

# **Структурированные типы**

# ***Массив***

- совокупность данных одного и того же типа, занимающих непрерывную область памяти. Число элементов массива фиксируется при описании типа и в процессе выполнения программы не изменяется

Формат описания массива:

имя\_массива : array [ список\_индексов ] of  
тип\_данных ;

где:

- имя\_массива — любой допустимый идентификатор;
- тип\_данных — любой тип языка.
- список\_индексов — перечисление диапазонов изменения номеров элементов массива; количество диапазонов совпадает с количеством измерений массива; диапазоны отделяются друг от друга запятой, а границы диапазона, представляющие собой интервальный тип данных, отделяют друг от друга двумя символами точки:  
[ индекс1\_нач .. индекс1\_кон , индекс2\_нач ..  
индекс2\_кон , ... ]

Например:

```
var
```

```
//Одномерный массив из 10 целых чисел.
```

```
a : array [ 1 .. 10 ] of byte ;
```

```
//Двумерный массив вещественных чисел  
(3 строки, 3 столбца).
```

```
b : array [ 1 .. 3 , 1 .. 3 ] of real ;
```

Для доступа к элементу массива достаточно указать его индекс, а если массив многомерный (например таблица), то несколько номеров:

имя\_массива[номер\_элемента]

Например: a[5], b[2, 1], M[3, 2, 4].

# **Использование языка Free Pascal для обработки массивов**

Для описания массива можно использовать предварительно определённые константы:

```
const
```

```
n=10;
```

```
m=12;
```

```
var
```

```
a : array [ 1 . . n ] of real ;
```

```
b : array [ 0 . . m ] of byte ;
```

Константы должны быть определены до использования, так как массив не может быть переменной длины!

# ***Операции над массивами***

- Для работы с массивом как с единым целым надо использовать имя массива (без указания индекса в квадратных скобках). Для доступа к элементу массива необходимо указать имя массива и в квадратных скобках индекс элемента массива, например `x[1]`, `y[5]`, `c[25]`, `A[8]`.



В языке Free Pascal определена операция присваивания для массивов, идентичных по структуре (с одинаковыми типами индексов и компонентов). Например, если массивы C и D описаны как

```
var C,D: array [ 0 .. 30 ] of real ;
```

то можно записать оператор

```
C:=D;
```

- Выполнение любой другой операции над массивом надо организовывать поэлементно, для чего необходимо организовать цикл, в котором последовательно обрабатывать элементы массива; сначала обрабатываем первый элемент массива, затем второй, третий, . . . , n-й. Для обработки элементов массива удобно использовать цикл `for..do`.

# ***Ввод-вывод элементов массива***

```
//Ввод элементов массива X с помощью цикла for.  
var  
x : array [ 1 .. 10 ] of real ;  
i , n : integer ;  
begin  
  for i :=1 to 10 do  
    begin  
      write ( ' x ( ' , i , ' ) = ' ) ; readln ( x [ i ] ) ;  
    end;  
  end.
```

# Ввести 10 чисел и найти минимальное.

```
program minim;  
uses crt;  
var  
  a:array [1..10] of integer;  
  min,i: integer;  
begin  
  clrscr;  
  for i:=1 to 10 do  
    readln(a[i]); //ВВОД ЭЛЕМЕНТОВ МАССИВА  
  min:=a[1];  
  for i:=2 to 10 do
```

# Случайные числа

# Генератор случайных чисел

- В языке программирования Паскаль для генерации псевдослучайных чисел в заданных диапазонах используется функция **random**. Перед ее использованием обычно выполняется процедура инициализации датчика случайных чисел - **randomize**; иначе программа всегда будет выдавать один и тот же результат. **Randomize** задает начальное значение последовательности, от которого вычисляются все последующие. При каждом запуске программы это значение будет разным, а значит и результат работы функции **random** будет различным.

- Функция **random** генерирует случайное число в диапазоне от 0 (включительно) до единицы.
- Если в скобках указан аргумент, то от 0 до значения указанного в скобках (не включая само значение). Так выражение **random (10)**, говорит о том, что будет получено любое число в диапазоне  $[0, 10)$ .

