

---

# **Глава 2. Программирование**

## **§18. Массивы**

---

# Массивы



*Практическая работа 17. Заполнение массивов*

## Что такое массив?

---

В программах, с которыми мы работали раньше, было всего несколько переменных. Каждой из них мы давали своё имя, и никаких сложностей при этом не возникало.

Объёмы данных, которые обрабатывают современные компьютеры, огромны: количество значений измеряется миллионами и миллиардами.

Если каждую из этих переменных называть своим именем, очень легко запутаться, и работать с ними очень неудобно.

## Что такое массив?

---

Допустим, мы хотим сложить значения 1000 ячеек с именами  $a_1, a_2, \dots, a_{1000}$ . Для этого нужно будет написать очень длинный оператор присваивания:

$$\text{sum} := a_1 + a_2 + \dots + a_{1000};$$

Придётся перечислить все 1000 имён переменных.

**Массив** – это группа переменных одного типа, расположенных в памяти друг за другом и имеющих общее имя.

## Что такое массив?

---

Чтобы использовать массив, надо его объявить – присвоить ему имя, определить тип входящих в массив переменных (*элементов массива*) и их количество.

По этим сведениям компьютер вычислит, сколько места требуется для хранения массива, и выделит в памяти нужное число ячеек.

Имена (*идентификаторы*) массивов строятся по тем же правилам, что и имена переменных.

## Что такое массив?

---

В алгоритмическом языке массивы называются таблицами. При их объявлении к названию типа данных добавляется **таб**:

**целтаб A[1:5]**  
**вещтаб V[0:5]**

В квадратных скобках через двоеточие записывают границы *индексов* – номеров элементов массива.

## Что такое массив?

---

**Индекс** – это значение, которое указывает на конкретный элемент массива.

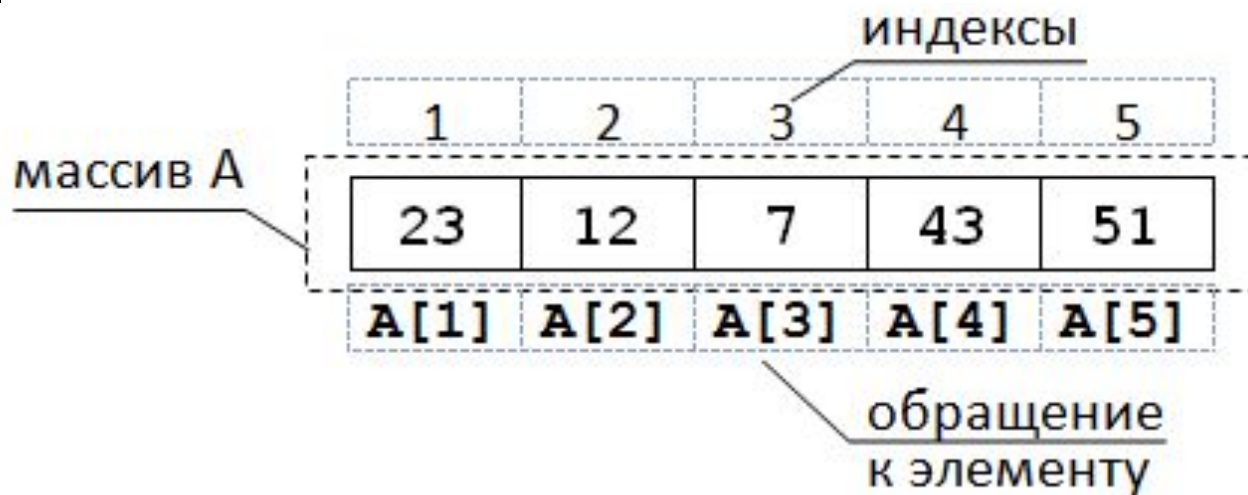
Массив  $A$  – это массив целых значений, элементы имеют индексы от 1 до 5. Массив вещественных значений  $V$  содержит 6 элементов с индексами от нуля до 5.

В языке Паскаль массивы объявляются в блоке объявления переменных.

```
var A: array[1..5] of integer;  
      V: array[0..5] of real;
```

## Что такое массив?

Для того чтобы обратиться к элементу массива (*прочитать или изменить его значение*), нужно записать имя массива и в квадратных скобках – индекс нужного элемента, например,  $A[3]$ .



Индексом может быть также значение целой переменной или арифметического выражения, результат которого – целое число.



## Что такое массив?

---

**Выход за границы массива** – это обращение к элементу с индексом, который не существует в массиве.

При выходе за границы массива программа обычно завершается аварийно.

Будем во всех программах использовать привычную для человека нумерацию с единицы, считая, что массив  $A$  объявлен так:

**цел  $N = 10$**

**целтаб  $A[1:N]$**

**`const N = 10;`**

**`var A: array[1..N] of integer;`**

## Что такое массив?

---

цел  $N = 10$

целтаб  $A[1:N]$

```
const N = 10;
```

```
var A: array[1..N] of integer;
```

Здесь размер массива обозначен как  $N$ . В программе, как правило, размер массива встречается во многих командах, и при его изменении нужно исправить число только в одном месте программы.

