



# ОДНОМЕРНЫЕ МАССИВЫ ЦЕЛЫХ ЧИСЕЛ

## АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

9 класс

# Ключевые слова

- массив
- описание массива
- заполнение массива
- вывод массива
- обработка массива
- последовательный поиск
- сортировка



# Массив

**Массив** - это поименованная совокупность однотипных элементов, упорядоченных по индексам, определяющим положение элемента в массиве.

## Одномерный массив

Значение элемента массива					Индекс элемента массива			
1	2	3	4	5	6	7	8	9
<b>7</b>	<b>-5</b>	<b>9</b>	<b>1</b>	<b>0</b>	<b>-2</b>	<b>4</b>	<b>3</b>	<b>6</b>

Решение разнообразных задач, связанных с обработкой массивов, базируется на решении таких типовых задач, как:

- суммирование элементов массива;
- поиск элемента с заданными свойствами;
- сортировка массива.

# Описание массива

Общий вид описания массива:

```
var <имя_массива>: array [<мин_знач_индекса> ..  
<макс_знач_индекса>] of <тип_элементов>;
```

Имя массива

Тип элементов  
массива

```
var a: array [1..10] of integer;
```

Минимальное значение  
индекса

Максимальное значение  
индекса

Значение 1-го  
элемента  
массива



```
const b: array [1..5] of integer = (4, 2, 3, 5, 7);
```

Массив **b** с постоянными значениями описан в разделе описания констант.

# Способы заполнения массива

**1 способ.** Ввод каждого значения с клавиатуры:

```
for i:=1 to 10 do read (a[i]);
```



**2 способ.** С помощью оператора присваивания (по формуле):

```
for i:=1 to 10 do a[i]:=i;
```

**3 способ.** С помощью оператора присваивания (случайными числами):

```
randomize;
```

```
for i:=1 to 10 do a[i]:=random(100);
```

$a[i]=2*a[i]+6$

37	58	45	58	58	38
38	59	46	59	59	39
39	01	47	01	01	01
40	02	48	02	02	02
41	03	49	03	03	03

# Вывод массива

**1 способ.** Элементы массива можно вывести в строку, разделив их пробелом:

```
for i:=1 to 10 do write (a[i], ' ');
```

```
45 21 56 43 83 56 69 34 28 15
```

**2 способ.** Вывод с комментариями:

```
for i:=1 to 10 do writeln ('a[', i, ']=', a[i]);
```

```
a[1]=4  
a[2]=1  
a[3]=6  
a[4]=3  
a[5]=8  
a[6]=5  
a[7]=9  
a[8]=4  
a[9]=8  
a[10]=7
```

# Заполнение массива A(10) случайными числами и вывод элементов массива

Объявление  
массива



Заполнение  
массива



Вывод  
массива

```
program n_2 ;  
  var i: integer;  
      a: array[1..10] of integer;  
  
  randomize;  
begin  
  for i:=1 to 10 do a[i]:=random(100);  
  
  for i:=1 to 10 do write (a[i],` `);  
end.
```

# Вычисление суммы элементов массива

Суммирование элементов массива осуществляется за счёт поочерёдного добавления слагаемых:

Определяется ячейка памяти (переменная  $s$ ), в которой будет последовательно накапливаться результат суммирования

Переменной  $s$  присваивается начальное значение  $0$  - число, не влияющее на результат сложения

Для каждого элемента массива из переменной  $s$  считывается её текущее значение и складывается со значением элемента массива; полученный результат присваивается переменной  $s$ .

# Вычисление суммы элементов массива

$s := 0$	$s = 0$
$s := s + a[1]$	$s = 0 + a[1]$
$s := s + a[2]$	$s = 0 + a[1] + a[2]$
$s := s + a[3]$	$s = 0 + a[1] + a[2] + a[3]$
...	...
$s := s + a[n]$	$s = 0 + a[1] + a[2] + a[3] + \dots + a[n]$

Основной фрагмент программы:

**$s := 0;$**

**for  $i := 1$  to  $n$  do  $s := s + a[i];$**

# Вычисление суммы элементов массива

```
program n_3;  
  var s, i: integer;  
      a: array[1..10] of integer;  
begin  
  s:=0;  
  randomize;  
  for i:=1 to 10 do a[i]:=random(100);  
  for i:=1 to 10 do write (a[i], ` `);  
  for i:=1 to 10 do s:=s+a[i];  
  writeln ('s=', s)  
end.
```



