

# Организация ввода-вывода в Pascal

**Ввод данных** – передача информации от внешнего носителя в оперативную память для обработки.

**Вывод данных** – обратный процесс, когда данные передаются после обработки из оперативной памяти на внешнее устройство.

Для выполнения операций ввода-вывода служат четыре процедуры:

1. read,
2. readln,
3. write,
4. writeln.

# Процедура чтения (*read, readln*)

**Назначение:** процедура чтения считывает значения, вводимые с клавиатуры, и записывает их в переменные, которые указаны.

Формат:

*read* (<имя переменной>, ..., <имя переменной>);

## ***Порядок выполнения:***

1. Программа приостанавливает свою работу до тех пор, пока не будут введены значения для указанных в списке переменных.
2. Значения вводятся с клавиатуры через пробел.
3. После набора данных нажимается клавиша ввода <Enter>. При этом каждой переменной будет присвоены соответствующие данные в порядке их ввода.

**Пример:**

**var** p: real;

k, h: integer;

**begin**

read (p, k);

read (h);

.....

**end.**

*Строка ввода: 23.5 21 45*

При выполнении программы переменные получат значения:  $p = 23.5$ ,  $k = 21$ ,  $h = 45$

Процедура *readln* аналогична процедуре *read*.

Отличие в том, что после считывания последнего в списке значения для одной процедуры *readln* данные для следующей процедуры *readln* будут считываться с начала новой строки.

Запись *Readln* без переменных даёт задержку на экране, выход – Enter.

***Пример:***

```
var p: real;  
h, k : integer;  
begin  
readln (p, k);  
readln (h);  
.....  
end.
```

*Строка ввода: 23.5 21*

# Процедура вывода ( *write, writeln* )

**Назначение:** процедура вывода производит вывод числовых данных, символов, строк и логических значений на экран.

Формат: *write* (<список вывода>);

Элементы списка вывода разделяются запятыми.

Элементом может быть:

1. текст, заключенный в апострофы; в него могут входить любые символы, в том числе и русские буквы. При этом текст будет напечатан на экране точно в таком же виде, в каком указан в операторе.

**Например:** в программе - Write ('Привет!');  
на экране - Привет!

2. имя переменной - на экране будет напечатано ее значение.

**Например:** если  $X=675$ , то в программе - Write (X);  
на экране - 675

3. выражение - будет вычислено его значение и напечатан результат.

**Например:** в программе - Write ( $34*2+102$ );  
на экране - 170

Процедура *writeln* аналогична *write*.

Отличие в том, что *write* после вывода последнего в списке значения оставляет курсор на той же строчке, а *writeln* переводит его после печати на новую строку.

Процедура *writeln*, записанная без параметров, вызывает перевод строки.

## *Пример:*

```
var A, B, C: integer;  
begin  
  Readln (A, B, C);  
  Write ('A*B =',A*B);  
  Write ('C*B =',C*B);  
end.
```

На экране:

3 4 5 {Enter}.

A\*B = 12

C\*B =20

# Формат печати

## 1. Печать целых чисел.

Формат печати целых чисел:

**Writeln ( <имя>: <колич. позиций>);**

**Пример:** в программе: `Writeln (A:4, B:6);`

на экране:        `_ _45_ _ _123`

## 2. Печать дробных чисел.

Формат печати дробного числа:

**Writeln ( <имя>: <колич. всех знаков>: <колич. знаков на дробную часть>);**

*Пример:* пусть  $X=1243,563$ , тогда

Writeln (X:8:2); ==> \_1234.56

Writeln (X:13:5); ==> \_ \_ \_1234.56300

Writeln (X:10:3); ==> \_ \_1234.563