетете матрицата
- Отпечатайте **сумата на числата** във всяка колона

```
3, 6
7 1 3 3 2 1
1 3 9 8 5 6
4 6 7 9 1 0
```

```
12
10
19
20
8
7
```

```
3, 3
1 2 3
4 5 6
7 8 9
```

```
12
15
18
```

Решение: Сума на колоните на матрица (1)

SoftUni Foundation

```
var sizes = Console.ReadLine().Split(", ")
    .Select(int.Parse).ToArray();
int[,] matrix = new int[sizes[0], sizes[1]];
for (int r = 0; r < matrix.GetLength(0); r++)
{
  var col = Console.ReadLine().Split()
      .Select(int.Parse).ToArray();
  for (int c = 0; c < matrix.GetLength(1); c++)
  {
    matrix[r, c] = col[c];
  }
}
```

Решение: Сума на колоните на матрица (2)

SoftUni Foundation

```
for (int c = 0; c < matrix.GetLength(1); c++)
{
  int sum = 0;
  for (int r = 0; r < matrix.GetLength(0); r++)
  {
    sum += matrix[r, c];
  }
  Console.WriteLine(sum);
}
```

Тествайте решението си в Judge: https://judge.softuni.org/Contests/Practice/Index/4156#3

Задача: Квадрат с най-голяма сума

- Намерете **квадрата** с най-голяма сума в матрица **с размери 2x2**
  - Прочетете матрицата от конзолата
  - Намерете **най-голямата сума** с размери 2x2
  - Отпечатайте квадрата и сумата му

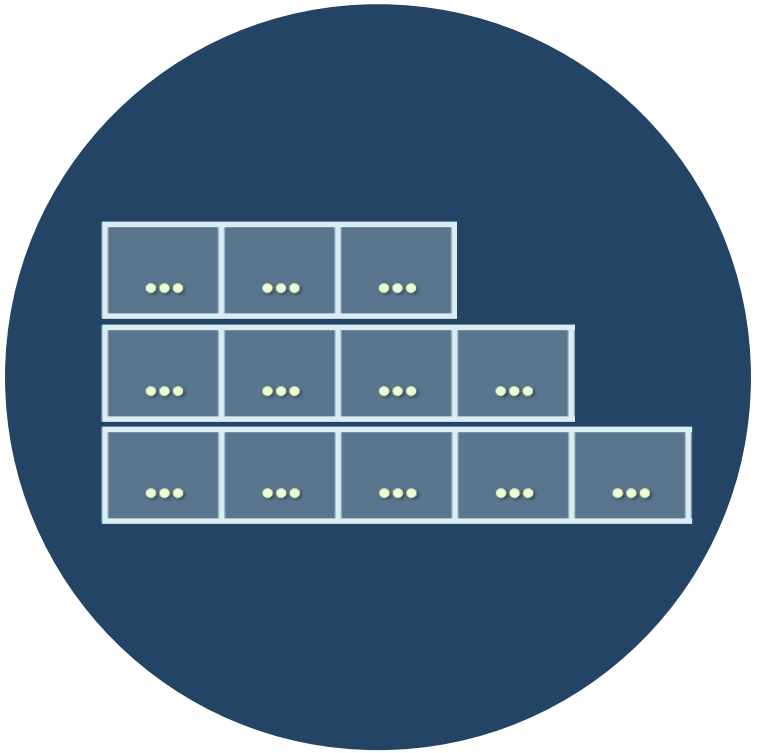
```
int[,] matrix = {
  {7, 1, 3, 3, 2, 1},
  {1, 3, 9, 8, 5, 6},
  {4, 6, 7, 9, 1, 0}
};
```

```
9 8
7 9
33
```

Решение: Квадрат с най-голяма сума

```
// TODO: Прочетете входа от конзолата
for (int row = 0; row < matrix.GetLength(0) - 1; row++) {
  for (int col = 0; col < matrix.GetLength(1) - 1; col++) {
    var newSquareSum = matrix[row, col] +
                       matrix[row + 1, col] +
                       matrix[row, col + 1] +
                       matrix[row + 1, col + 1];
    // TODO: Проверете дали сумата е по-голама
  }
}
// TODO: Отпечатайте квадрата и сумата му
```

