

# Общие сведения о языке программирования **Pascal.**

Урок изучения нового материала

Внимательно изучаем новый материал.

# Повторение пройденного материала

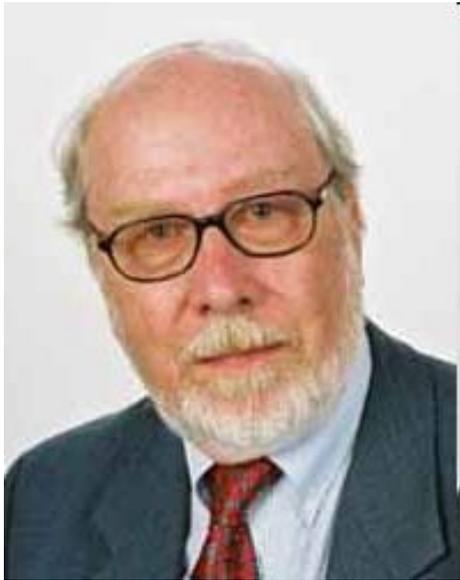
**Устно ответить на вопросы:**

- 1) Какие алгоритмы называются линейными?**
- 2) Какие алгоритмы называются разветвляющимися?**
- 3) Какие виды разветвляющихся алгоритмов вы знаете? Чем они отличаются?**
- 4) Какие алгоритмы называются циклическими?**
- 5) Что называют телом цикла?**
- 6) Какие типы циклов вы знаете? Охарактеризуйте каждый из них.**

# Изучение нового материала

**Языки программирования** - это формальные языки, предназначенные для записи алгоритмов, исполнителем которых будет компьютер.

Записи алгоритмов на языках программирования называются **программами**.



**Никлаус Вирт** (1934 года рождения) - швейцарский учёный в 1968 – 1969 гг. , специалист в области информатики, один из известнейших теоретиков в области разработки языков программирования, профессор информатики (компьютерных наук). **Разработчик языка Паскаль и ряда других языков программирования.**

# Язык Паскаль

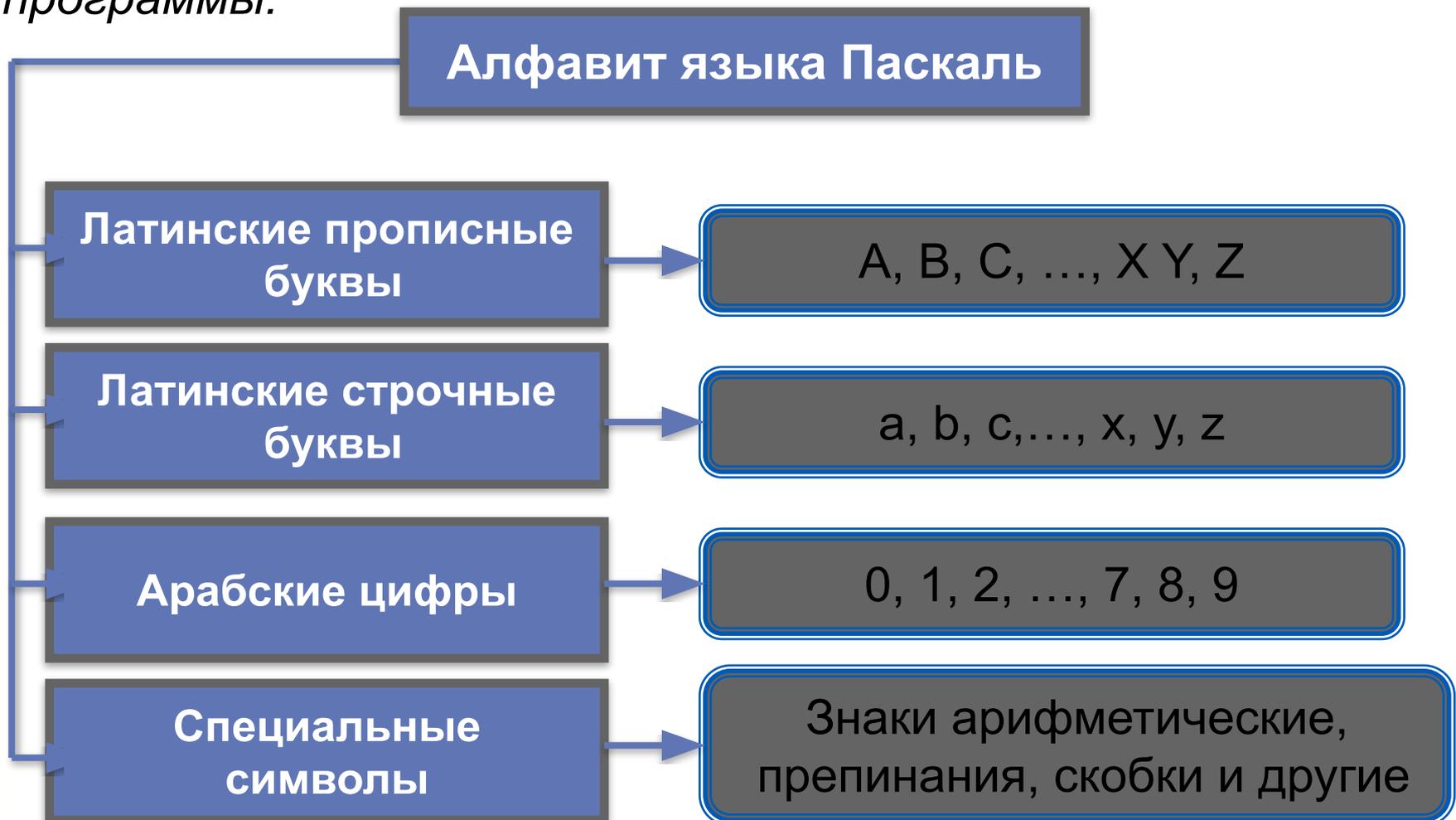
*Язык Паскаль* – универсальный язык программирования. Он может применяться для записи алгоритмов решения самых разных задач: вычислительных, обработки текста, построения графических изображений...



