

**Массив – пронумерованная совокупность ячеек памяти, названная одним именем**

1	2	3	4	5	6	7
44	21				30	19

Имя массива

Индекс  
(порядковый  
номер) элемента  
массиваЗначение  
элемента  
массива

**A [ 4 ] := 11;**

# ОДНОМЕРНЫЕ МАССИВЫ ЦЕЛЫХ ЧИСЕЛ

**Const a:array[1..5] of integer =**

Массив констант

**(3,-2,1,4,3);**

**Var R:Array [-20..20] of Real;**

**Var N:Array ['A'..'Z'] of Integer;**

В зависимости от задачи индексы элементов могут начинаться не только с единицы или быть символьного типа

# Ключевые слова

- массив
- описание массива таблица
- 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
7	-5	9	1	0	-2	4	3	6

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.

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

**Объявление  
массива**



**Заполнение  
массива**



**Вывод  
массива**

```
program n_1 ;  
  var i: integer;  
      a: array[1...10] of integer;  
  
begin  
  for i:=1 to 10 do a[i]:=random(50);  
  
  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

Присваивается переменной  $s$  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

$$s = 0$$

$$s = 0$$

$$s = s + a[1]$$

$$s = 0 + a[1]$$

$$s = s + a[2]$$

$$s = 0 + a[1] + a[2]$$

$$s = s + a[3]$$

$$s = 0 + a[1] + a[2] + a[3]$$

...

...

$$s = s + a[10]$$

$$s = 0 + a[1] + a[2] + a[3] + a[10]$$

Основной фрагмент программы:

```
s:=0;
```

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

# Вычисление суммы элементов массива

```
program n_2;  
  var s, i: integer;  
      a: array[1..10] of integer;  
begin  
  s:=0;  
  for i:=1 to 10 do a[i]:=random(50);  
  for i:=1 to 10 do write (a[i], ' ');  
  for i:=1 to 10 do s:=s+a[i];  
  writeln ('s=', s);  
end.
```



## 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 записать это число.

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 массиве



```
program n_3;  
  var s, i, imax: integer;  
      a:array[1...10] of integer;  
begin  
  s:=0;  
  for i:=1 to 10 do a[i]:=random(50);  
  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.
```

# Нахождение элемента массива с заданными свойствами

Результатом поиска элемента, значение которого равно заданному, может быть:

- $n$  - индекс элемента массива такой, что  $a[n] = x$ , где  $x$  - заданное число;
- сообщение о том, что искомого элемента в массиве не обнаружено.

10	50	1	3	50	14	21	50	10	21
----	----	---	---	----	----	----	----	----	----

Здесь:

- трём равен 4-й элемент;
- десяти равны 1-й и 9-й элементы;
- нет элемента, равного 12.

# Поиск элемента, равного 50

```
program n 4;  
  var n, i: integer;  
      a: array[1..10] of integer;  
begin  
  for i:=1 to 10 do a[i]:=random(60);  
  for i:=1 to 10 do write (a[i], ' ');  
  n:=0;  
  for i:=1 to 10 do  
    if a[i]=50 then n:=i;  
    if n=0 then write('Нет') else write (i)  
  end.
```

В программе найден последний из элементов, удовлетворяющих условию.

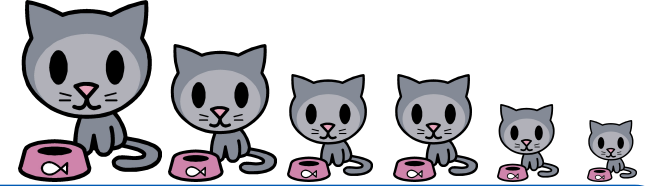
10	50	1	3	50	14	21	50	10	21
----	----	---	---	----	----	----	----	----	----

# Сумма значений элементов, удовлетворяющих условию

```
program n_7;  
  var s, i: integer;  
      a: array[1..10] of integer;  
begin  
  for i:=1 to 10 do a[i]:=random(60);  
  for i:=1 to 10 do write (a[i], ' ');  
  s:=0;  
  for i:=1 to 10 do  
    if a[i]>10 and (a[i]<30 then s:=s+a[i];  
  write('s=', s)  
end.
```

10	50	1	3	50	14	21	50	10	21
----	----	---	---	----	----	----	----	----	----

# Сортировка массива



Сортировка элементов массива по убыванию выбором осуществляется следующим образом:

1. В массиве выбирается максимальный элемент

2. Максимальный и первый элемент меняются местами (первый элемент считается отсортированным)

3. В неотсортированной части массива снова выбирается максимальный элемент; он меняется местами с первым неотсортированным элементом массива

Действия пункта 3 повторяются с неотсортированными элементами массива, пока не останется один неотсортированный элемент (минимальный)

# Сортировка массива



Индекс	1	2	3	4	5	6	7	8	
Значение	0	1	9	2	4	3	6	5	
Шаги	1	0	1	9	2	4	3	6	5
	2	9	1	0	2	4	3	6	5
	3	9	6	0	2	4	3	1	5
	4	9	6	5	2	4	3	1	0
	5	9	6	5	4	2	3	1	0
	6	9	6	5	4	3	2	1	0
	7	9	6	5	4	3	2	1	0
	Итого:	9	6	5	4	3	2	1	0

# Сортировка массива

0	1	9	2	4	3	6	5
---	---	---	---	---	---	---	---

```
program n_8;  
  var n, i, j, x, imax: integer;  
      a:array[1...10] of integer;  
begin  
  for i:=1 to 10 do read (a[i]);  
  for i:=1 to 10 do write (a[i], ' ');  
  for i:=1 to 9 do  
  begin  
    imax:=i;  
    for j:=i+1 to 10 do if a[j]>a[imax] then imax:=j;  
    x:=a[i];  
    a[i]:=a[imax];  
    a[imax]:=x  
  end;  
  for i:=1 to 10 do write (a[i], ' ');  
end;
```

9	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---



# Самое главное

**Массив** — это именованный совокупность однотипных элементов, упорядоченных по индексам, определяющим положение элементов в массиве. В языках программирования массивы используются для реализации таких задач, связанных как с обработкой массивов, так и с базированием на таких типовых задачах, как:

- использование в программе массив должен быть описан
- суммирование элементов массива,

```
тип_элемента массива > [имя_массива [значение_индекса] ...  
<символ_закрывающей_скобки массива >] of тип_элементов;
```

# Вопросы и задания

неделю от температуры воздуха. С помощью этих данных в 32. Дана функция, описывающая зависимость от времени  $t$  в часах количества пассажиров, писавших диктанты по русскому языку в течение некоторого периода времени. Определите, какому количеству пассажиров пришлось писать диктанты в субботу и воскресенье.

День	Количество пассажиров
Суббота	16
Воскресенье	14

# 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

Поиск элемента во  
свойствам

Сортировка элементов массива

**СПАСИБО ЗА  
ПРОСМОТР !!!!!**