

# Организация ввода и вывода данных

Урок №2

Для вывода данных из оперативной памяти на экран монитора используется

```
write (<выражение 1> ,< выражение 2> , ...,< выражение N>)
```

список вывода

**Пример: Оператор write ('s=',s) выполняется так:**

- на экран выводятся символы, заключённые в апострофы: s=
- на экран выводится значение переменной, хранящееся в ячейке оперативной памяти с именем s.

Если значение переменной s равно 15 и она имеет целочисленный тип, то на экране появится: s=15.

При выполнении оператора вывода все элементы списка вывода печатаются непосредственно друг за другом. Так, в результате работы оператора `write (1, 20, 300)` на экран будет выведена последовательность цифр 120300, которая будет восприниматься нами как число 120300, а не как три отдельные числовые константы. Сделать выводимые данные более доступными для восприятия можно разными способами:

Вариант организации вывода	Оператор вывода	Результат
Без разделителей	<code>write(1, 20, 300).</code>	120300
Добавить разделители – запятые	<code>write(1, ',', 20, ',', 300)</code>	1, 20, 300
Добавить разделители – пробелы	<code>write(1, ' ', 2, ' ', 3)</code>	1 20 300
Указать формат вывода	<code>write(1:3, 20:4, 300:5)</code>	1 20 300

Формат вывода — это указываемое после двоеточия целое число, определяющее, сколько позиций на экране должна занимать выводимая величина.

Если цифр в числе меньше, чем зарезервированных под него позиций на экране, то свободные позиции дополняются пробелами слева от числа. Если указанное в формате вывода после двоеточия число меньше, чем необходимо, то оно автоматически будет увеличено до минимально необходимого.

Для вывода вещественного числа в формате с фиксированной запятой в списке вывода для каждого выражения указываются два параметра:

- общее количество позиций, отводимых под число;
- количество позиций в дробной части числа:

Оператор вывода	Результат выполнения оператора
<code>write ('s=', s:2:0);</code>	<code>s=15</code>
<code>write ('s=', s:3:1);</code>	<code>s= 15.0</code>
<code>write ('s=', s:5:1);</code>	<code>s= 15.0</code>

При выполнении нового оператора `write` вывод продолжается в той же строке. Чтобы осуществить переход к новой строке, используется оператор `writeln`. Других различий между операторами `write` и `writeln` нет.



# Составим первую программу

- Составим окружность
- Исходные данные: радиус:  $r$
- Результат: величина  $s$  круга.  $s, S$
- Исходные данные: соотношение математики
- Программная формула

```

program n_1;
  const pi=3.14;
  var r, c, s: real;
begin
  r:=5.4;
  c:=2*pi*r;
  s:=pi*r*r;
  writeln ('c=', c :6:4);
  writeln ('s=', s:6:4)
end.

```

# ИИУ

ГСЯ

ИТЬ

адъ

НЫ

pca

ИМ

**read** ( <имя переменной1>, ..., <имя переменной N> )

список ввода

При выполнении оператора `read` компьютер переходит в режим ожидания данных: пользователь должен ввести данные с клавиатуры и нажать клавишу Enter. Несколько значений переменных числовых типов могут вводиться через пробел или через запятую. При вводе символьных переменных пробел и запятая воспринимаются как символы, поэтому ставить их нельзя. Первое введенное пользователем значение переменной помещается в ячейку памяти, имя которой расположено первым в списке ввода, и т.д. Поэтому типы вводимых значений (входного потока) должны соответствовать типам переменных, указанных в разделе описания переменных.

# Задачи на разбор

**Задача № 3. Вывести на экран сообщение «Hello World!»**

Формулировка. Вывести на экран сообщение «Hello World!».

**Задача № 4. Посчитать площадь и периметр прямоугольника.**

Формулировка. Ввести два числа и найти их удвоенную сумму и произведение.

**Задача № 5. Посчитать выражение**

**$S := b(ac + 10) / (bc + a)$ , где  $a := b + 2$ ; b, c - переменные**

Формулировка. Ввести два числа и найти значение a, вычислить выражение.