Своё название этот язык получил в честь французского ученого **Блеза Паскаля**, известного не только своими достижениями в математике, физике и философии, но и созданием первой в мире механической машины, выполнявшей сложение двух чисел.

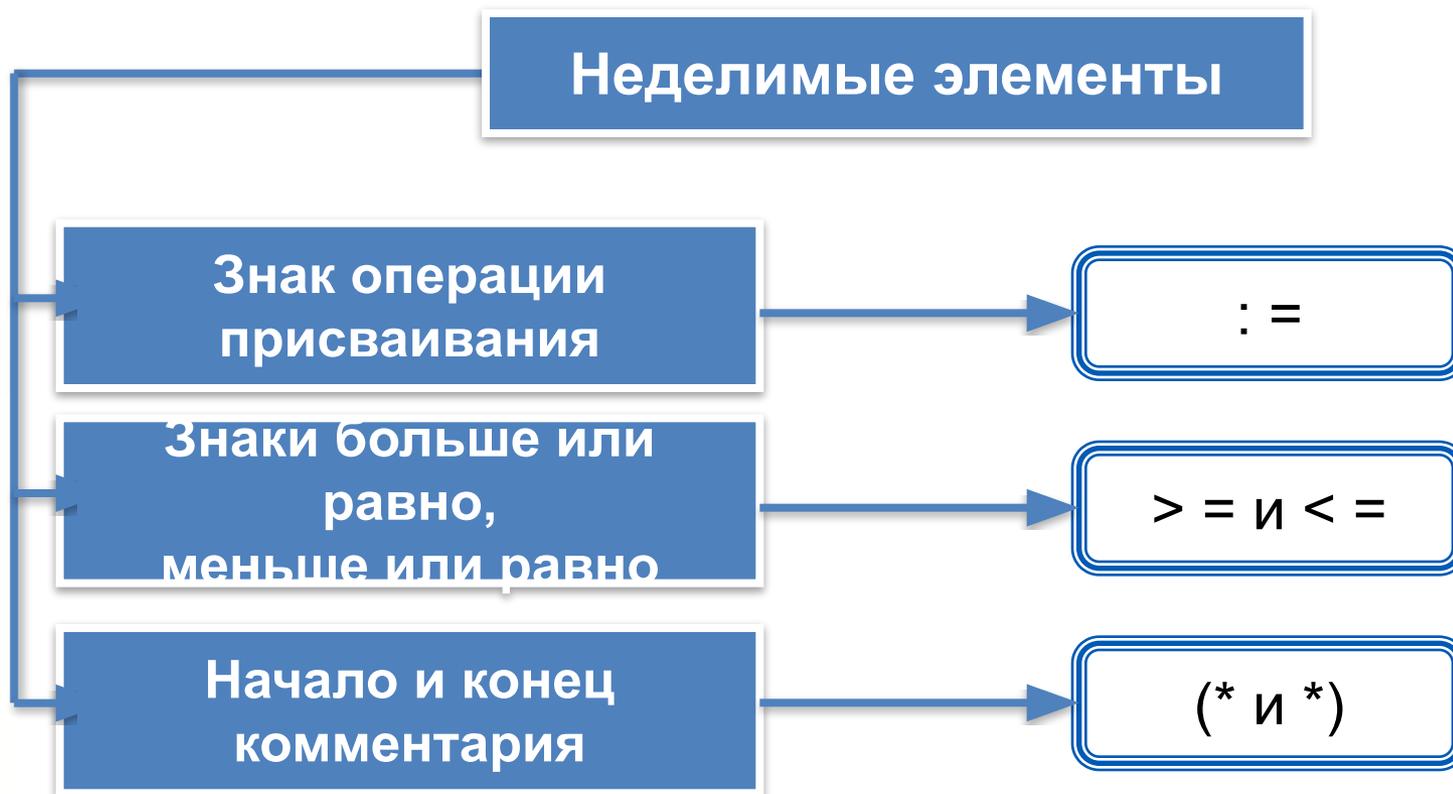
# Алфавит языка

*Язык Паскаль* содержит процедурный стиль программирования, то есть программа представляет собой последовательность операндов, задающих то или иное действие. *Символы, которые можно использовать для записи программы:*



# Алфавит языка

В алфавит языка Паскаль включены неделимые элементы (составные символы).



# Словарь языка

Служебное слово языка Паскаль	Значение служебного слова
<b>and</b>	и
<b>array</b>	массив
<b>begin</b>	начало
<b>do</b>	выполнить
<b>else</b>	иначе
<b>for</b>	для
<b>if</b>	если
<b>of</b>	из
<b>or</b>	или
<b>procedure</b>	процедура
<b>program</b>	программа
<b>repeat</b>	повторять
<b>then</b>	то
<b>to</b>	до (увеличивая до)
<b>until</b>	до (до тех пор, пока)
<b>var</b>	переменная
<b>while</b>	пока

# Алфавит и словарь языка

**Имена** (констант, переменных, программ и других объектов) - любые отличные от служебных слов последовательности букв, цифр и символа подчеркивания, начинающиеся с буквы или символа подчеркивания.

## Правильные имена

x  
velichina  
zzz  
polnaja\_summa  
tri\_plus\_dva  
s25  
\_k1  
a1b88qq  
oshibka



## Неправильные имена

**Ж** - буква не латинского алфавита  
**polnaja summa** - содержится символ (пробел), не являющийся буквой, цифрой или знаком подчеркивания.

**2as** - начинается с цифры

**Domby&Son** - содержится символ &, не являющийся буквой, цифрой или знаком подчеркивания

Прописные и строчные буквы в именах не различаются.  
Длина имени может быть любой.

