



Pascal

# Pascal

Паскаль был разработан швейцарским ученым Никлаусом Виртом. Паскаль считается важнейшим инструментом для обучения методам структурного программирования и с 1983 г. введен в учебные курсы в школах для учащихся, которые специализируются в области информатики.

В дальнейшем язык Паскаль совершенствовался и приобрел новые свойства, отличные от авторского варианта.

- ❖ Язык Паскаль относительно прост в изучении, довольно ясен и логичен и, будучи первым изучаемым языком программирования, приучает к хорошему стилю.

- *Как и естественные языки, каждый язык программирования имеет свой стиль и свои правила.*
- **Синтаксис** языка программирования – это набор правил, которые определяют способы построения правильных программ из символов алфавита.

# Алфавит языка Паскаль

**26 латинских строчных и 26 латинских прописных букв:**

**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

**a b c d e f g h i j k l m n o p q r s t u v w x y z**

**подчеркивание \_**

**10 цифр:**

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

**знаки операций:**

**+ - \* / = <> < > <= >= := @**

**ограничители (разделители):**

**. , ' ( ) [ ] ( . . ) { } (\* \*) .. : ;**

**спецификаторы:**

**^ # \$**

# Структура программы

*{1. заголовок программы}*

**program**      Имя\_Программы;

*{2. раздел указания используемых модулей}*

**uses**      Список\_Используемых\_Модулей;

*{3. Раздел описаний}*

**label**      Описания\_меток;

**const**      Описания\_Констант;

**type**      Описания\_Типов;

**var**      Описания\_Переменных;

**procedure**      Описания\_Процедур\_и\_функций;

**function**

**exports**      Описания\_Экспортируемых\_Имен;

*{4. Раздел операторов}*

**begin**

    Операторы

**end.**

# Упрощенная структура программы

*{1. заголовок программы}*

**program** Имя\_Программы;

*{2. раздел указания используемых модулей}*

**uses** Список\_Используемых\_Модулей;

*{3. Раздел описаний}*

**const** Описания\_Констант;

**var** Описания\_Переменных;

*{4. Раздел операторов}*

**begin**

Операторы программы

**end.**

- Команды языка программирования называются операторами
- Разделителем операторов в Паскале является `;` (точка с запятой)



***Комментарий*** представляет собой фрагмент текста программы, ограниченный символами **{}**.  
Комментарии в программе выполняют информационную функцию.

***{Моя первая программа }***

```
program first;  
begin  
    writeln('Hello, World!')  
end.
```

## Пример: периметр прямоугольника

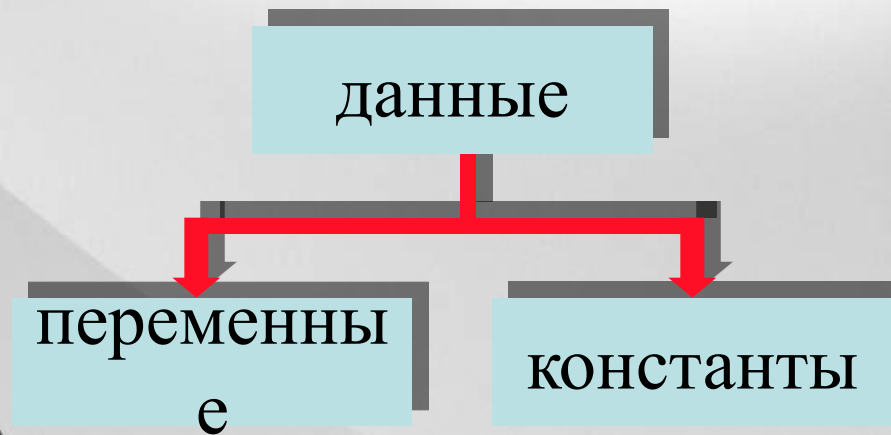
```
program perimetr;           {заголовок программы}
uses crt;                 {crt – необходим для очистки экрана}
var a,b:integer;         {объявление переменных}
    P:integer;
Begin                     {начало программы}
    clrscr;                {очистка экрана}
    a:=12;                 {присваиваем переменной a значение 12}
    b:=7;                  {присваиваем переменной b значение 7}
    P:=2*(a+b);           {значение выражения присваиваем P}
    write('P = ',P);      {выводим на экран значение P}
end.                     {конец программы}
```