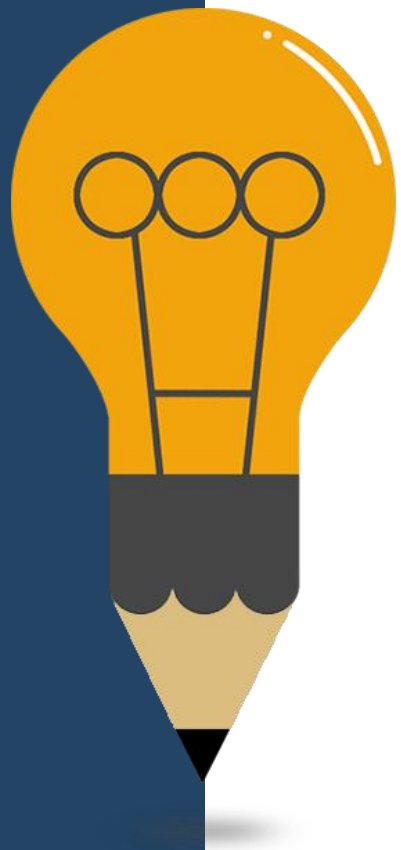
Тествайте решението си в Judge: https://judge.softuni.org/Contests/Practice/Index/4156#4



Назъбени масиви

Определение и използване

Какво е назъбен масив?



- **Назъбен масив** == многомерен масив, но всяко измерение има **различна дължина**
  - Назъбеният масив е **масив от масиви**
  - Всеки масив има **различна дължина**

```
int[][] jagged = new int[3][];
jagged[0] = new int[3];
jagged[1] = new int[2];
```

  - **Достъп** до елемент

```
int element = jagged[0][0];
```

Индекс на **колоната**

Индекс на **редицата**

Запълване на назъбен масив

SoftUni Foundation

```
int[][] jagged = new int[5][];

for (int row = 0; row < jagged.Length; row++)
{
  string[] inputNumbers = Console.ReadLine().Split(' ');
  jagged[row] = new int[inputNumbers.Length];

  for (int col = 0; col < jagged[row].Lenght; col++)
  {
    jagged[row][col] = int.Parse(inputNumbers[col]);
  }
}
```

Отпечатване на назъбен масив – Пример

SoftUni Foundation

- For-цикъл

```
int[][] matrix = ReadMatrix();
for (int row = 0; row < matrix.Length; row++)
  for (int col = 0; col < matrix[row].Length; col++)
    Console.Write("{0} ", matrix[row][col]);
Console.WriteLine();
```

- Foreach-цикъл

```
int[][] matrix = ReadMatrix();
foreach (int[] row in matrix)
{
  Console.WriteLine(string.Join(" ", row));
}
```

Задача: Модификация на назъбен масив

- На първия ред получавате броя на редовете: **n**
- На следващите **n** редове получавате елементите за всеки ред
- Докато получите "**END**", четете командите:
  - Add **{ред} {колона} {стойност}**
  - Subtract **{ред} {колона} {стойност}**
- Ако координатите са **невалидни**, отпечатайте: **"Invalid coordinates"**
- Когато получите "**END**", отпечатайте назъбения масив

```
3
1 2 3
4 5 6
7 8 9
Add 0 0 5
Subtract 1 1 2
END
```

Решение: Модификация на назъбен масив (1)

```
int rowSize = int.Parse(Console.ReadLine());
int[][] matrix = new int[rowSize][];

for (int r = 0; r < rowSize; r++)
{
  int[] col = Console.ReadLine()
                     .Split()
                     .Select(int.Parse)
                     .ToArray();
  matrix[r] = col;
}
// Продължаваме на следващия слайд
```

Решение: Модификация на назъбен масив (2)