# Простые типы данных

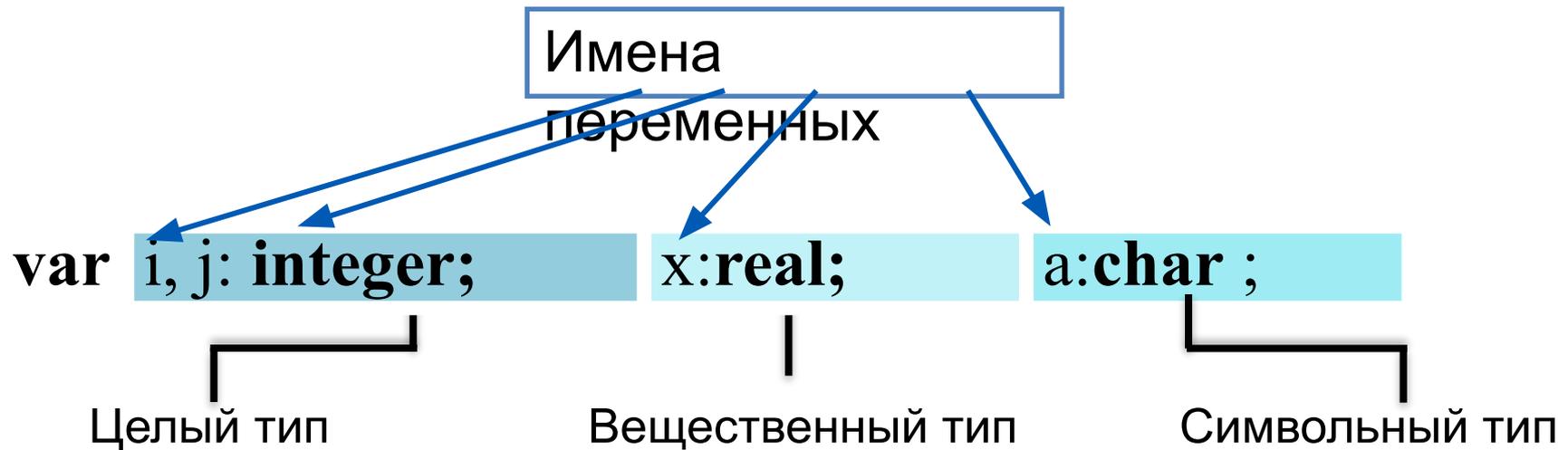
В языке Pascal используются различные типы данных. Мы будем пользоваться только некоторыми из них – простыми типами данных.

Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	integer	- 32 768 ... 32 767	2 байта со знаком
Вещественный	real	$\pm(2.9 * 10^{-39} \dots 1.7 * 10^{+38})$	6 байтов
Символьный	char	Произвольный символ алфавита	1 байт
Строковый	string	Последовательность символов длиной меньше 255	1 байт на символ
Логический	boolean	True и False	1 байт
Целочисленный	Byte	0...255	1 байт

# Структура программы на языке Паскаль



# Пример раздела описания переменных:



# Общий вид программы

```
