

## Массивы

Массив – это набор элементов данных одного и того же типа, занимающих последовательные единицы памяти в некотором отрезке памяти. Каждый массив обладает следующими атрибутами:

1. Имя массива.
2. Тип массива.
3. Набор значений (элементов).
4. Число измерений.
5. Размер (протяженность) массива по каждому измерению.
6. Нижняя и верхняя границы массива по каждому измерению.

Массив объявляется в операторах описания типа, в операторах DIMENSION и COMMON. Общая форма описания массива имеет вид

*имя\_массива*(L<sub>1</sub>:U<sub>1</sub>, L<sub>2</sub>:U<sub>2</sub>, ..., L<sub>7</sub>:U<sub>7</sub>)

где L<sub>i</sub> и U<sub>i</sub> – нижняя и верхняя границы i-го измерения. В Фортране 77 максимальное число измерений равно 7. Размер i-го измерения – D<sub>i</sub> = U<sub>i</sub> – L<sub>i</sub> + 1. Размер n-мерного массива – общее число элементов в массиве – D<sub>1</sub>\*D<sub>2</sub>\*...\*D<sub>n</sub>.

Положение элемента массива в памяти определяется значением адресного выражения для этого элемента. Значение адресного выражения изменяется от 1 (первый элемент) до величины размера массива (последний элемент). Значение адресного выражения вычисляется по индексам элемента и параметрам размерностей массива. В качестве индексного выражения допускается любое правильно построенное целое выражение, не содержащее обращений к таким функциям, выполнение которых может повлиять на значения других индексных выражений в том же индексе. Число индексных выражений должно совпадать с числом измерений массива, указанным в его описателе. Каждое индексное выражение является индексом по соответствующему измерению, и его значение должно лежать в пределах границ массива по этому измерению.

### Правила вычисления адресного выражения

n	Параметры размерностей массива	Размеры элементов	Индексы	Адресное выражение
1	(L <sub>1</sub> :U <sub>1</sub> )	D <sub>1</sub> = U <sub>1</sub> – L <sub>1</sub> + 1	(K <sub>1</sub> )	1 + (K <sub>1</sub> – L <sub>1</sub> )
2	(L <sub>1</sub> :U <sub>1</sub> , L <sub>2</sub> :U <sub>2</sub> )	D <sub>1</sub> = U <sub>1</sub> – L <sub>1</sub> + 1 D <sub>2</sub> = U <sub>2</sub> – L <sub>2</sub> + 1	(K <sub>1</sub> , K <sub>2</sub> )	1 + (K <sub>1</sub> – L <sub>1</sub> ) + + (K <sub>2</sub> – L <sub>2</sub> )*D <sub>1</sub>
3	(L <sub>1</sub> :U <sub>1</sub> , L <sub>2</sub> :U <sub>2</sub> , L <sub>3</sub> :U <sub>3</sub> )	D <sub>1</sub> = U <sub>1</sub> – L <sub>1</sub> + 1 D <sub>2</sub> = U <sub>2</sub> – L <sub>2</sub> + 1 D <sub>3</sub> = U <sub>3</sub> – L <sub>3</sub> + 1	(K <sub>1</sub> , K <sub>2</sub> , K <sub>3</sub> )	1 + (K <sub>1</sub> – L <sub>1</sub> ) + + (K <sub>2</sub> – L <sub>2</sub> )*D <sub>1</sub> + + (K <sub>3</sub> – L <sub>3</sub> )*D <sub>1</sub> *D <sub>2</sub> +
...				
7	(L <sub>1</sub> :U <sub>1</sub> , ..., L <sub>7</sub> :U <sub>7</sub> )	D <sub>1</sub> = U <sub>1</sub> – L <sub>1</sub> + 1 D <sub>2</sub> = U <sub>2</sub> – L <sub>2</sub> + 1 D <sub>3</sub> = U <sub>3</sub> – L <sub>3</sub> + 1 ... D <sub>7</sub> = U <sub>7</sub> – L <sub>7</sub> + 1	(K <sub>1</sub> , ..., K <sub>7</sub> )	1 + (K <sub>1</sub> – L <sub>1</sub> ) + + (K <sub>2</sub> – L <sub>2</sub> )*D <sub>1</sub> + + (K <sub>3</sub> – L <sub>3</sub> )*D <sub>1</sub> *D <sub>2</sub> + ... + (K <sub>7</sub> – L <sub>7</sub> )*D <sub>1</sub> *D <sub>2</sub> *D <sub>3</sub> *D <sub>4</sub> *D <sub>5</sub> *D <sub>6</sub>

Если нижняя граница какого-либо измерения опущена при описании, то по умолчанию она берется равной 1.

Память для массива отводится в той программной компоненте, в которой он декларируется, если только он не описан как формальный параметр этой подпрограммы; в последнем случае массив передается подпрограмме извне в качестве аргумента. Декларацию массива в программной компоненте, в которой для него фактически будет отведена память, будем называть *декларацией фактического массива*. Декларацию массива в подпрограмме, в которую он будет передан как параметр, будем называть *декларацией формального массива*. В декларации фактического массива число его измерений и размер по каждому измерению не могут быть переменными. В декларации формального массива размер массива по каждому измерению может быть как регулируемым, так и постоянным. При вычислении значения адресного выражения верхняя граница последнего измерения не используется. Именно поэтому ее можно (настоятельно рекомендуется) заменить звездочкой (\*) в операторе описания формального массива.