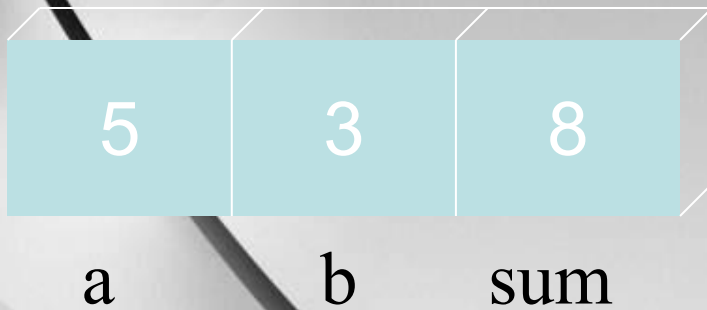
## ***Программа в своей работе имеет дело с данными.***

Некоторые данные устанавливаются еще до того, как программа начнет выполняться, а после ее запуска сохраняют свои значения неизменными на всем протяжении работы программы. Это ***константы***.

Другие данные могут изменяться во время выполнения программы. Они называются ***переменными***.

- ✓ Различие между *переменной* и *константой* довольно очевидно: во время выполнения программы значение переменной может быть изменено, а значение константы нет.





Под переменной мы будем понимать ячейку («коробку»), куда компьютер может записывать («складывать») данные.

Под ячейкой мы в действительности подразумеваем «кусочек памяти» в котором хранится информация.

Чтобы воспользоваться информацией, хранящейся в ячейке, нужно, чтобы каждая ячейка имела свое **имя** или, как часто говорят, - **идентификатор**.

# ***Идентификаторы.***

- ✓ ***Именами*** (*идентификаторами*) называют элементы языка - константы, метки, типы, переменные, процедуры, функции, модули, объекты.
- ✓ **Идентификатором** является последовательность букв, цифр и знаков подчеркивания, которая начинается с буквы или символа подчеркивания и не содержит пробелов.

- ✓ Имя может содержать произвольное количество символов, но значащими являются 63 символа.
- ✓ Не разрешается в языке ПАСКАЛЬ использовать в качестве имен **служебные слова** и **стандартные имена**, которыми названы стандартные константы, типы, процедуры, функции и файлы.
- ✓ Примеры имен языка ПАСКАЛЬ:
  - A   b12   r1m   SIGMA   gamma   I80\_86

В Паскале разница между строчными и прописными буквами игнорируется, поэтому имена **NaMe** и **name** одинаковы.



- ***Служебное слово*** – это слово, которое в языке ПАСКАЛЬ имеет определенное смысловое значение, которое не может быть изменено. Иногда его называют **ключевым словом**.

## **Служебные (зарезервированные) слова:**

<b>ABSOLUTE</b>	<b>EXPORTS</b>	<b>LIBRARY</b>	<b>SET</b>
<b>ASSEMBLER</b>	<b>EXTERNAL</b>	<b>MOD</b>	<b>SHL</b>
<b>AND</b>	<b>FAR</b>	<b>NAME</b>	<b>SHR</b>
<b>ARRAY</b>	<b>FILE</b>	<b>NIL</b>	<b>STRING</b>
<b>ASM</b>	<b>FOR</b>	<b>NEAR</b>	<b>THEN</b>
<b>ASSEMBLER</b>	<b>FORWARD</b>	<b>NOT</b>	<b>TO</b>
<b>BEGIN</b>	<b>FUNCTION</b>	<b>OBJECT</b>	<b>TYPE</b>
<b>CASE</b>	<b>GOTO</b>	<b>OF</b>	<b>UNIT</b>
<b>CONST</b>	<b>IF</b>	<b>OR</b>	<b>UNTIL</b>
<b>CONSTRUCTOR</b>	<b>IMPLEMENTATION</b>	<b>PACKED</b>	<b>USES</b>
<b>DESTRUCTOR</b>	<b>IN</b>	<b>PRIVATE</b>	<b>VAR</b>
<b>DIV</b>	<b>INDEX</b>	<b>PROCEDURE</b>	<b>VIRTUAL</b>
<b>DO</b>	<b>INHERITED</b>	<b>PROGRAM</b>	<b>WHILE</b>
<b>DOWNTO</b>	<b>INLINE</b>	<b>PUBLIC</b>	<b>WITH</b>
<b>ELSE</b>	<b>INTERFACE</b>	<b>RECORD</b>	<b>XOR</b>
<b>END</b>	<b>INTERRUPT</b>	<b>REPEAT</b>	
<b>EXPORT</b>	<b>LABEL</b>	<b>RESIDENT</b>	

