

ОБРАБОТКА МАССИВОВ

Скалярный тип – простой тип данных. Скалярное данное неделимо. *Массив* – это структурированный тип данных. Массив состоит из нескольких элементов. Ко всему массиву можно обращаться по его имени. Можно обращаться к его элементу, но для этого надо задать индекс (индексы). Массивы бывают одномерные и многомерные. Для объявления массива необходимо задать типы его индексов и компонент:

ARRAY [Тип индексов] OF <Тип компонент>;

Тип компонент массива – это просто тип данных, ассоциированный с каждой компонентой массива. Тип компонент может быть любым REAL, INTEGER, CHAR, BOOLEAN, перечислимым, интервальным. В качестве компоненты массива может быть взят и тип массив.

Тип индекса должен быть одним из упорядоченных типов, т.е. любым скалярным типом, кроме REAL: INTEGER, CHAR, интервальный, перечислимый. Тип индекса определяет границы изменения индекса. Если сделана попытка использовать несуществующую компоненту, то возникает ошибка (ошибка неверного индекса).

Одномерные массивы

Одномерный массив можно задать (объявить) двумя способами:

1. С помощью служебного слова TYPE описывается тип массива, а затем с помощью VAR вводится переменная этого типа.

Общая форма записи

TYPE <тип массива> = ARRAY [тип индекса] OF <тип компонент>;
VAR <переменная>: <тип массива>;

2. С помощью слова VAR сразу описывается переменная типа массив.

Общая форма записи

VAR <переменная>: ARRAY [тип индекса] OF <тип компонент>;

Например, объявление массива из 100 элементов типа REAL можно осуществить двумя способами:

1. type R100 = array [1..100] of real;

var A: R100;

2. var A: array [1..100] of real.

Здесь задан массив с именем A, и его элементы имеют имена: A[1],...,A[100]. Чаще всего для типа индекса используют интервальный тип на основе типов INTEGER и CHAR. Однако можно в качестве индексов брать перечислимый тип.

Пр и м е р 1. Подсчет числа вхождений букв в текст определенной длины.

```
program COUNTER;
  var COUNT: array ['a'..'z'] of integer;
      CH: char; N: integer;
begin
  for CH := 'a' to 'z' do
    COUNT [CH] := 0; N := 0;
  repeat
    read (CH); N := N + 1;
    if (CH >= 'a') and (CH <= 'z') then
      COUNT [CH] := COUNT [CH] + 1;
  until CH = '.';
  for CH := 'a' to 'z' do
    writeln (CH, COUNT [CH]:5);
end.
```

Пояснение. В этом примере тип индекса есть интервальный тип на базе типа CHAR, а тип компонент есть целое число. Таким образом, элементы массива – числа, а их индексы – буквы, т.е. число элементов массива равно 26 (по числу букв латинского алфавита).

Рассмотрим теперь случай, когда тип индекса задан перечислимым типом, а компоненты массива представлены компонентами интервального типа на базе типа INTEGER.

Пр и м е р 2. Присваивание переменной с именем месяца числа дней этого месяца.

```
program NUMBRDAY;
  type MONAT = (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG,
                SEP, OKT, NOV, DEC);
  var DAY : array [MONAT] of 28..31; T : MONAT;
begin
  for T := JAN to DEC do
    case T of
      JAN, MAR, MAY, JUL, AUG, OKT, DEC: DAY [T] := 31;
      APR, JUN, SEP, NOV: DAY [T] := 30;
      FEB : DAY [T] := 28;
    end;
end.
```

Многомерные массивы

Для определения позиции элемента в двумерном массиве необходимы два индекса. Любой двумерный массив есть матрица, а матрица есть таблица. Поэтому удобно описывать двумерные массивы путем указания границ изменения индексов (номеров) строк и столбцов.