- Например, после выполнения кода:

```
var a:integer;
```

```
begin
```

```
a:=random (20);
```

```
writeln(a);
```

- переменная a примет случайное значение в диапазоне от 0 до 19



Если требуется получить значение в каком-либо другом диапазоне (не от нуля) используется формула:

**`random(max-min+1) + (min);`**

Таким образом, чтобы получить случайное число в диапазоне от -10 до 10 функцию random следует записать:

**`random(21)-10;`**

# Заполнение массива случайными числами от -5 до 10

```
program sl;  
var a:array [0..9] of integer;//объявляем массив  
i:integer;  
begin  
  randomize;//инициализация генератора случайных  
  чисел  
  for i:=0 to 9 do  
  begin  
    a[i]:=random(16)-5;//генерация случ. числа от -5 до 10  
    write(a[i], ' ');//вывод элемента массива на экран  
  end;  
end.
```

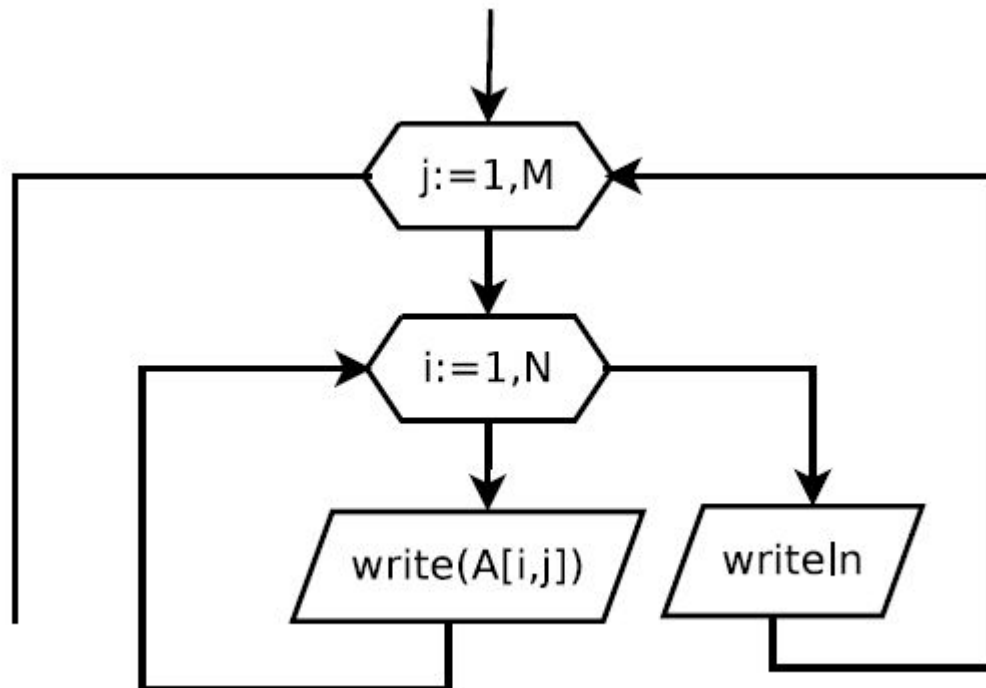
# Двумерные массивы

- В двумерном массиве элемент имеет два индекса – индекс строки и индекс столбца. Но в памяти элементы двумерного массива располагаются последовательно друг за другом.

Вывод можно осуществлять по строкам или по столбцам, но лучше, если элементы располагаются построчно, например,

```
2 3 13 35
5 26 76 37
52 61 79 17
```

Тогда в алгоритме необходимо внести следующие изменения



# Пример ввода-вывода

```
var
a : array [ 1 .. 20 , 1 .. 20 ] of real ;
i , j , n , m : integer ;
begin
{Ввод размеров массива}
writeln ( 'Введите количество строк и столбцов массива A' ) ;
readln ( N , M ) ;
{ВВОД ЭЛЕМЕНТОВ МАССИВА}
writeln ( 'Введите массив' ) ;
for i := 1 to N do
for j := 1 to m do
read ( A [ i , j ] ) ;
{ВЫВОД ЭЛЕМЕНТОВ МАССИВА}
writeln ( 'массив A ' ) ;
for i := 1 to n do
begin
for j := 1 to m do
write ( a [ i , j ] : 8 : 3 , ' ' ) ; {выводится строка}
writeln ; {переход на новую строку}
end;
```