



M. C. E S C H E R.

Массивы

1

Определение типа массив (1)

(1)

Рассмотренные простые типы определяют различные множества атомарных (неразделимых) значений.

Составные типы задают множества «сложных» значений; каждое значение из такого множества образует некоторую структуру (совокупность) нескольких значений другого типа.

Мы вводим вычислительную структуру, использующую абстрактное понятие «конечная последовательность».

Каждое значение типа массив состоит из фиксированного числа элементов одного и того же базового типа (т. е. значение содержит фиксированное число однотипных компонент).

2

Определение типа массив (2)

Способ образования позволяет обозначать значения этих типов *одним групповым именем*, а доступ к отдельным элементам массивов посредством указания этого группового имени и порядкового номера (*индекса*) необходимого элемента.

Для корректного определения регулярного типа массив необходимо задать две характеристики:

- *тип элементов массива,*
- *количество и «способ нумерации» элементов.*

type

<идентификатор - тип массива>= array[T1] of T2;

T1 - тип индекса:

ограниченный,
литерный,
перечислимый (см. дальше);

T2 - тип элементов массива (любой тип)

3

Определение типа массив (3)

Примеры:

```
const n=20;
type
  vector=array[1..100] of real;
  mas=array[-5..6] of integer;
  matrix=array[1..n] of array[1..n] of integer;
var
  v: vector;
  m :mas;
  arr1, arr2 : matrix;
```

В том случае, когда определен тип массива, элементами которого являются снова массивы, можно использовать эквивалентное определение *двухмерного* массива.

Следующие два определения эквивалентны:

```
v:array[1..10] of array[1..20] of integer;
v:array[1..10,1..20] of integer;
```

Число индексов (*размерность массива*) не ограничена.

4

Определение типа массив (4)

Единственное возможное действие над массивом в целом это присваивание:

v1:=v2;

Типы обоих массивов в данном случае должны совпадать.

Доступ к элементам массива:

<идентификатор массива>[<индекс>]

или

<идентификатор массива>[<список индексов через запятую>]

В качестве индексов произвольные выражения, тип которых должен совпадать с типом индекса.

Примеры:

v[1]

v[(i+2)*6]

Пусть

v2 : array[1..10] of array[5..20] of integer

– двумерный массив, тогда **v2[k]** - *k*-ый массив в группе из 10 массивов целых чисел,

v2[k][5] – 5-ый элемент этого массива, тот же элемент получаем и так **v2[k,5]**.

5

Определение типа массив (5)

Элемент массива считается **переменной**:

он может получать значения (например, в операторе присваивания);

он может участвовать в выражениях там, где может присутствовать переменная данного типа.

Ассортимент операций над элементами массива полностью определяется типом этих элементов.

Примеры:

v2[i,j]:=v2[i,j-1]+1;

m[k]:=m[k-1] mod 5;

Ошибки в работе с массивами:

Выход индекса за допустимые пределы

var

v:array[0..10] of real;

....

v[11]:=0.5; <==== транслятор обнаружит ошибку

i:=11;

v[i]:=0.5; <==== эта ошибка может быть обнаружена только во время исполнения и то, если осуществляется контроль диапазона, иначе это может привести к непредсказуемым последствиям

6

Программы с одномерными массивами

(1)

1. Даны n чисел a_1, a_2, \dots, a_n . Найти наименьшее из них.

```
const
  n=5;
var
  a: array[1..n] of real;
  i: integer;
  min: real;

begin
  writeln('введите ',n,' чисел');
  for i:=1 to n do
    read(a[i]);
  min:=a[1];
  for i:=1 to n do
    if a[i]<min then min:=a[i];
  writeln('наименьшее число = ',min)
end.
```

7

Программы с одномерными массивами

(2)

Прежняя задача, но другое описание массива.

```
var
  a: array[1..100] of real;
  n,i: integer;
  min: real;

begin
  writeln('Введите количество чисел');
  readln(n);
  writeln('введите ',n,' чисел');
  ...
end.
```

8

Программы с одномерными массивами (3)

2. Даны n чисел a_1, a_2, \dots, a_n . Найти первое отрицательное из них.

```
var
  a: array[1..100] of real;
  n,i: integer;
  x: real;
  p:boolean;
begin
  writeln('Введите количество чисел');
  readln(n);
  writeln('введите ',n,' чисел');
  for i:=1 to n do
    read(a[i]);
  p:=false; {p=false – значит отрицательное число еще не найдено}
  for i:=1 to n do
    if (a[i]<0) and not p then begin x:=a[i]; p:=true end;
  if p then writeln('первое отрицательное число = ',x)
  else writeln('отрицательных чисел нет')
end.
```

9

Программы с одномерными массивами (4)

3. Даны n чисел a_1, a_2, \dots, a_n . Написать фрагмент программы для вычисления $y=a_1-a_2+a_3-\dots-a_{n-1}+a_n$.