# Вычисление суммы элементов массива

```
program n_3_1;  
  var s, i: integer;  
      a: array[1..10] of integer;  
begin  
  s:=0;  
  randomize;  
  for i:=1 to 10 do  
    begin  
      a[i]:=random(100);  
      write (a[i],` `);  
      s:=s+a[i]  
    end;  
  writeln ('s=', s)  
end.
```



**Второй способ оформления:**  
Используем операторные скобки и всё (ввод, вывод, обработку) делаем в одном цикле.

# Вычисление суммы элементов массива

The screenshot shows the PascalABC.NET 3.7.1 IDE. The main window displays the following Pascal code:

```
program summa;  
var  
    mas: array [1..7] of integer;  
    s,i: integer;  
begin  
    s:=0;  
    for i:=1 to 7 do  
        begin  
            mas [i]:=i;  
            s:=s+mas[i];  
        end;  
    writeln (mas);  
    writeln (s);  
end.
```

At the bottom of the IDE, the output window shows the results of the program execution:

```
Окно вывода  
[1, 2, 3, 4, 5, 6, 7]  
28
```

## Типовые задачи поиска

Нахождение наибольшего (наименьшего) элемента массива

Нахождение элемента массива, значение которого равно заданному значению

## Нахождение наибольшего элемента в стопке карточек с записанными числами:

1) Взять верхнюю карточку, записать на доске (запомнить) число как наибольшее.

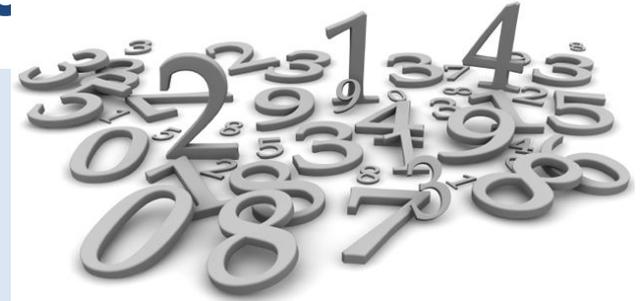
2) Взять следующую карточку, сравнить числа. Если на карточке число больше, то стереть старую запись и записать это число.

3) Повторить действия, описанные в пункте 2 для всех оставшихся карточек.

**!** При организации поиска наибольшего элемента массива правильнее искать его индекс.



# Программа поиска наибольшего элемента в массива



```
program n_4;  
  var i, imax: integer;  
      a: array[1..10] of integer;  
begin  
  randomize;  
  for i:=1 to 10 do a[i]:=random(100);  
  for i:=1 to 10 do write (a[i], ' ');  
  imax:=1;  
  for i:=2 to 10 do  
    if a[i]>a[imax] then imax:=i;  
  write ('Наибольший элемент a[' , imax, ']=', a[imax])  
end.
```

# Программа поиска наибольшего элемента в массиве

The image shows a screenshot of the PascalABC.NET 3.7.1 IDE. The main window displays the source code for a Pascal program named Program3.pas. The code generates a random array of 6 integers and finds the maximum value.

```
Program z1;  
var a: array [1..6] of integer;  
    i, max: integer;  
begin  
    for i:=1 to 6 do  
    begin  
        a[i]:=random (100);  
        writeln (a[i]);  
    end;  
  
    max:=-11;  
  
    for i:=1 to 6 do  
    begin  
        if ( a[i])> max then max:=a[i];  
    end;  
    writeln (max);  
end.
```

The output window at the bottom shows the results of the program's execution:

```
47  
65  
45  
18  
31  
9  
65
```

The status bar at the bottom indicates: "Компиляция прошла успешно (20 строк), 2 предупреждений" (Compilation successful (20 lines), 2 warnings).

# Опорный конспект

**Массив** - это поименованная совокупность однотипных элементов, упорядоченных по индексам, определяющим положение элементов в массиве.

```
var <имя_массива>: array [<мин_знач_индекса> ..  
<макс_знач_индекса>] of тип_элементов;
```

## Заполнение массива

Ввод с клавиатуры

Присваивание значений

## Задачи по обработке массива

Суммирование элементов

Сортировка элементов  
массива

Поиск элементов массива с заданными  
свойствами

Домашняя работа:

1. Написать программу поиска наименьшего элемента массива.
2. Написать программу вычисления суммы чётных элементов массива.