

Программирование (Паскаль)

§ 21. Массивы

Что такое массив?



Как ввести 10000 переменных?

Массив – это группа переменных одного типа, расположенных в памяти рядом (в соседних ячейках) и имеющих общее имя.

Надо:

- выделять память
- записывать данные в нужную ячейку
- читать данные из ячейки

Выделение памяти (объявление)

! Массив = таблица!

МИНИМАЛЬНЫЙ
ИНДЕКС

МАКСИМАЛЬНЫЙ
ИНДЕКС

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

Индекс элемента — это значение, которое указывает на конкретный элемент массива.

```
const N = 10;
```

размер через
константу

? Зачем?

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

Что неправильно?

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

...

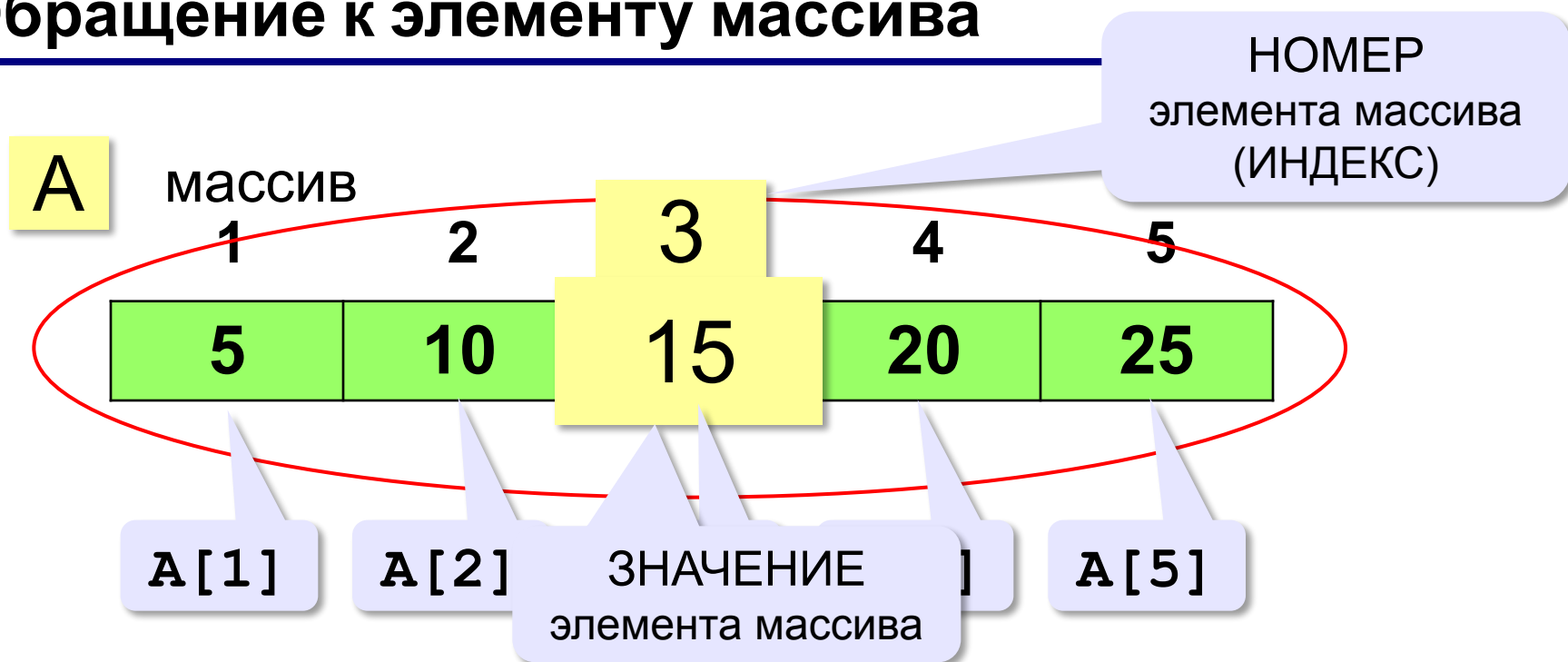
```
A[5] := 4.5;
```

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

...

```
A[15] := 'a';
```

Обращение к элементу массива



Обращение к элементу массива

1	2	3	4	5
23	12	7	43	51

```
var i: integer;  
i := 2;  
A[3] := A[i] + 2*A[i-1] + A[2*i];  
writeln( A[3]+A[5] );
```



Что получится?

```
A[3] := A[2] + 2*A[1] + A[4];  
writeln( A[3]+A[5] );
```

101
152

Что неверно?