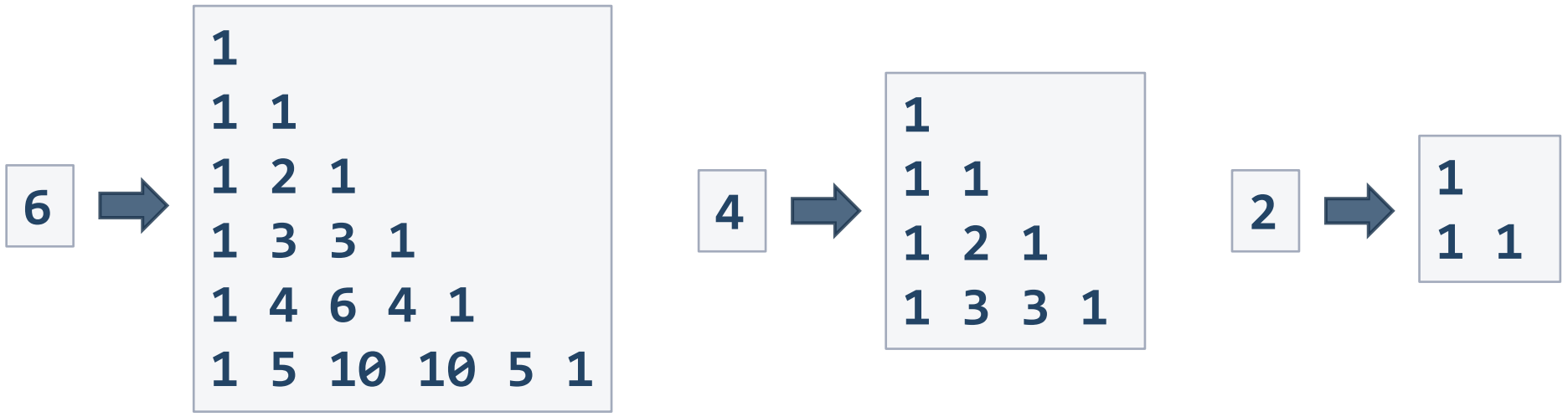
```
string line;
while ((line = Console.ReadLine()) != "END") {
  string[] tokens = line.Split();
  string command = tokens[0];
  int row = int.Parse(tokens[1]);
  int col = int.Parse(tokens[2]);
  int value = int.Parse(tokens[3]);
  if (row < 0 || row >= matrix.Length || … )
    { Console.WriteLine("Invalid coordinates"); }
  else
    {// TODO: Напишете командите}
}
// TODO: Отпечатайте матрицата
```

Проверяваме за колоната

Тествайте решението в Judge: https://judge.softuni.org/Contests/Practice/Index/4156#5

Задача: Триъгълника на Паскал

- Напишете програма, която отпечатва **триъгълника на Паскал**



Решение: Триъгълника на Паскал (1)

SoftUni Foundation

```
int height = int.Parse(Console.ReadLine());
long[][] triangle = new long[height][];
int currentWidth = 1;
for (long row = 0; row < height; row++)
{
  triangle[row] = new long[currentWidth];
  long[] currentRow = triangle[row];
  currentRow[0] = 1;
  currentRow[currentRow.Length - 1] = 1;
  currentWidth++;
  // TODO: Запълнете елементите на всеки ред
}
```

Решение: Триъгълника на Паскал (2)

```
if (currentRow.Length > 2)
{
  for (int i = 1; i < currentRow.Length - 1; i++)
  {
    long[] previousRow = triangle[row - 1];
    long prevoiousRowSum = previousRow[i] + previousRow[i - 1];
    currentRow[i] = prevoiousRowSum;
  }
}
// TODO: Отпечатайте триъгълника
foreach (long[] row in triangle)
  Console.WriteLine(string.Join(" ", row));
```

Тествайте решението си в Judge: https://judge.softuni.org/Contests/Practice/Index/4156#6

Какво научихме днес?

SoftUni Foundation

- **Многомерен масив**
  - Има **повече от едно** измерение
  - Масив от второ измерение е като таблица от **редици** и **колони**
- **Назъбен масив**
  - Масив от масиви
  - Всеки **елемент сам по себе** си е масив