



# Массивы

# Определения

**Массив** – группа элементов одного типа, объединенных под общим именем.

**Индекс** – что-то (чаще всего номер), что позволяет отличать элементы массива один от другого и обращаться к ним.

## Особенности:

- Все элементы имеют **один тип**
- весь массив имеет **одно имя**
- все элементы расположены в памяти **рядом**

Обычную книгу  
можно считать  
своего рода  
массивом.  
Почему?

Ответ: книга состоит из множества однотипных элементов – страниц, у каждой страницы есть номер (индекс), все страницы объединены под одним названием (название книги)



# Дом также можно считать массивом. Почему?

Ответ: дому соответствует один почтовый адрес (город, улица, номер). Элементами дома можно считать квартиры, у каждой из которых есть номер (индекс).



# Шахматную доску можно считать массивом. Почему?

Доска состоит из клеток, каждая  
клетка обозначается буквой и  
цифрой (двойной индекс).



Футбольную  
команду можно  
считать  
«массивом».  
Почему?

*Ответ: команда состоит из  
нескольких людей, у  
каждого из них есть  
номер (индекс).*



# Описание массивов

Массивы описываются в разделе описания переменных. Общий вид описания:

*<имя массива>: array [<начальный индекс>..*конечный индекс*>] of <тип элемента>;*

Примеры:

**A : array [1..10] of real;**

Создается массив из действительных чисел (real), с нумерацией от 1 до 10 (всего 10 элементов).



# Примеры

**M : array [-5..5] of integer;**

Сколько элементов в этом массиве? Какого они типа? Как они нумеруются?

**Z : array [-8..-1] of integer;**

Сколько элементов в этом массиве? Какого они типа? Как они нумеруются?

**Mass : array [5..15] of integer;**

Сколько элементов в этом массиве? Какого они типа? Как они нумеруются?



# Использование элемента массива

Для использования элемента указывается имя массива и в квадратных скобках индекс этого элемента.

Примеры:

**Read(A[1]);**

**A[3]:= (2 \* x + 6) mod 4;**

**A[4]:= (A[1]+A[2]) div A[3];**

**Допустимые операции определяются типом элемента (элемент массива ничем не отличается от обычной переменной, кроме особенного написания имени).**

# Ввод массива

Пусть массив описан так:

**A : array [1..10] of real;**

Тогда для ввода массива можно использовать цикл:

**For i:=1 to 10 do read( A[i] );**

*Вопрос: почему не repeat или while?*

*Вопрос: как будет выглядеть вывод всех элементов массива? Вывод первой половины массива?*

*Вопрос: как изменится ввод массива, если описание будет таким:*

**A : array [-8..8] of integer;**

# Массивы

## Объявление:

```
const N = 5;  
var a: array[1..N] of integer;  
    i: integer;
```

## Ввод с клавиатуры.

```
for i:=1 to N do begin  
    write('a[', i, ']=');  
    read ( a[i] );  
end;
```

```
a[1] = 5  
a[2] = 12  
a[3] = 34  
a[4] = 56  
a[5] = 13
```

## Выход

```
for i:=1 to N do a[i]:=a[i]*2;
```

```
writeln('Массив A:');  
for i:=1 to N do  
    write(a[i]:4);
```

```
Массив A:  
10  24  68 112  26
```

## Заполнение массива случайными числами

- «*Случайным* называется число, появление которого не связано ни с какой закономерностью. Например, случайным является число, выпадающее при бросании кубика для игры в кости. А сам кубик можно рассматривать как **генератор случайных чисел**. Во всех языках программирования реализованы генераторы случайных чисел.

- В Паскале для «включения» генератора случайных чисел используется функция ***Randomize***, а само случайное число можно получить, если использовать функцию ***Random***.
- Например: ***y:=Random(x);***
- Здесь ***y*** — целое случайное число в интервале от ***0*** до ***x—1***, ***x*** — целое число, задающее верхнюю границу интервала случайных чисел.
- Чтобы получить случайное число в интервале от ***A*** до ***B*** (***A < B***), надо использовать следующее присваивание:
- ***y:=Random(B-A+1)+A;***

