

МАССИВЫ

ОДНОМЕРНЫЕ МАССИВЫ

Представление массива

При обработке набора данных одного типа в языке Паскаль имеется специальная структура, называемая массив.

Массив - это структура, представляющая собой упорядоченную совокупность элементов одного типа, объединенных одним именем.

Массив в Паскале имеет фиксированное количество элементов.

Имя массива, количество элементов и тип элементов массива определяются пользователем при описании массива.

Для получения доступа к элементу массива используется индекс.

Индекс массива - это величина целого типа, характеризующая положение элемента в массиве.

Каждому массиву, используемому в программе, выделяется место в памяти.

Массиву отводится не одна ячейка, а последовательность расположенных друг за другом ячеек, в каждую из которых записывается значение соответствующего элемента.

Пример:

Представим массив, состоящий из 88 элементов типа real, в котором элементы нумеруются от 11 до 99.

	x [11]	x [12]	x [13]	...	x [98]	x [99]
Индекс	11	12	13	...	98	99
Значение	1.23	1.61	0.01	...	1.34	1.91

Описание массива

Описание массива включает в себя:

1) Имя массива:

2) Ключевое слово `array`;

3) Диапазон изменения индексов, определяемый граничной парой, например: `1..40`. `1..N`. `5..20`.

Нижняя граница показывает наименьшее возможное значение индекса, верхняя - наибольшее. Нижняя граница не может превосходить верхнюю.

Нижняя граница отделяется от верхней двумя точками;

4) Тип элементов массива.

Описание массива строится по следующей схеме:

<имя массива> : array [граничная пара] of <тип элементов>
(массив) (из)

Если несколько массивов имеют один и тот же тип, и одинаковый диапазон изменения индекса, то их описание можно объединить, разделив имена массивов запятыми:

Var

< имя 1, имя2.....>: array [граничная пара] of <тип эл-тов>

Примеры:

Var

Vector: array [1 ..60] of integer;

xBlock: array [40..50] of real;

X1,x2: array [0..60] of integer;

Действия над элементами массива

После объявления массива каждый его элемент можно обработать, указав имя массива и индекс элемента в квадратных скобках.

Например, запись `xBlock [42]`, `Vektor[10]` позволяет обратиться к элементу массива `xBlock` с индексом 42 и элементу массива `Vektor` с индексом 10.

Заполнение массива.

Заполнить элементы одномерного массива значениями мы можем:

- вводя значения с клавиатуры;**
- случайным образом;**
- по формуле.**

Надо помнить, что во всех трех случаях нам не обойтись без организации цикла.

Будем считать, что объявили массив из 10 элементов (тип элементов в данном случае значения не имеет).

1. Заполнение с клавиатуры.

```
FOR i:=1 to 10 do  
begin  
writeln('введите значение элемента');  
readln(a[i]);  
end;
```

2. Заполнение случайным образом.

Для этого мы должны подключить датчик случайных чисел.

```
FOR i:=1 to 10 do  
a[i]:=random(n); {n должно быть заранее задано}  
RANDOM(N) – вещественное число из [0,N);  
TRUNC(RANDOM(N))- целое число из [0,N)  
TRUNC(RANDOM(N)+1)– целое число из [0,N]  
TRUNC(RANDOM(B-A)+A)– целое число из [A,B)
```

3. Заполнение по формуле.

Каждый элемент массива должен быть рассчитан по формуле (например, $a_i = \sin i - \cos i$).

```
FOR i:=1 to 10 do  
a[i]:=sin(i)-cos(i);
```


Печать линейного массива.

Линейную таблицу можно распечатать на экране в строчку или в столбик.

```
For i:=1 to 7 do
  begin
    write (TEMP[i] :5);
  end;
```

```
For i:=1 to 7 do
  begin
    writeln(TEMP[i]);
  end;
```

Вычисление суммы элементов

```
S:=0;  
FOR i:=1 to 10 do  
S := S + a[i];
```

Вычисление среднего значения элементов массива

```
S:=0;  
FOR i:=1 to 10 do  
S := S + a[i];  
SR:= S/10;
```

План программы для работы с массивами.

```
Program Task;  
Uses Crt; {очистка экрана}  
const n=...; (количество элементов массива)  
  var  
<имя массива>: array [1..n] of <тип>;  
  i:integer;.....  
BEGIN  
  writeln('Программу составили...');  
{заполнение массива одним из способов}  
  for i:=1 to n do  
    begin  
      .....  
    end;  
{Печать массива одним из способов}  
  for i:=1 to n do  
    begin  
      .....  
    end;  
  Команды обработки массива  
  Вывод результата  
END.
```

Поиск максимального (минимального) элемента массива.

