

§6. Ввод и вывод элементов массива

Для работы с массивом необходимо присвоить значения его элементам. Сделать это можно несколькими способами.

Значения элементов, которые не изменяются при работе программы (констант), можно задавать в разделе описаний, например, массив из 8 простых чисел: *const A: array[1..8] of integer = (2, 3, 5, 7, 11, 13, 17, 19);*

Для присвоения значения элементу указывают имя массива и индекс этого элемента, например, *A[4]:=7;*

Для работы с несколькими элементами используются циклические алгоритмы. В тех случаях, когда заранее известно количество элементов, используют цикл *for*. Важно помнить, что значение параметра (счетчик) цикла *i* не может выходить за границы заданного в описании массива диапазона.

Для ввода значений с клавиатуры используются **стандартные** процедуры *read* или *readln*. Например, следующая программа присваивает значения, вводимые с клавиатуры, пяти элементам массива.

```
var A: array[1..5] of integer ;
    i: integer;
begin
    for i := 1 to 5 do read(A[i]) ;           { ввод значений элементов }
end.
```

При использовании процедуры *read(A[i])* в строке ввода можно набрать значения нескольких или сразу всех элементов массива, отделяя их друг от друга пробелами, и нажать **Enter**. При этом каждый набор отображается в окне вывода в одну строку (рис.2.2 а). Если в строке ввода набрать больше пяти чисел, то будут присвоены только первые пять значений. Если набрать меньше пяти чисел, то программа будет ожидать продолжения ввода.

При использовании процедуры *readln(A[i])* значения вводят по одному и после каждого нажимают **Enter**. При этом они отображаются в окне вывода в столбец (рис.2.2 б).

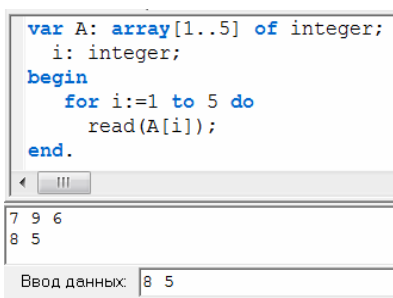
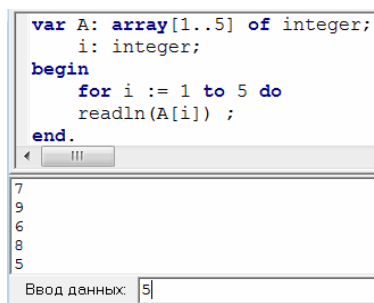


Рис. 2.2 а)



б)

Для вывода используют процедуры *write* или *writeln*. Процедура *write* выводит значения элементов массива в строку. При этом выводимые значения необходимо отделять пробелами или иными символами (например, запятой, точкой с запятой), иначе все они будут напечатаны слитно. Для вывода значений элементов в столбец используют процедуру *writeln*. Для вывода значений элементов в обратном порядке используют цикл *for ... downto...*

Во многих случаях удобно использовать форматированный вывод. Напомним формат вывода: для целых чисел *X:m*, а для вещественных - *X:m:n*, где *m* - ширина поля вывода, а *n* - количество **знаков** после десятичной точки.

Пусть, например, массив описан следующим образом:

```
const A: array[1..5] of integer = (2, 3, 5, 7, 9);
```

Приведем некоторые примеры применения процедур *write* и *writeln*.

Команды	Вывод
<i>for i := 1 to 5 do write (A[i]);</i>	23579
<i>for i := 1 to 5 do write (A[i], ' ');</i>	2, 3, 5, 7, 9,
<i>for i := 2 to 4 do write (A[i], ' ');</i>	3; 5; 7;
<i>for i := 5 downto 1 do write (A[i], ' ');</i>	9; 7; 5; 3; 2;
<i>for i := 1 to 4 do writeln(A[i]);</i>	2 3 5 7
<i>for i := 1 to 5 do write (A[i]:6);</i>	$\underbrace{2}_{6} \underbrace{3}_{6} \underbrace{5}_{6} \underbrace{7}_{6} \underbrace{9}_{6}$ позиций

Для удобства при вводе/выводе значений элементов массива можно использовать текстовые подсказки и указывать индекс, например: ‘введите значение *i*-го элемента’ или ‘*i*-й результат равен’.

Нередко значения элементов массивов не вводятся с клавиатуры, а вычисляются с помощью арифметических выражений, например, массив из 9 нечетных чисел можно сформировать так: *for i := 1 to 9 do A[i]:= 2*i - 1;*

Вычисление значений функции на заданном отрезке с определенным шагом называется *табулированием* функции. В 8 классе с этой целью Вы использовали циклы. При этом значения функций не сохранялись – они вновь вычислялись на каждом шаге цикла.

Для хранения таблиц значений функций используют массивы. Аргументами в простейших случаях могут служить индексы элементов (либо арифметические выражения от индексов).

Пример 1. Составить программу, которая осуществляет табулирование функции $y = x^2$, сохранение результатов в массиве и вывод значения элементов массива в строку.

Аргумент изменяется от -6 до 6 с шагом 1.

```

program Primer6_1;
var y: array[-6..6] of integer;
    i: integer;
begin
    for i := -6 to 6 do y[i] := sqr(i);      { формирование массива }
    for i := -6 to 6 do write( y[i] :4);   { форматированный вывод в строку}
end.
    
```

Результат работы программы будет выглядеть так:

36	25	16	9	4	1	0	1	4	9	16	25	36
----	----	----	---	---	---	---	---	---	---	----	----	----

При решении многих практических задач в науке и технике, экономике и производстве используются числа, которые задаются случайным образом. Часто их применяют при тестировании и отладке программ, чтобы не набирать исходные данные вручную, например, значения элементов массива.

