

## Массивы.

**Массив** – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип.

Название *регулярный тип* (или ряды) массивы получили за то, что в них объединены однотипные (логически однородные) элементы, упорядоченные (урегулированные) по индексам, определяющим положение каждого элемента в массиве.

В качестве элементов массива можно использовать и любой другой ранее описанный тип, поэтому вполне правомерно существование массивов записей, массивов указателей, массивов строк и т.д. Элементами массива могут быть данные любого типа, включая структурированные. Тип элементов массива называется *базовым*. Особенностью языка Паскаль является то, что число элементов *фиксируется* при описании и в процессе выполнения программы не меняется.

Элементы, образующие массив, упорядочены таким образом, что каждому элементу соответствует совокупность *номеров* (индексов), определяющих *его местоположение в общей последовательности*. **Индекс** представляет собой выражение любого порядкового типа. Чтобы обратиться к элементу массива, надо *указать его базовое имя и индекс* (номер).

### Описание массива.

Массивы можно описать 2-мя способами: через описание типа массива и простое описание:

#### Описание через тип:

```
TYPE
  Matrix = Array [0..9] of Char;
VAR
  Mas : Matrix;
  ...
  Mas [6] := 'A' ;
  ...
```

#### Простое описание:

```
VAR
  Mas : Array [0..9] of Char;
  ...
  Mas [6] := 'A' ;
  ...
```

#### Общий вид описания массива

```
<имя_массива> : ARRAY [<список_индексов>] OF <тип>;
```

<имя\_массива> - правильный идентификатор;

ARRAY, OF - служебные слова (массив, из);

<список\_индексов> - список из одного или нескольких индексных типов, разделённых запятыми;

<тип> - любой тип Паскаля.

Если в форме описания массива задан один индекс, массив называется **одномерным**, если два индекса – **двумерным**, если  $n$  индексов –  **$n$ -мерным**. Одномерный массив соответствует понятию *линейной таблицы (вектора)*, двумерный – понятию *прямоугольной таблицы (матрицы)*.

#### Примеры описаний массивов разной размерности:

```
VAR
  Mas1 : Array [1..6] Of Char; {Одномерный массив на 6 элементов}
  Mas2 : Array [1..5, 1..5] Of Byte; {Двумерный массив 5x5 элементов}
  Mas3 : Array [1..3, 1..4, 1..2] Of String; {Трёхмерный массив}
```

В место числа, точной ссылки на элемент, можно подставить *переменную (математическое выражение)* порядкового типа, значение которой (ого) не выходит за пределы диапазона массива.

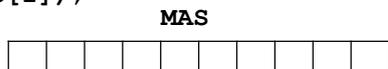
Индексированные элементы массива называются *индексированными переменными* и могут быть использованы так же, как и простые переменные. Например, они могут находиться в выражениях в качестве операндов, использоваться в операторах **FOR**, **WHILE**, **REPEAT**, **CASE**, входить в качестве параметров в операторы **READ**, **READLN**, **WRITE**, **WRITELN**; им можно присваивать любые значения, соответствующие их типу.

## Заполнение массива

Инициализация (присваивание начальных значений) массива заключается в присваивании каждому элементу массива значения, соответствующего базовому типу. Наиболее эффективно эта операция выполняется с помощью оператора **FOR**.

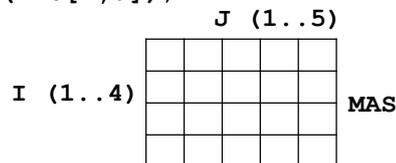
Заполнение одномерного массива:

```
FOR I:=1 TO 10 DO
  READ (MAS[I]);
```



Заполнение двумерного массива (использование вложенного цикла):

```
FOR I:=1 TO 4 DO
  FOR J:=1 TO 5 DO
    READ (MAS[I,J]);
```



I – строка;  
J – столбец.

Паскаль не имеет средств ввода-вывода элементов массива сразу, поэтому ввод и вывод значений производится поэлементно. Значения элементам массива можно присвоить с помощью *оператора присваивания*, однако чаще всего они вводятся с помощью оператора **READ** или **READLN** с использованием оператора цикла **FOR** (см. выше).

## Операции над массивами

Для работы с массивом как единым целым используется **идентификатор** массива без указания индекса в квадратных скобках. Массив может участвовать только в **операциях** отношения «равно», «не равно» и в *операторе присвоения*. Массивы, участвующие в этих действиях, должны быть **идентичны** по структуре, т.е. иметь одинаковые типы индексов и одинаковые типы компонентов. Например, если массивы описаны так

```
TYPE
  Massiv = Array [1..20] Of Real;
VAR
  A, B : Massiv;
```

то применение к ним допустимых операций даст следующий результат:

**A = B** True, если значение каждого элемента массива **A** равно соответствующему значению элемента массива **B**;  
**A <> B** True, если хотя бы одно значение элемента массива **A** не равно значению соответствующего элемента массива **B**;  
**A :=B** Все значения элементов массива **B** присваиваются соответствующим элементам массива **A**. Значения элементов массива **B** остаются неизменными.

Если массивы описаны следующим образом

```
VAR
  A : Array [1..20] Of Real;
  B : Array [1..20] Of Real;
```

то к ним **не применимы** перечисленные выше операции (*т.е. возникнет ошибка при попытке их использования*).

## Пример программы:

```
Uses CRT;
Var
  Mas : Array [1..10] Of Real;
  I : Integer;
  Sum : Real;
BEGIN
  ClrScr;
  Sum :=0;
  Writeln ('Введите 10 чисел :');
  For I:=1 To 10 Do
    Begin
      Read (Mas[I]);
      Sum :=Sum + Mas[I]
    End;
  ClrScr;
  Writeln ('Сумма 10 чисел = ', Sum:12:4);
  Writeln ('Сами числа :');
  For I:=1 To 10 Do
    Write (Mas[I]:7:2);
END.
```