program <имя программы>;  
  const <список постоянных значений>;  
  var <описание используемых переменных>;  
begin <начало программного блока>  
  <оператор 1>;  
  <оператор 2>;  
  . . .  
  <оператор n>  
end.
```

**Операторы** - языковые конструкции для записи действия, выполняемого над данными в процессе решения задачи.

Точка с запятой служит разделителем между операторами, но не является окончанием соответствующего оператора.

# Оператор присваивания

Основное преобразование данных, выполняемое компьютером, - присваивание переменной нового значения, что означает изменение содержимого области памяти.

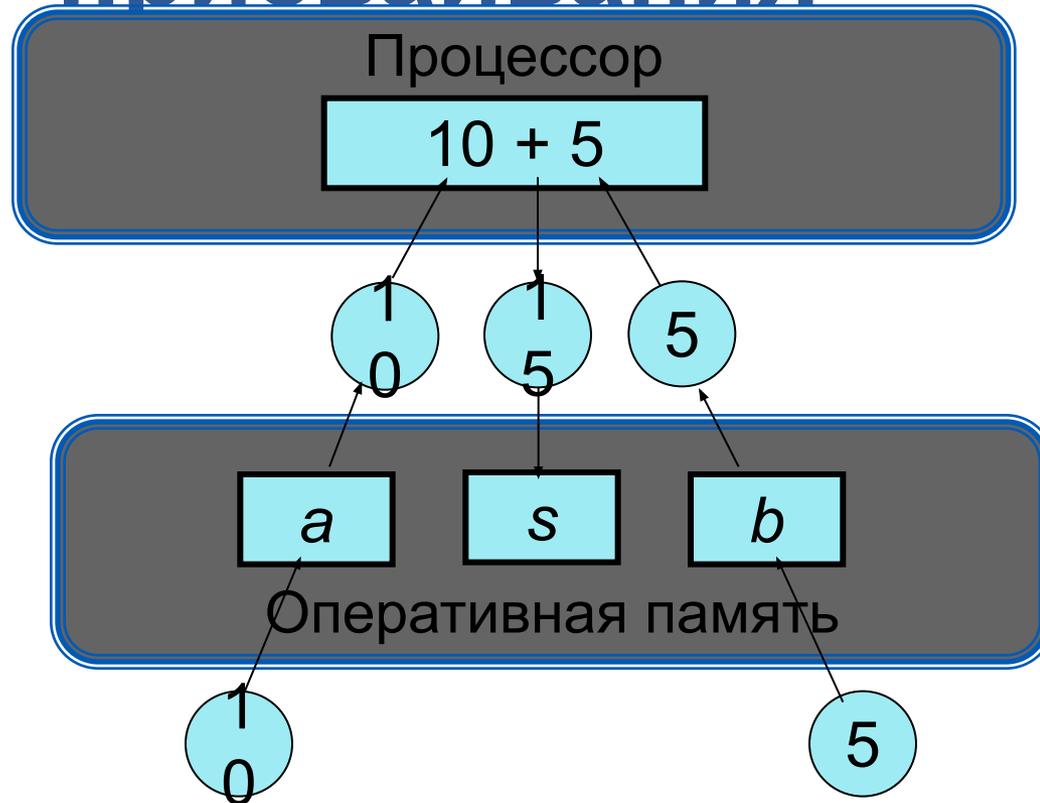
Общий вид оператора:

**<имя переменной>:=<выражение>**

Операция присваивания допустима для всех приведённых в таблице выше типов данных. Выражения в языке Pascal конструируются по рассмотренным ранее правилам для алгоритмического языка.

# Выполнение оператора присваивания

`a:=10;`  
`b:=5;`  
`s:=a+b`



При выполнении оператора `a:=10` в ячейку оперативной памяти компьютера с именем `a` заносится значение 10; при выполнении оператора `b:=5` в ячейку оперативной памяти компьютера с именем `b` заносится значение 5. При выполнении оператора `s:=a+b` значения ячеек оперативной памяти с именами `a` и `b` переносятся в процессор, где над ними выполняется операция сложения. Полученный результат заносится в ячейку оперативной памяти с именем `s`.

# Вывод данных

Всё, что мы с вами посмотрели нам поможет в записи программы преобразования данных. Но чтобы вывести преобразованные данные из оперативной памяти на экран монитора, используется оператор вывода **write**.

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

СПИСОК ВЫВОДА

**Выражения** - символьные, числовые, логические, в том числе переменные и константы

**Пример:** `write ('s=', s).`



На экран выводятся значения переменной, хранящейся в ячейке оперативной памяти с именем s

Информация в кавычках выводится на экран без изменений

# Варианты организации вывода

Вариант организации вывода	Оператор вывода	Результат
Без разделителей	<code>write (1, 20, 300).</code>	120300
Разделители – запятые	<code>write (1, ',', 20, ',', 300)</code>	1, 20, 300
Разделители – пробелы	<code>write (1, ' ', 2, ' ', 3)</code>	1 20 300

# Формат вывода

**Формат вывода** позволяет установить количество позиций на экране, занимаемых выводимой величиной.

**write (s:x:y)**

**x** - общее количество позиций, отводимых под число;

**y** - количество позиций в дробной части числа.

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

**writeln** - вывод с новой строки!

# Первая программа

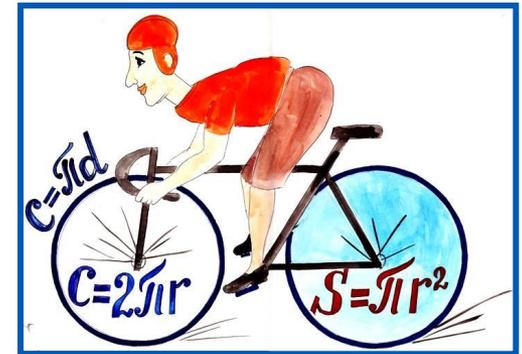
Составим программу, вычисляющую длину окружности и площадь круга радиуса 5,4 см.

Исходным данным в этой задаче является радиус  $r = 5,4$  см. Результатом работы должны быть величины:  $c$  – длина окружности и  $s$  – площадь круга.  $s$ ,  $c$  и  $r$  – величины вещественного типа.