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

```
x: integer;
```

```
...
```

```
x := 2;
```

```
writeln( A[x-3] );
```

```
A[x+4] := A[x-1] + A[2*x];
```



Что плохо?

```
writeln( A[-1] );  
A[6] := A[1] + A[4];
```

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

Перебор элементов массива

```
const N = 10;  
var A: array[1..N] of integer;
```

Перебор элементов: просматриваем все элементы массива и, если нужно, выполняем с каждым из них некоторую операцию.

```
for i:=1 to N do begin  
    { здесь работаем с A[i] }  
end;
```


Заполнение массива

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



Что произойдёт?

В развёрнутом виде

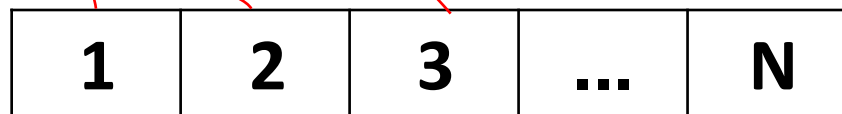
```
A[1] := 1;
```

```
A[2] := 2;
```

```
A[3] := 3;
```

...

```
A[N] := N;
```



Заполнение массива в обратном порядке

N	...	3	2	1
---	-----	---	---	---

```
A[1] := N;  
A[2] := N-1;  
A[3] := N-2;  
...  
A[N] := 1;
```

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

? Как меняется X?

X = N, N-1, ..., 2, 1

начальное
значение

уменьшение
на 1

Заполнение массива в обратном порядке

N	...	3	2	1
---	-----	---	---	---

$$A[i] := X;$$

? Как связаны i и X ?

i	X
1	N
2	N-1
3	N-2
...	...
N	1

Note: A callout bubble with "+1" points to the 'i' column, and another with "-1" points to the 'X' column.

```
for i:=1 to N do
  A[i] := N + 1 - i
end;
```

! Сумма i и X не меняется!

$$i + X = N + 1$$

$$X = N + 1 - i$$

Вывод массива на экран

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



Что плохо?

интервал между значениями

или так:

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

в столбик

или так:

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



Как убрать?

[1,2,3,4,5,]

Ввод с клавиатуры

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



Что плохо?

С подсказкой для ввода:

```
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

Заполнение случайными числами

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

сразу вывод на экран



Какой отрезок?

В других языках программирования

Python:

```
A = [0]*N
for i in range(N):
    A[i] = i + 1
print(A)
```



Нумерация элементов
всегда с нуля!

C++:

```
int A[N], i;
for (i = 0; i < N; i++)
    A[i] = i + 1;
for (i = 0; i < N; i++)
    cout << A[i] << " ";
```

Задачи

- «А»: а) Заполните все элементы массива значением X , введённым с клавиатуры.
- б) Заполните массив первыми N натуральными числами, начиная с X (значение X введите с клавиатуры).
- «В»: а) Заполните массив натуральными числами в обратном порядке, начиная со значения X , введённого с клавиатуры. Последний элемент должен быть равен X , предпоследний равен $X-1$ и т. д.
- б) Заполните массив степенями числа 2 (от 2^1 до 2^N), так чтобы элемент с индексом i был равен 2^i .

Задачи

- «С»:** а) Заполните массив степенями числа 2, начиная с конца, так чтобы последний элемент массива был равен 1, а каждый предыдущий был в 2 раза больше следующего.
- б) С клавиатуры вводится целое число X . Заполните массив, состоящий из нечётного числа элементов, целыми числами, так чтобы средний элемент массива был равен X , слева от него элементы стояли по возрастанию, а справа – по убыванию. Соседние элементы отличаются на единицу. Например, при $X = 3$ массив из 5 элементов заполняется так: 1 2 3 2 1.

Задачи-2

«А»: Напишите программу, которая заполняет массив из $N = 8$ элементов случайными числами в диапазоне $[0, 10]$, выводит его на экран, а затем выводит на экран квадраты всех элементов массива.

Пример:

Массив: 5 6 2 3 1 4 8 7

Квадраты: 25 36 4 9 1 16 64 49

«В»: Напишите программу, которая заполняет массив из $N = 10$ случайными числами в диапазоне $[100, 300]$ и выводит его на экран. После этого на экран выводятся средние цифры (число десятков) всех чисел, записанных в массив.

Пример:

Массив: 142 324 135 257 167 295 126 223 138 270

Число десятков: 4 2 3 5 6 9 2 2 3 7

Задачи-2

«С»: Напишите программу, которая заполняет массив из $N = 10$ случайными числами в диапазоне $[100, 500]$ и выводит его на экран. После этого на экран выводятся суммы цифр всех чисел, записанных в массив.

Пример:

Массив: 162 425 340 128 278 195 326 414 312 177

Суммы цифр: 9 11 7 11 17 15 11 9 6 15