

**ЕДНОМЕРЕН МАСИВ.**

# 1. МАСИВИ.

Масивът е област от паметта (структура), съставена от еднотипни полета с еднакъв размер, в които могат да се съхраняват стойности от един и същи тип. Масивът има име, което е общо за всички полета и дължина – броят на полетата. Полето на масива се нарича елемент на масива. Елементите на масива се различават един от друг по своя индекс. Номерирането на елементите започва от нула. В C# името на елемент на масив се образува като след името на масива се постави индекса му ограден в квадратни (средни) скоби. Например: `a[1]`, `a[2]` и т.н.



## 2. ДЕКЛАРИРАНЕ НА МАСИВ.

Както променливите така и масивите трябва да се декларират.

Синтакси на оператора за деклариране на масив е:


**<тип>[ ] <име на масив>**

Например: `int [ ] a; double [ ] points; string [ ] names;`

В един оператор могат да се декларират няколко масива.

Например: `int [ ] a, b, c;`

За разлика от декларирането на променливи, декларацията на масив не означава автоматично заделяне на памет за масива по време на изпълнение на програмата, както е с обикновенните променливи. Никъде в оператора за деклариране на масив не се указва размера му.



Затова преди да се използва един масив, той трябва да се разположи в паметта с помощта на оператора за заделяне на памет. Синтаксисът на оператора за заделяне на памет за деклариран масив е:

**<име на масив> = new <тип> [<размер>];**

Например: `a = new int [8]; point = new double[2];`

Подобно на инициализацията на променливите, заделянето на памет може да стане в оператор за деклариране на масива.

Например: `int [ ] a=new int[8];`

След като е указана дължината на масива, например `n`, в програмата могат да се използват само елементи с индекси от `0` до `n-1`.



### 3. ИНИЦИАЛИЗАЦИЯ НА МАСИВ, ВЪВЕЖДАНЕ И ИЗВЕЖДАНЕ НА ЕЛЕМЕНТИТЕ.

Инициализацията на масив се състои в инициализация на елементите му. По подразбиране елементите на целочислените масиви се инициализират с нули, масивите от тип `char` – с малки латински букви `'a'`, а от тип `string` – с празни низове. Ако потребителя иска друга инициализация има няколко начина да се направи това:

- В оператор за деклариране

Например: `int [ ] a = {1,2,3,4,5,6,7,8}`

При този начин стойностите на елементите се разделят един от друг със запетая и се заграждат в къдрави скоби. В този случай няма нужда от служебната дума `new` и задаване на размера.



За масива се отделя толкова място в паметта колкото са елементите в инициализиращия списък.

- В кода на програмата. В този случай даването на стойност става с оператор за присвояване за всеки елемент поотделно. Например:
- $A[0] = 1; a[1]=2; \text{ и т.нат.}$
- В цикъл  $\text{for}(i=0; i \leq n; i++) a[i] = I;$

Въвеждане на елементите на масив:

```
int n, i;
```

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

```
int [ ] array = new int[n];
```

```
for(i=0; i<n; i++)
```

```
{array[i] = int.Parse(Console.ReadLine( ));}
```



Потребителят сам определя размера на масива по време на изпълнение на програмата. В примера броят на елементите на масива се въвежда в променливата  $n$  и след това се декларира самия масив с размер  $n$ .

Извеждане на елементите на масив:

```
for (int i=0; i<5; i++)  
{Console.WriteLine(a[i]);}
```



## 4. РАБОТА С КОМПЮТЪР.

Зад1: Зададени са цели числа в определен ред. Да се въведат в компютъра и да се изведат в конзолата в ред, обратен на реда на въвеждането им.





```
static void Main(string[] args)
{
    int i, n = 10;
    int[] array = new int[n];
    for ( i=0; i<n; i++)
    {array[i] = int.Parse(Console.ReadLine);}
    for (i = n - 1; i >= 0; i--)
    { Console.WriteLine(array[i]); }
```



C:\Windows\system32\cmd.exe

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
Press any key to continue . . .
```