***Пробелы*** нельзя использовать внутри  
сдвоенных символов и зарезервированных  
слов.

***Существуют имена которые  
называются стандартными.***

sin cos real true

В отличие от служебных слов смысл  
стандартных имен ***может быть  
переопределен*** программистом.

# Объявление переменных

- ✓ **Это указание компилятору, сколько памяти необходимо зарезервировать для переменных нашей программы.**
- ✓ **В откомпилированной программе для всех переменных отведено место в памяти, и всем переменным присвоены нулевые значения.**
- *Все переменные, используемые в программе необходимо объявить в разделе описания переменных после зарезервированного слова **var**.*

Данные бывают различных  
ТИПОВ...

# *Типы данных*

- Имена *стандартных* типов являются predetermined идентификаторами и действуют в любой точке программы. Они описаны в стандартном модуле System, который по умолчанию подключается в список используемых модулей.
- *Пользовательские* типы – это дополнительные типы (простые и структурированные) описанные пользователем.

# ***Стандартные типы данных***

1. группа целых типов (Shortint, Integer, Longint, Byte, Word);
2. группа действительных типов (Single, Real, Double, Extended, Comp);
3. логические (булевские типы) (Boolean, ByteBool, WordBool, LongBool);
4. символьный (Char);
5. строковый (String);
6. указательный (Pointer);
7. текстовый тип (Text).

# Группа целых типов

Название типа	Тип	диапазон значений	требуемая память
Короткое целое со знаком	Shortint	-128 .. 127	1 байт
Целое со знаком	<b>Integer</b>	-32768 .. 32767	2 байта
Длинное целое со знаком	Longint	-2147483648 .. 2147483647	4 байта
Короткое целое без знака	Byte	0 .. 255	1 байт
Целое без знака	Word	0 .. 65535	2 байта



## *Объявление переменных целого типа*

```
Var b : byte;  
    summa, count : integer;
```

## *Испозование*

```
summa:= -365;
```

**Числа** в языке ПАСКАЛЬ обычно записываются в десятичной системе счисления.

Положительный знак числа может быть опущен.

**Целые** числа записываются в форме без десятичной точки, например:

**217   -45   8954   +483**

# ***Группа вещественных типов***

определяет те данные, которые реализуются подмножеством действительных чисел.

```
Var A: real;
```

```
begin
```

```
  ...
```

```
  A:=0.65;
```

```
  ...
```

```
End.
```

Название типа	Тип	Диапазон значений	Количество цифр мантиссы	Размер (байт)
Вещественное число одинарной точности	<b>Real</b>	$2.9e^{-39} .. 1.7e^{+38}$	11	6
Вещественное число	<b>Single</b>	$1.5e^{-45} .. 3.4e^{+38}$	7	4
Вещественное число двойной точности	<b>Double</b>	$5.0e^{-324} .. 1.7e^{+308}$	15	8
Вещественное число повышенной точности	<b>Extended</b>	$3.4e^{-4932} .. 1.1e^{+4932}$	19	10
Целое число в формате вещественного	<b>Comp</b>	$-9.2e^{+18} .. 9.2e^{+18}$	19	8

**Действительные числа** записываются в форме с десятичной **точкой** или в форме с использованием десятичного порядка, который изображается буквой **E**:

**28.6      0.65      -0.018      4.0**

**5E12      -1.72E9      73.1E-16**

**Булевскому** типу данных соответствует идентификатор **Boolean**. Переменные булевского типа имеют размер **1 байт** и могут содержать значения **TRUE** или **FALSE**.

