



• **Массив** представляет собой совокупность данных одного типа с общим для всех элементов именем.

- Массив относится к структурированным типам данных (упорядоченная совокупность данных).
- Номера элементов массива иначе называются **индексами**, а сами элементы массива — **переменными с индексами (индексированными переменными)**.





Характеристики массива:

- ***тип*** — общий тип всех элементов массива;
- ***размерность*** (ранг) — количество индексов массива;
- ***диапазон изменения индекса*** (индексов) — определяет количество элементов в массиве





Способы описания массива

```
var ИмяМассива: array [НижняяГраница.. ВерхняяГраница] of Тип Элементов;
```

Например:

```
Const n=100;
```

```
var a: array[1..n] of real; { 100 элементов — вещественные числа }
```

```
b: array[0..50] of char; { 51 элемент — символы }
```

```
c: array[-3..4] of boolean; { 8 элементов — логические значения }
```

```
x,y: array[1..20] of integer; { два массива x и y содержат по 20  
элементов — целые числа }
```





Способы описания массива

Массив можно описать как типизированную константу.

Например:

```
const x: array[1..5] of integer=(1,3,5,7,9);
```

В этом примере не просто выделяется память под массив, а происходит заполнение ячеек заданными значениями по строкам.





Способы описания массива

Предварительное описание типа в разделе описания
ТИПОВ ДАННЫХ.

```
Type ИмяТипа = аггау [НижняяГраница.. ВерхняяГраница ] of Тип  
Элементов;
```

```
Var ИмяМассива : ИмяТипа;
```

Например.

```
Type z: array[1..20] of integer;
```

```
Var x, y: z
```





1. Как вы думаете, при выполнении программы обязательно заполнять все ячейки данными? Почему?
2. Если ячейка не заполнена то, какое значение в ней находится?
3. Может ли реальное количество элементов в массиве может быть меньше, чем указано при описании? Почему?
4. А может быть больше? Почему?





Способы заполнения массива

1. Ввод данных с клавиатуры:

```
for i:=1 to n do read (a[i]);
```

2. Ввод данных с помощью датчика случайных чисел.
Например. Заполним массив числами в диапазоне от -3 до 7.

```
randomize;
```

```
for i:=1 to n do a[i]:=random(11)-3;
```

3. Считывая значения элементов из файла:

```
for i:=1 to n do read (f, a[i]);
```





Способы заполнения массива

4. Присваивание заданных значений;

Например. Заполним массив четными числами

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

или

```
for i:=1 to n do begin
```

```
  readln (x);
```

```
  if x mod 2=0 then a[i]:=x
```

Вывод элементов массива

Вывод элементов массива осуществляется в цикле:

```
for i:=1 to n do write (a[i], ' ')
```





Действия с одномерными массивами

Например.

Var A, B: array[1..n] of integer;

Выражение	Результат
$A=B$	истина, если значение каждого элемента массива A равно соответствующему значению элемента B.
$A \neq B$.	результат истина, если хотя бы одно значение элемента массива A не равно значению соответствующего элемента массива B
$A:=B$	все значения элементов массива B присваиваются соответствующим элементам массива A.





Действия над элементами массива.

Вычислим сумму элементов.

...

```
Const n=10;
```

```
Var a:array[1..n] of integer; {описываем массив a}
```

```
i, s: integer;
```

```
begin
```

```
randomize; s:=0;
```

```
for i:=1 to n do begin
```

```
a[i]:=random(11)-3; {заполняем массив a случайными числами }
```

```
write (a[i], ' '); {вывожу заполненный массив}
```

```
end;
```

```
for i:=1 to n do s:=s+a[i]; {находим сумму элементов массива a}
```

```
writeln ('сумма элементов массива =', s) {выводим ответ }
```

```
end.
```





Действия над элементами массива.

Например: найти произведение элементов имеющих нечетный индекс.

...

```
Const n=10;
```

```
Var a:array[1..n] of integer; {описываем массив a}
```

```
i, p: integer;
```

```
begin
```

```
  randomize; p:=1;
```

```
  for i:=1 to n do begin
```

```
    a[i]:=random(11)-3; {заполняем массив a случайными числами }
```

```
    write (a[i], ' '); {вывожу заполненный массив}
```

```
  end;
```

```
  for i:=1 to n do if i mod 2<>0 then p:=p*a[i] {находим произведение  
    элементов массива a имеющих нечетный индекс}
```

```
  writeln ('произведение элементов массива =', s) {выводим ответ }
```

```
end.
```





Действия над элементами массива.

Например, найти номер первого из элементов массива А, имеющего значение равное нулю. Если таких элементов нет, вывести соответствующее сообщение.

```
Const n=10;
```

```
Var a:array[1..n] of integer; i, p: integer;
```

```
begin
```

```
randomize; p:=1;
```

```
for i:=1 to n do begin
```

```
a[i]:=random(11)-3; {заполняем массив а случайными числами }
```

```
write (a[i], ' '); {вывожу заполненный массив }
```

```
end; i:=1;
```

```
Repeat
```

```
i:=i+1;
```

```
until (a[i]=0) or (i=n) ; выход из цикла, когда нашли нужный элемент или массив закончился }
```

```
if a[i]=0 then writeln ('номер первого нулевого элемента=', i)
```

```
else writeln (' таких элементов нет!'); end.
```





Действия над элементами массива.

Поиск максимального (минимального) элемента и его номера.

Например, в одномерном массиве подсчитать количество элементов равных минимальному.





Подведение итогов урока

- Чем ценны массивы?
- Каким образом задается описание массива, что в нем указывается?
- Каким образом задается обращение к элементу массива?
- Почему при описании массива предпочтительнее употреблять константы , а не указывать размеры массива в явном виде?