В программе на Паскале размер массива объявлен как *константа* с помощью ключевого слова *const*.

## Перебор всех элементов массива

---

Перебор элементов состоит в том, что мы в цикле просматриваем все элементы массива и, если нужно, выполняем с каждым из них некоторую операцию.

Для этого удобнее всего использовать цикл по переменной, которая изменяется от минимального до максимального индекса.

Для массива, элементы которого имеют индексы от 1 до  $N$ , цикл выглядит так:

**нц для  $i$  от 1 до  $N$**

**...**

**кц**

**for  $i:=1$  to  $N$  do begin**

**...**

**end;**

## Перебор всех элементов массива

---

Заполним массив первыми  $N$  натуральными числами в обратном порядке: в первый элемент массива должно быть записано число  $N$ , во второй – число  $N-1$ , а в последний – единица.

Сначала запишем цикл в развёрнутом виде: операторы, которые должны быть выполнены:

<b>A[1]:= N</b>	<b>A[1]:= N;</b>
<b>A[2]:= N-1</b>	<b>A[2]:= N-1;</b>
<b>...</b>	<b>...</b>
<b>A[N]:= 1</b>	<b>A[N]:= 1;</b>

## Перебор всех элементов массива

---

Теперь запишем цикл:

```
нц для i от 1 до N
  A[i]:= X
кц
```

```
for i:=1 to N do
  A[i]:= X;
```

Величина  $X$  должна изменяться при переходе к следующему элементу.

Можно записать цикл так:

```
X:=N
нц для i от 1 до N
  A[i]:= X
  X:=X-1
кц
```

```
X:=N;
for i:=1 to N do begin
  A[i]:= X;
  X:=X-1;
```

## Перебор всех элементов массива

---

А можно его значительно упростить, заметив, что при увеличении номера элемента  $i$  на единицу значение  $X$  уменьшается, причём тоже на единицу. Поэтому сумма  $i+X$  остаётся постоянной! Её можно вычислить, например, для первого элемента: она равна  $1+N$ .

В элемент с номером  $i$  записывается значение  $N+1-i$ , поэтому цикл можно записать так:

нц для  $i$  от 1 до  $N$

$A[i] := N+1-i$

кц

for  $i:=1$  to  $N$  do

$A[i] := N+1-i;$

## Перебор всех элементов массива

---

Теперь предположим, что массив заполнен и попробуем увеличить все его элементы на единицу. Это значит, что нужно заменить значение элемента  $A[i]$  на  $A[i]+1$ :

нц для  $i$  от 1 до  $N$

$A[i] := A[i] + 1$

кц

for  $i:=1$  to  $N$  do

$A[i] := A[i] + 1;$

## Вывод массива

---

Массив – это набор элементов, поэтому в большинстве языков программирования нельзя вывести массив одной командой. Для этого используют цикл, в котором на каждом шаге выводится один элемент:

**нц для  $i$  от 1 до  $N$   
вывод  $A[i]$ , ' '  
кц**

**for  $i:=1$  to  $N$  do  
write( $A[i]$ , ' ');**

**нц для  $i$  от 1 до  $N$   
вывод  $A[i]:4$   
кц**

**for  $i:=1$  to  $N$  do  
write( $A[i]:4$ );**



## Ввод массива с клавиатуры

---

Иногда небольшие массивы вводятся с клавиатуры. В простейшем случае мы просто строим цикл, который выполняет оператор ввода для каждого элемента массива:

```
нц для i от 1 до N
  ввод A[i]
кц
```

```
for i:=1 to N do
  read(A[i]);
```

При этом пользователь вводит данные «вслепую», то есть программа не подсказывает ему, значение какого элемента вводится в данный момент.

## Ввод массива с клавиатуры

---

Значительно удобнее, если перед вводом появляется сообщение с подсказкой:

**нц для i от 1 до N**

**вывод 'A[' ,i,']='**

**ввод A[i]**

**кц**

**for i:=1 to N do begin**

**write('A[' ,i,']=');**

**read(A[i])**

**end;**

## Заполнение массива случайными числами

---

В учебных примерах массивы часто заполняют случайными числами, например, так:

```
нц для i от 1 до N  
  A[i]:=irand(20,100)  
  вывод A[i], ' '  
кц
```

```
for i:=1 to N do begin  
  A[i]:=20+random(81);  
  write(A[i],' '  
end;
```

Элементы массива сразу же выводятся на экран (в одну строку через пробелы).

## Выводы:

---

- Массив – это группа переменных одного типа, расположенных в памяти друг за другом и имеющих общее имя. Массивы используют для того, чтобы было удобно работать с большим количеством данных.
- Индекс элемента массива – это значение, которое указывает на конкретный элемент массива.
- При обращении к элементу массива индекс указывают в квадратных скобках. Это может быть число, имя переменной целого типа или арифметическое выражение, результат которого – целое число.
- Перебор элементов массива – это выполнение какой-то операции с каждым элементом. Для этого удобно использовать цикл по переменной, которая изменяется от минимального до максимального значения индекса.
- Массив вводится и выводится поэлементно, как правило, с помощью цикла.

# Интеллект-карта

---