Например, таблица символов $M \times N$, где M – число строк и N – число столбцов, может быть описана:

```
var TAB : array [1..M, 1..N] of char;
```

Общая форма записи

```
VAR <имя> : ARRAY [тип индекса строки, тип индекса столбца]  
                OF <тип компонент>;
```

Однако двумерный массив можно интерпретировать как вектор-столбец, каждый элемент которого, в свою очередь, является одномерным массивом (вектор-строка). Этот подход к определению двумерного массива влечет его описание с помощью двух строк, где первая содержит описание строки, а вторая – описание столбца:

```
type LINE = array [1..N] of char;  
      STOLB = array [1..M] of LINE;  
var TAB : STOLB.
```

Здесь $TAB[I]$ – переменная типа $LINE$, а $TAB[I][J]$ – переменная типа $CHAR$.

Общая форма записи

```
TYPE <тип строки> = ARRAY [тип индекса] OF <тип компонент>;  
<тип столбца> = ARRAY [тип индекса] OF <тип строки>;  
VAR <переменная массива> : <тип столбца>;
```

Эти два вида определения массивов задают и два способа обращения к элементам массива: $TAB[I, J]$ – в первом случае и $TAB[I][J]$ – во втором. Вполне очевидно, что сказанное выше для двумерного массива распространяется и на массивы большей размерности. Например, описание $VAR CUBE : ARRAY [1..M, 1..N, 1..K] OF INTEGER$ определяет задание трехмерного массива целых чисел.

Примеры работы с массивами

Обработка массивов включает в себя, как правило, следующие компоненты: ввод массива (с клавиатуры или с помощью датчика случайных чисел), вывод полученного массива на экран и собственно его обработка. Все эти компоненты рекомендуется

оформлять в виде отдельных процедур. При этом надо учитывать следующий фактор: если процедуре (или функции) будет передаваться массив, то надо объявить в ней этот массив как параметр с атрибутом VAR даже в том случае, если значение массива внутри процедуры не изменяется. Это нужно для того, чтобы не тратить времени и памяти на размещение внутри процедуры копии массива. Заметим, что параметр обязательно должен относиться к типу, имеющему имя.

Пример 1. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить сумму положительных элементов.

Решение. При написании процедур ввода и вывода следует обратить внимание, что элементы – вещественные числа, поэтому необходимо позаботиться о верной обработке дробной части. Вычисление суммы оформим в виде функции с одним аргументом – массивом. Локальными переменными функции будут индексная переменная i и дополнительная переменная s для хранения текущей суммы элементов. В начале тела функции обязательно обнуление s . Каждый элемент массива сравним с нулем, и, если значение положительно, добавим его к искомой сумме s . В конце функции запишем значение переменной s в результирующую переменную.

```
program massiv_1;
const N=10;
type mas=array [1..N] of Real;

procedure Vvodmas(var A:mas);
  var i:Integer;
begin
  for i:=1 to N do A[i]:=-50+Random(101)+random;
end;

procedure Vivodmas(A:mas);
  var i:Integer;
begin
  for i:=1 to N do Write(A[i]:8:2);
  Writeln
end;

function Summa(A:mas):real;
  var i: Integer; s:real;
begin
  s:=0;
  for i:=1 to N do if A[i]>0 then s:=s+A[i];
  Summa:=s;
end;

var A: mas;
begin
  Randomize; Vvodmas(A);
```

```

Writeln('Исходный массив:'); Vivodmas(A);
Writeln('Ответ: ', Summa(A):0:2);
Readln
end.

```

Пример 2. В двумерном массиве, состоящем из целочисленных элементов, в каждом столбце поменять местами наибольший по модулю и последний не принадлежащий интервалу (a, b) элементы массива.

Решение. Преобразования необходимо провести в каждом столбце массива, поэтому параметр внешнего цикла в процедуре обработки - номер столбца j , а вложенного – номер строки i .

Для перестановки двух элементов в столбце массива необходимо найти номера их строк $n1$ и $n2$, а затем поменять местами значения элементов с использованием промежуточной переменной p .

Чтобы найти наибольший по модулю элемент столбца, введем дополнительную переменную max , которая будет хранить максимальное по модулю значение в текущем столбце массива на данный момент. (Можно решить задачу без использования переменной max . Подумайте, как это сделать.)