Массивы Фортрана 77 располагаются в памяти, как говорят, «по столбцам». Это означает, что быстрее всего изменяется самый левый индекс.

Если в списке оператора READ (оператора PRINT) указано только имя массива, то значения элементов массива вводятся (выводятся) в том порядке, в каком элементы массива располагаются в памяти. Изменить порядок ввода-вывода элементов массива можно с помощью неявного цикла.

Указывая при обращении к подпрограмме (или подпрограмме-функции) имя фактического массива, мы тем самым передаем в подпрограмму адрес ячейки памяти, с которой начинается массив. При каждом обращении к элементу формального массива значение его адресного выражения вычисляется исходя из информации о размере массива. Адрес элемента фактического массива вычисляется по этому значению и начальному адресу фактического массива. Таким образом устанавливается соответствие между элементами формального и фактического массивов с одинаковыми значениями адресного выражения.

В Фортране 77 есть и другой способ передачи массивов в подпрограммы. Например, вместо имени массива в качестве фактического параметра можно использовать элемент массива. В этом случае начальным адресом массива будет адрес указанного элемента массива, и связь между формальным и фактическим массивами установится на основе этого адреса. Размер формального массива при этом не должен превышать числа оставшихся элементов фактического массива.

Например, в следующем наброске программы

```
DIMENSION A(100)                                SUBROUTINE SUB(X,...)
.....                                         DIMENSION X(*)
CALL SUB(A(10),...)                             .....
.....                                         END
END
```

фактическим аргументом при вызове подпрограммы SUB является не имя массива A, а имя элемента A(10). В подпрограмме используется формальный массив X, и связь между

массивами X и A устанавливается следующим образом: X(1) соответствует A(10), X(2) соответствует A(11) и т.д. Очевидно, что X(91) соответствует A(100), и поскольку верхняя граница индекса массива X определена как (\*), то X(91) – последний элемент формального массива.

Фортран 77 также допускает разные размерности у формальных и фактических массивов. Например, программа вида

```
DIMENSION A(5,10)
.....
CALL SUB(A,50,...)
.....
END

SUBROUTINE SUB(X,N,...)
DIMENSION X(*)
.....
END
```

устанавливает соответствие между X(1) и A(1,1), X(2) и A(2,1), ... , X(K) и A(I,J), где  $K=I+5*(J-1)$ . В приведенной программе фактический массив A и формальный массив X имеют одинаковое число элементов; однако такое равенство не является обязательным требованием Фортрана 77. На самом деле массив X может иметь любой размер, не превышающий размер фактического массива или размер используемой подпрограммой части массива.

### **Источники**

1. Катцан Г. Язык Фортран 77: Пер. с англ. – М.: Мир, 1982.
2. Программирование на Фортране 77: Пер. с англ./ Дж. Ашкрофт, Р. Элдридж, Р. Полсон, Г. Уилсон. – М.: Радио и связь, 1990.

## Пример

В качестве примера рассмотрим программу, которая вычисляет сумму всех элементов матрицы  $C$ , где  $C=A*B$ ,  $A$  –  $(N,M)$ -матрица,  $B$  –  $(M,K)$ -матрица. Элементы матриц  $A$  и  $B$  располагаются построчно в текстовых файлах `a.dat` и `b.dat` соответственно. Известно, что  $N*M + M*K < 1000$ .

```
DIMENSION MEM(1000)
OPEN (1, FILE='A.DAT')
READ (1,*) N,M1
OPEN (2, FILE='B.DAT')
READ (1,*) M2,K
IF(M1.NE.M2)
*   STOP 'МАТРИЦЫ ПЕРЕМНОЖАТЬ НЕЛЬЗЯ'
M=M1
CALL READAB (MEM(1), MEM(N*M+1), N, M, K)
CALL UMN (MEM(1), MEM(N*M+1), N, M, K, REZ)
PRINT *, 'РЕЗУЛЬТАТ:', REZ
STOP
END
```

```
SUBROUTINE READAB (A, B, N, M, K)
DIMENSION A(M,*), B(M,*)
READ (2,*) ((B(I,J), J=1,K), I=1,M)
CLOSE (2)
READ (1,*) ((A(J,I), J=1,M), I=1,N)
CLOSE (1)
RETURN
END
```

```
SUBROUTINE UMN (A, B, N, M, K, R)
DOUBLE PRECISION R8, RAB8
DIMENSION A(M,*), B(M,*)
R8=0D0
DO 10 I=1,N
DO 10 J=1,K
CALL SCAL (A(1,I), B(1,J), M, RAB8)
10 R8=RAB8+R8
R=R8
END
```

```
SUBROUTINE SCAL (X, Y, N, REZ8)
DOUBLE PRECISION RAB8, REZ8
DIMENSION X(*), Y(*)
RAB8=0D0
DO 1 I=1,N
RAB8=RAB8+X(I)*Y(I)
1 CONTINUE
REZ8=RAB8
END
```