Используем формулы:  $c=2\pi r$   $s=\pi r^2$

```
program C_and_S;
  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.
```

Проверьте работу  
этой программы в  
Pascal

A screenshot of the Pascal ABC IDE. The window title is "Pascal ABC". The menu bar includes "Файл", "Правка", "Вид", "Программа", and "Сервис". The toolbar contains icons for file operations and editing. The main window shows a program named "Program1.pas" with the following code:

```
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)
```

The output window at the bottom shows the results:

```
c =33.9120
s=91.5624
```

# Ввод данных с клавиатуры

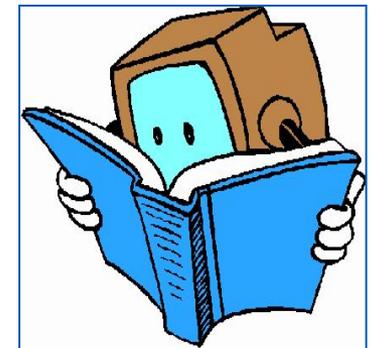
Ввод в оперативную память значений переменных :

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

список ввода

Выполнение оператора **read**:

- 1) компьютер переходит в режим ожидания данных:
- 2) пользователь вводит данные с клавиатуры:
  - несколько значений переменных числовых типов могут вводиться через пробел или через запятую;
  - при вводе символьных переменных пробел и запятую ставить нельзя;
- 3) пользователь нажимает клавишу **Enter**.



# Ввод данных с клавиатуры

**!** *Типы вводимых значений* должны *соответствовать типам переменных*, указанных в разделе описания переменных.

```
var i, j: integer; x: real; a: char;  
read (i, j, x, a);
```

Варианты организации входного потока:

```
1 0 2.5 A<Enter> 1,0 <Enter> 1<Enter>  
2.5, A<Enter> 0<Enter>  
2.5<Enter>  
A<Enter>
```

После выполнения оператора **readln** курсор переходит на новую строку.

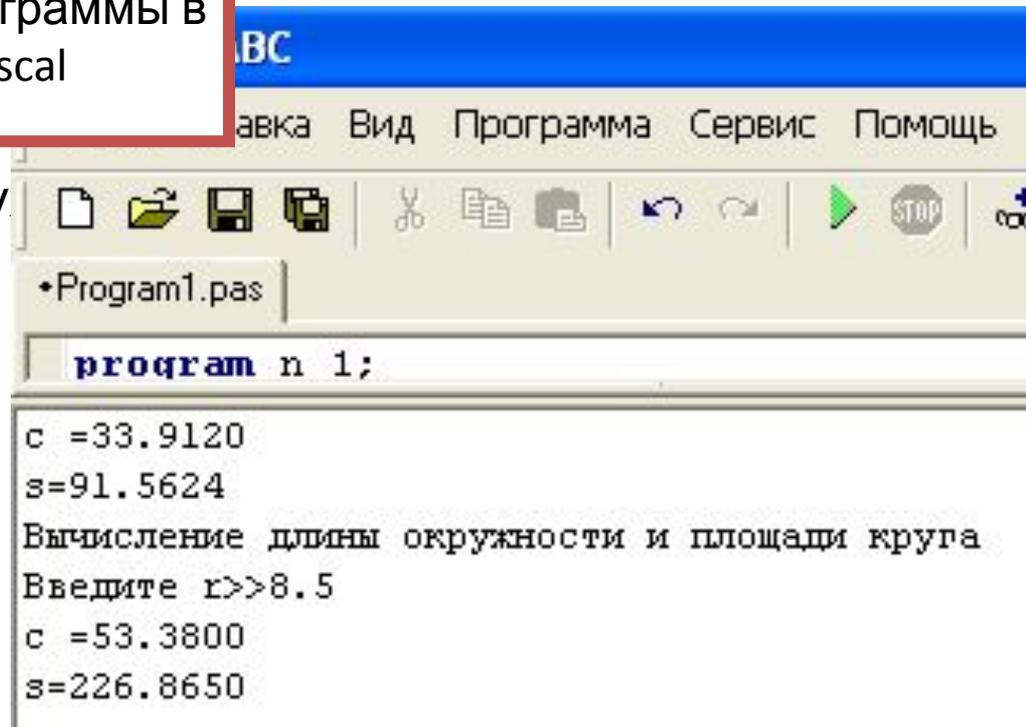
# Улучшенная программа

Улучшим предыдущую составленную программу, организовав в ней ввод данных с помощью оператора **read**.