Программа должна корректно работать с любыми входными данными, а значит и в тех случаях, когда некоторые или даже все столбцы массива содержат только элементы из интервала (a, b) , и обмен значений в некоторых столбцах или во всем массиве не нужен.

```

const n=10; m=7;
type mas=array [1..n,1..m] of Integer;

procedure Vvodmas(var D:mas);
  var i,j:Integer;
begin
  for i:=1 to n do
    for j:=1 to m do
      D[i,j]:=-50+Random(101);
    end;
  end;

procedure Vivodmas(D:mas);
  var i,j:Integer;
begin
  for i:=1 to n do
    begin
      for j:=1 to m do Write(D[i,j]:4);
      Writeln;
    end;
  end;
end;

```

```

procedure Obmen(a,b: real; var D:mas);
  var i,j,p,n1,n2,max: Integer;
begin
  for j:=1 to m do
    begin
      n1:=1; max:=abs(D[1,j]);{считаем первый элемент столбца наибольшим по
модулю}
      for i:=2 to n do
        if abs(D[i,j])>max then {обнаружен больший элемент}
          begin n1:=i; max:=abs(D[i,j]) end;

      i:=n; {перебираем элементы столбца, начиная с последнего}
      while (i>=1)and (D[i,j]>a)and(D[i,j]<b) do i:=i-1;
      n2:=i;
      if n2<>0 then {если элемент, не принадлежащий интервалу (a,b), был найден}
        begin
          p:=D[n1,j]; D[n1,j]:=D[n2,j]; D[n2,j]:=p; {обмен значений}
        end;
      end;
    end;
  end;

var D: mas; a,b:Real;
begin
  Randomize; Vvodmas(D);
  Writeln('Исходный массив:'); Vivodmas(D);
  Write('Введите через пробел концы интервала (a,b): '); Readln(a,b);
  Obmen(a,b,D);
  Writeln('Ответ:'); Vivodmas(D);
  Readln
end.

```

Вопросы для самоконтроля:

1. Как описываются в языке Паскаль одномерный и двумерные массивы?
2. Может ли массив содержать разнотипные данные?
3. В каком порядке указываются индексы при обращении к элементам двумерного массива?
4. Привести пример массива, описание которого выглядит следующим образом: *var A: array [1..3, 20..24] of real.*
5. Можно ли при обработке двумерных массивов использовать однократные циклы? Если да, то приведите примеры.

6. Каким образом надо находить первый и последний элементы одномерного массива, обладающие некоторым свойством (отрицательный, наибольший, входящий в интервал и *пр.*)?

Задания

Одномерные массивы

Вариант 1

1. Ввести массив $Z[1..5]$ и число C . Для каждого элемента массива вычислить $T = \sin^2 Z_i + C$. Исходный массив и результаты распечатать.

Вариант 2

1. Ввести массив $H[1..5]$ и число A . Для каждого элемента массива

вычислить $Z = \frac{\sqrt{H_i}}{A}$. Исходный массив и результаты распечатать.

Вариант 3

1. Ввести массив $A[0..5]$ и число P . Для каждого элемента массива вычислить

$B = e^{\frac{A_i}{P}}$. Исходный массив и результаты распечатать.

Вариант 4

1. Ввести массив $T[1..4]$ и число a . Для каждого элемента массива

вычислить $f = \sqrt[3]{T_i \cdot a}$. Исходный массив и результаты распечатать.

Вариант 5

1. Ввести массив $F[0..6]$ и число C . Для каждого элемента массива вычислить

$G = \operatorname{tg} \frac{F_i}{F_i + C}$. Исходный массив и результаты распечатать.

Вариант 6

1. Ввести массивы $T[1..4]$ и $C[1..4]$. Для каждого $T[i]$, $C[i]$ вычислить $d = 3.2 \cdot T_i^3 + C_i$. Исходные массивы и результаты распечатать.

Вариант 7

1. Ввести массив $Z[0..3]$ и $A[0..3]$. Для каждого $Z[i]$ и $A[i]$ вычислить $f = \sqrt{(z_i + a_i)^3}$
.. Исходный массив и результаты распечатать.

Вариант 8

1. Ввести массив $B[1..5]$. Для каждого элемента массива вычислить $T = \sqrt[3]{3 \cdot B_i}$. Исходный массив и результаты распечатать.