Первый вариант

```
y:=0;
for i:=1 to n do
  if i mod 2 = 0 then y:=y-a[i] else y:=y+a[i];
```

Второй вариант

```
y:=0; p:=true;
for i:=1 to n do
  begin
    if p then y:=y+a[i] else y:=y-a[i];
    p:= not p
  end;
```

Третий вариант

```
y:=0; z:=1;
for i:=1 to n do
  begin
    y:=y+z*a[i];
    z:= -z;
  end;
```

10

Программы с одномерными массивами

(5)

4. Даны n чисел a_1, a_2, \dots, a_n . Написать фрагмент программы для вычисления $y = a_1 a_n + a_2 a_{n-1} + \dots + a_n a_1$.

```
y:=0;
for i:=1 to n do
  y:=y+a[i]*a[n+1-i];
```

5. Даны n чисел a_1, a_2, \dots, a_n . Написать фрагмент программы для определения количества пар одинаковых соседних элементов.

```
k:=0;
for i:=1 to n-1 do
  if a[i]=a[i+1] then k:=k+1;
```

6. Даны n целых чисел a_1, a_2, \dots, a_n . Всегда ли после четного числа следует нечетное? Написать фрагмент программы.

```
p:=true; {если p=true, то ответ - всегда}
for i:=1 to n-1 do
  if (a[i] mod 2 = 0) and (a[i+1] mod 2 = 0) then p:=false;
```

11

Программы с матрицами

(1)

Программа 1
Ввести матрицу построчно
Возвести каждый элемент в квадрат
Напечатать полученную матрицу построчно

```
const
  n=4; m=3;
var
  a: array[1..n,1..m] of integer;
  i,j: integer;
begin
  {Ввод матрицы}
  writeln('введите матрицу ',n,' - строк', m,' - столбцов');
  for i:=1 to n do
    begin
      for j:=1 to m do
        read(a[i,j]); {Ввод потоком}
      writeln
    end;
```

12

Программы с матрицами (2)

```
Продолжение программы 1
{возводим элементы в квадрат}
for i:=1 to n do
  for j:=1 to m do
    a[i,j]:=a[i,j]*a[i,j];

{печать матрицы}
writeln('полученная матрица:');
for i:=1 to n do
  begin
    for j:=1 to m do
      write(a[i,j]:4);
    writeln
  end;
end.
```

13

Программы с матрицами (3)

```
Работа программы 1

введите матрицу 4 - строк 3 - столбцов
1 2 3
4 5 6
7 8 9
0 8 3

полученная матрица :
  1   4   9
 16  25  36
 49  64  81
  0  64   9
```

14

Программы с матрицами (4)

2. Задача. Вычислить максимальный элемент матрицы и его индексы.
(фрагменты программы: без ввода и вывода)

```
const n=4;m=5;
type matrix=array[1..n,1..m] of real;
var a:matrix;
    max:real; in,jm:integer;
    i,j:integer;
.....
in:=1;jm:=1;max:=a[1,1];
for i:=1 to n do
for j:=1 to m do
    if a[i,j]>max then
        begin
            max:=a[i,j];
            in:=i;
            jm:=j;
        end;
```

15

Программы с матрицами (5)

3. Задача. В квадратной матрице вычислить суммы элементов на главной и побочной диагоналях.
(фрагменты программы: без ввода и вывода)

```
const n=4;
type matrix=array[1..n,1..n] of real;
var a:matrix;
    s1,s2:real; i:integer;
.....
s1:=0; s2:=0;
for i:=1 to n do
    begin
        s1:=s1+a[i,i];
        s2:=s2+a[i,n+1-i]
    end;
```

```
1 2 3 7
4 5 6 8
7 8 9 9
0 8 3 5
```

```
{s1= a[1,1]+a[2,2]+a[3,3]+a[4,4]}
{s2= a[1,4]+a[2,3]+a[3,2]+a[4,1]}
```

16

Программы с матрицами (6)

4. Задача. В квадратной матрице вычислить суммы элементов, лежащих не ниже главной диагонали. Местоположение этих элементов помечено *.

```
* * * *
0 * * *
0 0 * *
0 0 0 *
```

(фрагменты программы: без ввода и вывода)

```
const n=4;
type matrix=array[1..n,1..n] of real;
var a:matrix; s:real; i,j:integer;
.....
s:=0;
for i:=1 to n do
for j:=1 to n do
if j>=i then s:=s+a[i,j];
```

17

Программы с матрицами (7)

5. Задача. Создать матрицу с элементами $a[i,j]=i+j$, не используя ввод.

```
const n=4; m=5;
type matrix=array[1..n,1..m] of integer;
var a:matrix; i,j:integer;
```

```
.....
for i:=1 to n do
for j:=1 to m do
a[i,j]:=i+j;
```

6. Задача. Создать матрицу с элементами со случайными целыми элементами в диапазоне от 0 до 10 включительно.

```
const n=4; m=5;
type matrix=array[1..n,1..m] of integer;
var a:matrix; i,j:integer;
```

```
.....
randomize;
for i:=1 to n do
for j:=1 to m do
a[i,j]:=random(11);
```

18