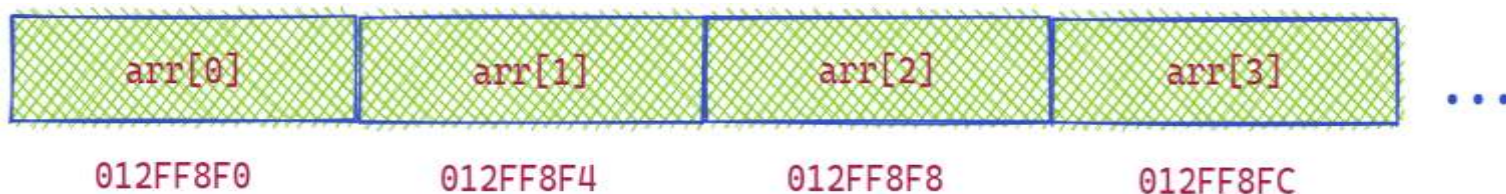


# Семинар по „Увод в програмирането“

## Масиви

### 1. Какво представляват масивите?

Масивът е структура от данни, която представлява крайна редица от елементи от един тип. Масивите ни позволяват пряк достъп до всеки един елемент. Достъпът става чрез идентификатор. Всеки един елемент се намира на определено място в паметта и си има свой адрес. За адреси ще си говорим в следващия семинар.



### 2. Деклариране

Декларираме масив по следния начин:

**<data type> <name of array>[<capacity>]**

Пример:

```
int arr[4];
```

Декларираме масив от тип **int** с 4 елемента.

Друг начин за деклариране на масив е:

```
int arr[4] = {1, 2, 3, 4};
```

Отново декларираме масив с 4 елемента, но този път и задаваме стойности на всяка една клетка. Не е задължително да се дава стойност на всяка клетка. Пример:

```
int arr[4] = { 1, 2, 3 };

std::cout << arr[0] << "\n";
std::cout << arr[1] << "\n";
std::cout << arr[2] << "\n";
std::cout << arr[3] << "\n";
```

Съответно резултата ще е следния:

```
1
2
3
0
```

### 3. Индексиране

За да достъпваме елементите в масива ни е нужен индекс. Индексирането на масива започва от 0 и завършва с  $n-1$ , като  $n$  е капацитета на масива.



В случай, че се опитаме да достъпим елемент на масива, извън неговия капацитет, ще получим грешка.

```
int arr[3] = { 1, 2, 3 };
```

```
arr[0] = 1;
arr[1] = 2;
arr[2] = 3;
arr[3] = 4;
```

(local variable) int arr[3]

[Search Online](#)

C6201: Index '3' is out of valid index range '0' to '2' for possibly stack allocated buffer 'arr'.

C6386: Buffer overrun while writing to 'arr': the writable size is '12' bytes, but '16' bytes might be written.

Най-често компилатора ни подсказва, че няма как да осъществим достъп до тази памет, тъй като тя не е заделена.

## 4. Въвеждане и извеждане на стойности на масив

### 1) Въвеждане на стойности в масив

Пример №1

```
int arr[4];  
  
for (int i = 0; i < 4; i++)  
{  
    std::cin >> arr[i];  
}
```

Пример №2

```
int arr[4];  
  
arr[0] = 1;  
arr[1] = 2;  
arr[2] = 3;  
arr[3] = 4;
```

Пример №3

```
int arr[4] = { 1, 2, 3, 4 };
```

! Грешка може да се получи, когато искаме да въведем повече стойности, от колкото разполага с елементи масива.

### 2) Извеждане на масива

```
int arr[4] = { 1, 2, 3, 4 };  
  
for (int i = 0; i < 4; i++)  
{  
    std::cout << arr[i];  
}
```

## 5. Многомерни масиви

### 1) Какво представляват?

Многомерните масиви представляват масиви от масиви, които също са от един и същ тип и също можем да достъпваме елементите им чрез индекси. За целта на курса ще използваме двумерни масиви или още познати като **матрици**.

	Column 1	Column 2	Column 3	Column 4	...
Row 1	<code>arr[0][0]</code>	<code>arr[0][1]</code>	<code>arr[0][2]</code>	<code>arr[0][3]</code>	
Row 2	<code>arr[1][0]</code>	<code>arr[1][1]</code>	<code>arr[1][2]</code>	<code>arr[1][3]</code>	
Row 3	<code>arr[2][0]</code>	<code>arr[2][1]</code>	<code>arr[2][2]</code>	<code>arr[2][3]</code>	
Row 4	<code>arr[3][0]</code>	<code>arr[3][1]</code>	<code>arr[3][2]</code>	<code>arr[3][3]</code>	

### 2) Деклариране

Декларираме матрица като използваме два пъти квадратни скоби като първите представляват колко реда ще има матрицата, а вторите колко колони. Пример:

```
int matrix[4][4];
```

```
int matrix[2][3] =  
{  
  {1, 2, 3},  
  {5, 6, 7}  
}
```

### 3) Индексиране

	Column 1	Column 2	Column 3	Column 4	
Row 1	0	1	2	3	...
Row 2	4	5	6	7	
Row 3	8	9	10	11	
Row 4					

`matrix[1][2]`

### 4) Задаване на стойности

Приемр №1:

```
int matrix[3][3];

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        std::cin >> matrix[i][j];
    }
}
```

Пример №2:

```
int matrix[3][3];

matrix[0][0] = 1;
matrix[0][1] = 2;
matrix[0][2] = 3;

matrix[1][0] = 4;
matrix[1][1] = 5;
matrix[1][2] = 6;

matrix[2][0] = 7;
matrix[2][1] = 8;
matrix[2][2] = 9;
```

5) Принтиране на матрица на конзолата

```
int matrix[3][3];

matrix[0][0] = 1;
matrix[0][1] = 2;
matrix[0][2] = 3;

matrix[1][0] = 4;
matrix[1][1] = 5;
matrix[1][2] = 6;

matrix[2][0] = 7;
matrix[2][1] = 8;
matrix[2][2] = 9;

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        std::cout << matrix[i][j] << " ";
    }
    std::cout << std::endl;
}
```