Вариант 9

1. Ввести массив $L[1..4]$ и число P . Для каждого элемента массива вычислить $C = \cos^3 \frac{P}{P - L_i}$. Исходный массив и результаты распечатать.

Вариант 10

1. Ввести массив $A[1..5]$. Для каждого элемента массива вычислить $P = A_i + \sqrt{\frac{A_i}{2}}$. Исходный массив и результаты распечатать.

Вариант 11

1. Ввести массив $D[0..5]$ и число g . Для каждого элемента массива вычислить $Y = e^{D[i] - g}$. Исходный массив и результаты распечатать.

Вариант 12

1. Ввести массив $G[1..6]$ и число z . Для каждого элемента массива вычислить $v = \cos^2 \frac{G_i}{z}$. Исходный массив и результаты распечатать.

Вариант 13

1. Ввести массив $S[1..4]$ и число g . Для каждого элемента массива вычислить $H = |S_i| + 0,5$. Исходный массив и результаты распечатать.

Вариант 14

1. Ввести массив $A[1..7]$ и число S . Для каждого элемента массива вычислить $Y = S^{A_i/3}$. Исходный массив и результаты распечатать.

Вариант 15

1. Ввести массив $R[0..6]$. Для каждого элемента массива вычислить $W = -\frac{1}{3} R_i \cdot e^{24}$. Исходный массив и результаты распечатать.

Вариант 16

$$S = e^{p_4} + \frac{A}{3 - A}$$

1. Ввести массив $p_0, p_1, p_2..p_6$. Найти и напечатать результаты распечатать

Вариант 17

1. Ввести x и массив $p_0, p_1, p_2..p_6$. Для каждого p_i вычислить и напечатать $z_i = x^{p_i+1}$. Исходный массив и результаты распечатать

Вариант 18

1. Ввести число c и массив $f_0, f_1, f_2..f_6$. Для каждого f_i вычислить и напечатать $z_i = \lg \frac{f_i}{f_i + c}$. Исходный массив и результаты распечатать

Двумерные массивы

1. Дана квадратная матрица порядка N . Вычислить среднее арифметическое положительных элементов матрицы, стоящих выше главной диагонали.	2. Дана матрица размерности N на M . Найти в матрице первую по порядку строку с наибольшей суммой элементов. Вывести ее номер.
3. Дана квадратная матрица порядка N . В матрице вычислить среднее арифметическое положительных элементов, стоящих на главной диагонали.	4. Дана квадратная матрица порядка N . Вывести строку матрицы, в которой элемент, стоящий на главной диагонали, максимален.
5. Дана матрица размерности N на M . Положительные элементы матрицы переписать подряд в одномерный массив V .	6. Дана матрица размерности N на M . Вычислить количество строк матрицы, в которых есть хоть один отрицательный элемент.
7. В квадратной матрице найти сумму элементов побочной диагонали и разделить на полученную сумму все элементы последнего столбца.	8. Дана квадратная матрица порядка N . Найти произведение элементов побочной диагонали квадратной матрицы.
9. Дана матрица размерности N на M . Вывести номера всех столбцов матрицы, не содержащих отрицательных элементов.	10. Дана матрица размерности N на M . В матрице найти первый по порядку столбец с максимальной суммой элементов. Вывести его номер.
11. Дана квадратная матрица порядка N . Вывести столбец матрицы, в котором элемент, стоящий на главной диагонали, минимален, среди элементов главной диагонали.	12. Дана матрица размерности N на M . В матрице найти первый по порядку столбец с минимальной суммой модулей его элементов. Вывести его номер.
13. Найти сумму положительных элементов квадратной матрицы, находящихся ниже главной	14. Дана квадратная матрица порядка N . Найти максимальный и минимальный элементы матрицы и

диагонали.	поменять местами соответствующие им строку и столбец (строка для максимального элемента, столбец для минимального элемента).
15. Дана квадратная матрица порядка N . Найти количество четных элементов квадратной матрицы, расположенных ниже побочной диагонали.	16. Дана матрица размерности N на M . Найти произведение максимальных четных элементов столбцов матрицы