Значению FALSE соответствует 0, любое число отличное от нуля считается TRUE.

```
Var кнопка, flag : boolean;
```

```
Begin
```

```
    кнопка:=true;
```

***Символьному*** типу соответствует стандартный идентификатор **Char**.

Переменные и константы символьного типа могут принимать значения из множества символов кода ASCII.

Объявление символьных переменных

```
Var simvol, bukva, z : char;
```

**Строковому** типу соответствует стандартный идентификатор **String**.

Var

**S : String;**

{строка от 0 до 255 символов}

**S2: String[5];**

{стока из 5-ти символов}



***Строка символов*** представляет собой последовательность символов из набора символов кода ASCII, заключенную в одиночные кавычки.

*Строки* в языке ПАСКАЛЬ - это последовательность символов, записанная между апострофами. Если в строке в качестве содержательного символа необходимо употребить сам апостроф, то следует записать два апострофа.

Примеры строк:

'СТРОКА' 'STRING' 'АД"ЮТАНТ'

- ✓ Символьный тип, а также целые и булевские типы относят к, так называемым, **порядковым** типам.
- ✓ Множество допустимых значений любого порядкового типа представляет собой упорядоченную последовательность, каждый элемент которой имеет свой порядковый номер (начиная с 0).

# Функция Ord

возвращает порядковый номер этого значения в описании типа.

`Ord(2)=2, Ord('0')=48`

```
WriteLn(ord('e'));
```

```
WriteLn(ord('9'));
```

# *Таблица кодировки ASCII*

- **ASCII** (*American Standard Code for Information Interchange*; произносится «áски») — компьютерная кодировка для представления латинского алфавита, арабских цифр, некоторых знаков пунктуации и управляющих символов.
- Ее ввел американский институт стандартизации ANSI.

<b>Код</b>	<b>Символ</b>	<b>Код</b>	<b>Символ</b>	<b>Код</b>	<b>Символ</b>	<b>Код</b>	<b>Символ</b>
<b>32</b>	<b>пробел</b>	<b>56</b>	<b>8</b>	<b>80</b>	<b>P</b>	<b>104</b>	<b>h</b>
<b>33</b>	<b>!</b>	<b>57</b>	<b>9</b>	<b>81</b>	<b>Q</b>	<b>105</b>	<b>i</b>
<b>34</b>	<b>"</b>	<b>58</b>	<b>:</b>	<b>82</b>	<b>R</b>	<b>106</b>	<b>j</b>
<b>35</b>	<b>#</b>	<b>59</b>	<b>;</b>	<b>83</b>	<b>S</b>	<b>107</b>	<b>k</b>
<b>36</b>	<b>\$</b>	<b>60</b>	<b>&lt;</b>	<b>84</b>	<b>T</b>	<b>108</b>	<b>l</b>
<b>37</b>	<b>%</b>	<b>61</b>	<b>=</b>	<b>85</b>	<b>U</b>	<b>109</b>	<b>m</b>
<b>38</b>	<b>&amp;</b>	<b>62</b>	<b>&gt;</b>	<b>86</b>	<b>V</b>	<b>110</b>	<b>n</b>
<b>39</b>	<b>'</b>	<b>63</b>	<b>?</b>	<b>87</b>	<b>W</b>	<b>111</b>	<b>o</b>
<b>40</b>	<b>(</b>	<b>64</b>	<b>@</b>	<b>88</b>	<b>X</b>	<b>112</b>	<b>p</b>
<b>41</b>	<b>)</b>	<b>65</b>	<b>A</b>	<b>89</b>	<b>Y</b>	<b>113</b>	<b>q</b>
<b>42</b>	<b>*</b>	<b>66</b>	<b>B</b>	<b>90</b>	<b>Z</b>	<b>114</b>	<b>r</b>