Пусть мы имеем одномерный массив: 20,-2,4, 10,7,21,-12,0,4, 17.

Алгоритм поиска максимального (минимального) элемента мы построим так, чтобы сравнивать пару чисел, повторяя действие сравнения нужное количество раз.

Введем дополнительную переменную с именем `max`.

Она и будет одним из чисел, второе число — это очередной элемент массива.

Для того, чтобы провести первую операцию сравнения необходимо переменной `max` присвоить некоторое начальное значение.

Здесь могут быть два варианта:

- 1) присвоить переменной `max` первый элемент массива, тогда количество повторений операции сравнения равно $n - 1$;
- 2) присвоить число заведомо меньше всех элементов массива. В этом случае, количество повторений операции сравнения n .

ЗАДАНИЕ

Для заданного одномерного массива найти максимальный элемент.

program max;

Var

mas: array[1..10] of integer;

i: integer;

max: integer;

begin

{блок заполнения}

for i:=1 to 10 do

readln(a[i]);

{поиск максимального элемента}

max:=a[1];

for i:=2 to 10 do

if a[i]>max then max:=a[i];

writeln('максимальный элемент=',max)

end.

Пример 1

В одномерном массиве посчитать сумму положительных элементов

```
program sum;
uses crt;
var a:array[1..10] of integer;
    i,n,s:integer;
begin
  clrscr;
  write('Введите число элементов');
  read(n);
  for i:=1 to n do
    read(a[i]);
  for i:=1 to n do begin
    write(a[i]:4);
    if a[i]>0 then s:=s+a[i];
    end;
  gotoxy(5,10); {вывод в позиция экрана для TP}
  write('s= ',s);
end.
```

В одномерном массиве вычислить минимальный элемент

```
program minimum;  
uses crt;  
var a:array[1..10] of integer;  
    i,n,min:integer;  
begin  
    clrscr; write('Введите число элементов');  
    read(n);  
    for i:=1 to n do  
        read(a[i]); writeln;  
    for i:=1 to n do  
        write(a[i]:4);  
        min:=a[1];  
        for i:=2 to n do  
            if a[i]<min then min:=a[i];  
        writeln;  
        write('Мин. элемент= ',min);  
end.
```

Пример 3

Дан массив целых чисел. Записать его наоборот

```
program massiv;  
uses crt;  
var a:array[1..10] of integer;{описание массива}  
    i,n:integer;  
begin  
    clrscr;  
    randomize; {генератор случайных чисел}  
    write('Введите число элементов');  
    read(n);  
    for i:=1 to n do begin  
        a[i]:=random(10); {заполнение массива}  
        write(a[i]:4); {вывод массива}  
    end;  
    writeln;{переход на след. строку}  
    for i:=n downto 1 do  
        write(a[i]:4); {вывод массива наоборот}  
end.
```


Пример 4

Определить, сколько пар одинаковых соседних элементов содержится в массиве.

```
program massiv;  
uses crt;  
var a:array[1..10] of integer;{описание массива}  
    i,n,s:integer;  
begin  
    clrscr;  
    randomize; {генератор случайных чисел}  
    write('Введите число элементов');  
    read(n);  
    for i:=1 to n do begin  
        a[i]:=random(10); {заполнение массива}  
        write(a[i]:4); {вывод массива}  
    end;  
    writeln;{переход на след. строку}  
    for i:=1 to n do  
        if a[i]=a[i+1] then s:=s+1;  
        write('s=',s); {вывод массива наоборот}  
end.
```

Пример 5

В одномерном массиве заменить все нулевые элементы на 10

```
program massiv;  
uses crt;  
var a:array[1..10] of integer;{описание массива}  
    i,n,s:integer;  
begin  
    clrscr;  
    randomize; {генератор случайных чисел}  
    write('Введите число элементов'); read(n);  
    for i:=1 to n do begin  
        a[i]:=random(10); {заполнение массива}  
        write(a[i]:4); {вывод массива}  
    end;    writeln;{переход на след. строку}  
    for i:=1 to n do begin  
        if a[i]=0 then a[i]:=10;  
        write(a[i]:4); {вывод массива наоборот}  
    end;  
end.
```

Самостоятельно:

1. В одномерном массиве найти разность наибольшего и наименьшего элементов.
2. Дан одномерный массив. Вывести на экран числа > 5 .
3. В одномерном массиве найти произведение ненулевых элементов.