

10a ТЕМА 9 – МАСИВИ

Структура от данни, съдържаща елементи от един и същи тип с фиксиран размер.

Индексацията: Винаги започва от 0.

Едномерен масив; Деклариране и инициализация:

```
int[] numbers = new int[5]; // Масив с 5 елемента (по подразбиране са 0)
string[] fruits = { "Apple", "Banana", "Cherry" }; // Директна инициализация
```

Достъп до елемент: `fruits[0]` връща " **Apple**".

```
int[] grades = { 4, 5, 6, 3, 6 };
Console.WriteLine("Първа оценка: " + grades[0]);
// Промяна на стойност
grades[3] = 4;
```

Многомерен масив (Двумерен)

Представен, като таблица с редове и колони.

Деклариране:

```
int[,] matrix = new int[3, 4]; // 3 реда и 4 колони
int[,] table = { { 1, 2 }, { 3, 4 } };
```

Достъп: `table[0, 1]` (ред 0, колона 1) връща **2**.

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

Тримерен и n-мерен масив

Представени като "куб" или "книга" с няколко страници (таблици).

Пример за 3D масив:

```
int[, ,] cube = new int[2, 3, 4]; // (дълбочина, ред, колона)
```

```
int[, ,] space = new int[2, 2, 2] {  
    { {1, 2}, {3, 4} },  
    { {5, 6}, {7, 8} }  
};  
// Достъп до числото 6  
Console.WriteLine(space[1, 0, 1]);
```

Назъбен масив (Jagged Array) - Бонус концепция

Масив от масиви, където всеки ред може да има различна дължина.

```
int[][] jagged = new int[2][];  
jagged[0] = new int[] { 1, 2, 3 };  
jagged[1] = new int[] { 4, 5 };
```

Практическа задача: Среден успех на класа

Условие: Да се създаде програма, която съхранява оценките на ученици по предметите им. Програмата трябва да изчисли и изведе средния успех на целия клас.

Допълнение: да изчислява и средния успех на всеки отделен ученик и вариант освен **Random** вариант да има възможност за въвеждане на оценките за всеки ученик и предмет.(примерно 2 ученика и 3 предмета или 20 ученика и 10 предмета