Всегда будем выводим на экран текстовое сообщение с помощью оператора **writeln**, описывающее то или иное действие, чтобы пользователю было понятно, какое действие сейчас происходит.

```
program C_and_S_2;  
  const pi=3.14;  
  var r, c, s: real;  
begin  
  writeln('Вычисление длины окру  
  write('Введите r>>');  
  readln(r);  
  c:=2*pi*r;  
  s:=pi*r*r;  
  writeln ('c = ', c:6:4);  
  writeln ('s = ', s:7:4)  
end.
```

Проверьте работу  
этой программы в  
Pascal



```
program n 1;  
с =33.9120  
s=91.5624  
Вычисление длины окружности и площади круга  
Введите r>>8.5  
с =53.3800  
s=226.8650
```



## Первая программа.pas

```
program C_and_S;  
  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.
```

**После написания  
программы нужно  
нажать кнопку  
"Выполнить"**

## Окно вывода

**Результат выполнения программы появится здесь.**

**Если вы допустили ошибку в написании программы, то здесь выйдут сведения об этой ошибке, тогда нужно будет её исправить и снова "Выполнить воспроизведение программы".**

# Закрепление изученного материала

1) Прежде всего, все установите на ПК, за которым вы занимаетесь программу PascalABC - в свободном доступе её можно скачать из интернета, например с сайта:

<https://pascal-abc.ru.net/>

2) Откройте в данной программе файл «Первая программа», который я сбросила - в нём написана программа вычисления длины окружности и площади круга со слайда 18; проверьте правильность её работы.

3) Улучшите эту программу, переписав алгоритм программы со слайда 21, проверьте правильность её работы.

4) Попробуйте сами написать **программу для выведения на экран нечетного числа из двух чисел с разной четностью** и воспроизвести в PascalABC;

## **5) Выполните письменно задания:**

- Выписать основные моменты из презентации, которые вам в дальнейшем пригодятся и запомнить их (структуру программы на языке Pascal, служебные слова, типы данных, способы организации ввода и вывода данных);
- Выполнить задания:

# Закрепление изученного материала

**Задание 1:** Определите значения переменных после выполнения фрагмента программы.

а)

```
x:=11;
```

```
y:=5;
```

```
z:=y;
```

```
y:=x mod y;
```

```
x:=z;
```

```
y:=(y+2)*z;
```

<i>x</i>	<i>y</i>	<i>z</i>

б)

```
x:=13;
```

```
y:=3;
```

```
z:=x;
```

```
z:=z div y;
```

```
y:=x;
```

<i>x</i>	<i>y</i>	<i>z</i>

# Закрепление изученного материала

**Задание 2:** Запишите раздел описания переменных и операторы, необходимые для вычисления:

а) значения функции  $y = \sqrt{x}$ :

-----  
-----  
-----

б) гипотенузы прямоугольного треугольника по двум известным катетам:

-----  
-----  
-----

**Например:** ... корней квадратного уравнения:

```
var a, b, c, d: integer; x1, x2: real;
```

```
d:=b^2-4*a*c;
```

```
x1:=(-b+sqrt(d))/2*a;
```

```
x2:=(-b-sqrt(d))/2*a;
```

# Закрепление изученного материала

**Задание 3:** Найдите ошибку в программе.

```
program a1;  
  var c: integer;  
begin  
  c:=4.75;  
  writeln ('c=', c)  
end.
```

Опишите её...

Ответ: .....

**Задание 4:** Установите соответствие между операторами и результатами их выполнения, если значение переменной равно 5.

1) write(a)

2) write('a')

3) write('a=', a)

A) a

Б) a = 5

В) 5

# Закрепление изученного материала

## Задание 5:

Дан фрагмент программы:

```
read (a); read (b); c:=a+b; write (a, b); write (c)
```

Упростите его, сократив число операторов ввода и вывода.

## Задание 6:

Дан фрагмент программы:

```
a:=10; b:=a+1; a:=b-a; write (a, b)
```

Какие числа будут выведены на экран компьютера?

Письменную работу и файл с написанной программой в Pascal (задание №4 с 22-го слайда) отправить мне в сетевом городе личным сообщением до следующего занятия, либо ссылку на эту работу в облаке.