# Заполнение массива случайными числами

*Заполнить массив M, состоящий из 9 элементов случайными числами в интервале от 50 до 150 и вывести его на экран.*

```
program qq;  
var M: array [1..9] of integer;  
    i: integer;  
begin  
    writeln('Исходный массив:');  
    for i:=1 to 9 do begin  
        M[i] := random(100) + 50;  
        write( M[i]:4);  
    end;  
end.
```

# Заполнение массива константами

*элементы массива — константы.*

В этом случае заполнение происходит в разделе `const`. В этом разделе после описания массива надо в круглых скобках через запятую указать конкретные значения элементов. Например, рассмотрим такую задачу: заполнить массив `K[8]` константами: 4, 6, 90, 23, 56, 79, 34, 12. Решение будет таким:

```
const K: array[1..8] of Integer = (4, 6, 90, 23, 56,  
79, 34, 12);
```

Вывод массива осуществляется в самой программе так же, как было рассмотрено ранее



# Заполнение массива константами

```
program qq;  
const M: array [1..8] of integer=(4, 6, 90, 23, 56, 79, 34, 12);  
var i: integer;  
begin  
    writeln('Исходный массив:');  
    for i:=1 to 8 do begin  
write('M[' , i , ']=' , M[i]:4);  
    end;  
end.
```


# Заполнение массива, элементы которого связаны некоторой закономерностью

*элементы массива связаны некоторым законом.*

- Например, элементы массива — нечетные положительные числа от 1 до 23, т. е. массив состоит из чисел 1, 3, 5, 7, 9, ... , 23, т.е. значение элемента связано с его индексом по закону:  $M[i] = 2*i - 1$  для  $i$  от 1 до 12. Количество элементов такого массива можно посчитать по формуле:  $n = (23 - 1)/2 + 1$ .
- Элементы массива — положительные четные числа, то используется другое соотношение:  $M[i] = 2*i$ , количество элементов вычисляется аналогично.

- Заполнить массив нечетными целыми числами в интервале от 1 до 41 и вывести его на экран.
- В этой задаче надо посчитать количество элементов массива по формуле  $n = (41-1)/2 + 1 = 21$ .
- Остальное решение не требует пояснения, и можно сразу написать программу:

```
Program prog8;  
uses Crt;  
var  
  b: array[1..21] of Integer;  
  j: Integer;  
Begin  
  ClrScr;  
  for j:=1 to 21 do b[j]:=2*j-1  
  for j:=1 to 21 do  
    Write(b[j], ' ');  
  Readln; end.
```



Задача. Дано 5 целых чисел. Опишите алгоритм (программу) нахождения наибольшего среди них.

## Пример программы без использования массивов

```
program Maximum;  
var a,b,c,d,e : integer;  
begin  
  read(a,b,c,d,e);  
  if (a>=b) and (a>=c) and (a>=d) and (a>=e) then write(a);  
  if (b>=a) and (b>=c) and (b>=d) and (b>=e) then write(b);  
  if (c>=a) and (c>=b) and (c>=d) and (c>=e) then write(a);  
  if (d>=a) and (d>=b) and (d>=c) and (d>=e) then write(a);  
  if (e>=a) and (e>=b) and (e>=c) and (e>=d) then write(a);  
end.
```

*Вопрос: как изменится программа, если чисел станет 10?*

*Вопрос: как изменится программа, если чисел станет 1000?*

# Пример программы с использованием массива

```
program Maximum;  
var    A : array [1..5] of integer;  
        i, max : integer;  
begin  
    for i:=1 to 5 do read( A[i] );  
    max:= A[1];  
    for i:=2 to 5 do if A[i] > max then max:= A[i];  
    write(max);  
end.
```

*Вопрос: как изменится программа, если чисел станет 10?*

*Вопрос: как изменится программа, если чисел станет 1000?*

*Вопрос: как изменится программа, если числа будут действительными?*

- Что такое массив?
- Что такое элемент массива?
- Что такое индекс массива?
- Что такое размерность массива?
- Как можно обратиться к ячейке массива?
- Какого типа могут быть элементы массива?
- Какого типа может быть индекс массива?
- Как объявить массив?
- Как заполнить массив с клавиатуры.?  
(запишите блок ввода массива)
- Как заполнить массив случайными числами?  
(запишите блок ввода массива)
- Напишите программу нахождения суммы элементов массива.
- Напишите программу нахождения произведения элементов стоящих на четных местах.