<b>43</b>	<b>+</b>	<b>67</b>	<b>C</b>	<b>91</b>	<b>[</b>	<b>115</b>	<b>s</b>
<b>44</b>	<b>,</b>	<b>68</b>	<b>D</b>	<b>92</b>	<b>\</b>	<b>116</b>	<b>t</b>
<b>45</b>	<b>-</b>	<b>69</b>	<b>E</b>	<b>93</b>	<b>]</b>	<b>117</b>	<b>u</b>
<b>46</b>	<b>.</b>	<b>70</b>	<b>F</b>	<b>94</b>	<b>^</b>	<b>118</b>	<b>v</b>
<b>47</b>	<b>/</b>	<b>71</b>	<b>G</b>	<b>95</b>	<b>_</b>	<b>119</b>	<b>w</b>
<b>48</b>	<b>0</b>	<b>72</b>	<b>H</b>	<b>96</b>	<b>`</b>	<b>120</b>	<b>x</b>
<b>49</b>	<b>1</b>	<b>73</b>	<b>I</b>	<b>97</b>	<b>a</b>	<b>121</b>	<b>y</b>
<b>50</b>	<b>2</b>	<b>74</b>	<b>J</b>	<b>98</b>	<b>b</b>	<b>122</b>	<b>z</b>
<b>51</b>	<b>3</b>	<b>75</b>	<b>K</b>	<b>99</b>	<b>c</b>	<b>123</b>	<b>{</b>
<b>52</b>	<b>4</b>	<b>76</b>	<b>L</b>	<b>100</b>	<b>d</b>	<b>124</b>	<b> </b>
<b>53</b>	<b>5</b>	<b>77</b>	<b>M</b>	<b>101</b>	<b>e</b>	<b>125</b>	<b>}</b>
<b>54</b>	<b>6</b>	<b>78</b>	<b>N</b>	<b>102</b>	<b>f</b>	<b>126</b>	<b>~</b>
<b>55</b>	<b>7</b>	<b>79</b>	<b>O</b>	<b>103</b>	<b>g</b>	<b>127</b>	

# ***Стандартный ввод и вывод***

- ✓ Осуществляется встроенными процедурами
- ✓ ***Read***(список переменных); – чтение значений, введенных с клавиатуры
- ✓ ***Readln***(список переменных); - ln в конце означает переход на новую строку
- ✓ ***Write***(список переменных); - вывод на экран
- ✓ ***Writeln***(список переменных);

# Пример

```
Program primer;  
Uses crt;  
Var a,b,summa:integer;  
Begin  
  clrscr;  
  write('Введите 2 целых числа: ');  
  readln(a,b);  
  Summa:=a+b;  
  write('Сумма = ',summa);  
End.
```



- ✓ Вводить можно переменные *целых, вещественных, символьного и строкового* типов.
- ✓ Допускается вывод значений *целых, вещественных, символьного, строкового и булевских* типов.

# Операции и выражения

## Выражение

- в программировании служит для определения действий.
- Выражения состоят из операций и операндов.
- По количеству операндов выражения делятся на унарные (один операнд с символом операции:  $-5$ ,  $-(-6)$ ,  $\text{not False}$ ) и бинарные (два операнда, между которыми ставится символ операции:  $5+7$ ,  $(4-2)*5+10$ , *True or False* - истина или ложь).

## *Классификация операций*

1. арифметические операции:
  - унарные: +, -
  - бинарные: +, -, \*, /, div, mod
2. операции отношения: =, <>, <, >, <=, >=
3. булевские (логические) операции:  
**not** (логическое отрицание), **and** (лог. И), **or** (лог. ИЛИ), **xor** (исключающее ИЛИ)
4. строковая операция (конкатенация) +

**div** – частное от деления

$$7 \text{ div } 3 = 2$$

***A:=10 div 3;***

**mod** – остаток от деления

$$7 \text{ mod } 3 = 1$$

***A:=25 mod 10;***

# ***Операторы***

предназначены для описания действий, которые будут выполняться при реализации алгоритма.

***Простые операторы*** не содержат в себе других операторов.

1. Оператор *присваивания* **:=**
2. Оператор *процедуры* состоит из имени, за которым в круглых скобках может располагаться список фактических параметров. **Swap(x, y).**
3. Оператор *перехода* **GoTo**

***Структурные операторы*** включают в себя другие операторы.

1. Составной оператор
2. Условные операторы (*if, case*)
3. Операторы цикла (*while, repeat, for*)
4. Оператор присоединения *with*