Случайные значения элементов массива задаются с помощью стандартной функции *random*.

Функция *random* без аргумента генерирует случайные вещественные числа на промежутке [0,1).

Если случайные вещественные числа должны принадлежать иному промежутку, например, [3, 4), то значение элемента задается выражением $X[i] := random + 3$; (рис 2.3).

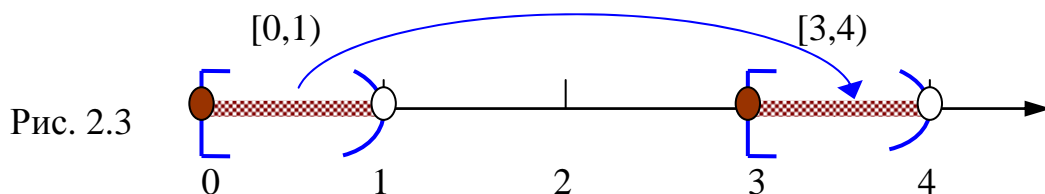


Рис. 2.3

Пример 2. Составить программу, которая формирует массив из семи случайных вещественных чисел, принадлежащих промежутку [3; 4). и выводит пять первых чисел массива с двумя десятичными цифрами в столбец шириной 8 знаков.

```

program Primer6_2;
var X: array[1..7] of real;
    i: integer;
begin
    for I :=1 to 7 do
        X[i] := random+3;
    for i:=1 to 5 do
        writeln( X[i] :8:2);
end.

```

{ формирование массива }

{ вывод значений элементов с номерами от 1 до 5 }
{ в столбец шириной 8 знаков, два десятичных }

3.46
3.59
3.83
3.39
3.27

Результат работы программы может выглядеть так:

Целые случайные числа из промежутка **[0; n]** генерирует функция *random(n+1)*. Если целочисленные значения элементов массива должны принадлежать промежутку [a; b], то их вычисляют с помощью выражения:

$$X[i] := \text{random}(b - a + 1) + a;$$

Например, массив из 10 случайных целых чисел, принадлежащих промежутку [20; 50], можно сформировать так :

```

for i := 1 to 10 do X[i] := random(31) + 20;

```

{ 50 - 20 + 1 = 31 }

Элементы массива в программах могут использоваться так же, как и обычные переменные. Покажем это на примере вывода графики.

Напомним, что рисование производится в графическом окне и требует подключения модуля *GraphAbc*.

Пример 3. Составить программу, рисующую 9 концентрических окружностей, радиусы которых хранятся в массиве случайных целых чисел, принадлежащих промежутку [20; 200].

```

program Primer6_3;
uses GraphAbc;
var R: array[1..9] of integer;
    i: integer;
begin
    for i:=1 to 9 do
        R[i]:=random(181) + 20;
    setBrushStyle(bsClear);
    setPenColor(clRed);
    for i:=1 to 9 do
        circle(240, 240, R[i] );
end.

```

{ подключение графического модуля }

{ формирование массива из 9 }

{ случайных целых чисел от 20 до 200 }

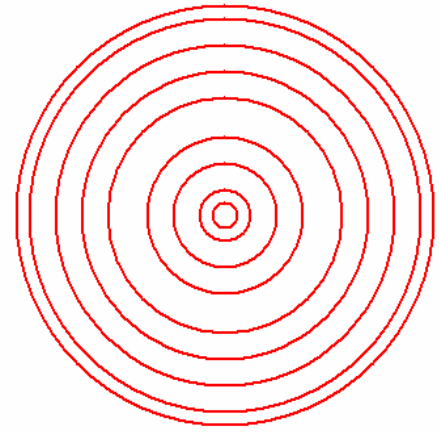
{ задание прозрачного фона }

{ задание цвета пера }

{ рисование окружности }

Результат работы

может выглядеть так (рис. 2.4):



1. Какие процедуры используются для ввода значений элементов массива?
- ? 2. Какие процедуры используются для вывода значений элементов массива?
3. Какая функция используется для формирования массива случайных чисел?

Упражнения

1. Заполните пропуски в программе ввода и вывода 10 целочисленных элементов массива, заданных случайным образом на промежутке [20; 40]:

```
var A: array[ 1..___ ] of integer;
    i: integer;
begin
    for i:=1 to ___ do
        begin
            A[i] := random(____) + ____;
            write(____);
        end;
    end.
```

2. Составьте программу для ввода в массив 7 целых чисел с клавиатуры и вывода этих чисел в обратном порядке.
3. Составьте программу для ввода и вывода температуры за неделю

1	сентября	температура	была	17
2	сентября	температура	была	15
3	сентября	температура	была	16
4	сентября	температура	была	19

Результат может быть таким

4. Составьте программу, которая формирует массив из девяти положительных четных чисел и выводит 5 первых чисел одной строкой.
5. Составьте программу, которая формирует массив из 12 случайных целых чисел, принадлежащих промежутку [-5; 5], и выводит эти числа в столбец.
6. Составьте программу, которая формирует массив из 10 случайных вещественных чисел, принадлежащих промежутку [4; 5), и выводит числа с 3-го по 7-е в столбец с двумя десятичными цифрами.