

Общие понятия

1. Запуск интегрированной среды Delphi.
 - 1) Выполнить команду Пуск-Программы-Borland Developer Studio 2006-Delphi for Microsoft Win32.
 - 2) Выполнить команду File-New-Other-Console Application.
2. Структура программы.

```

program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

begin
  { TODO -oUser -cConsole Main :
  Insert code here }
end.

```

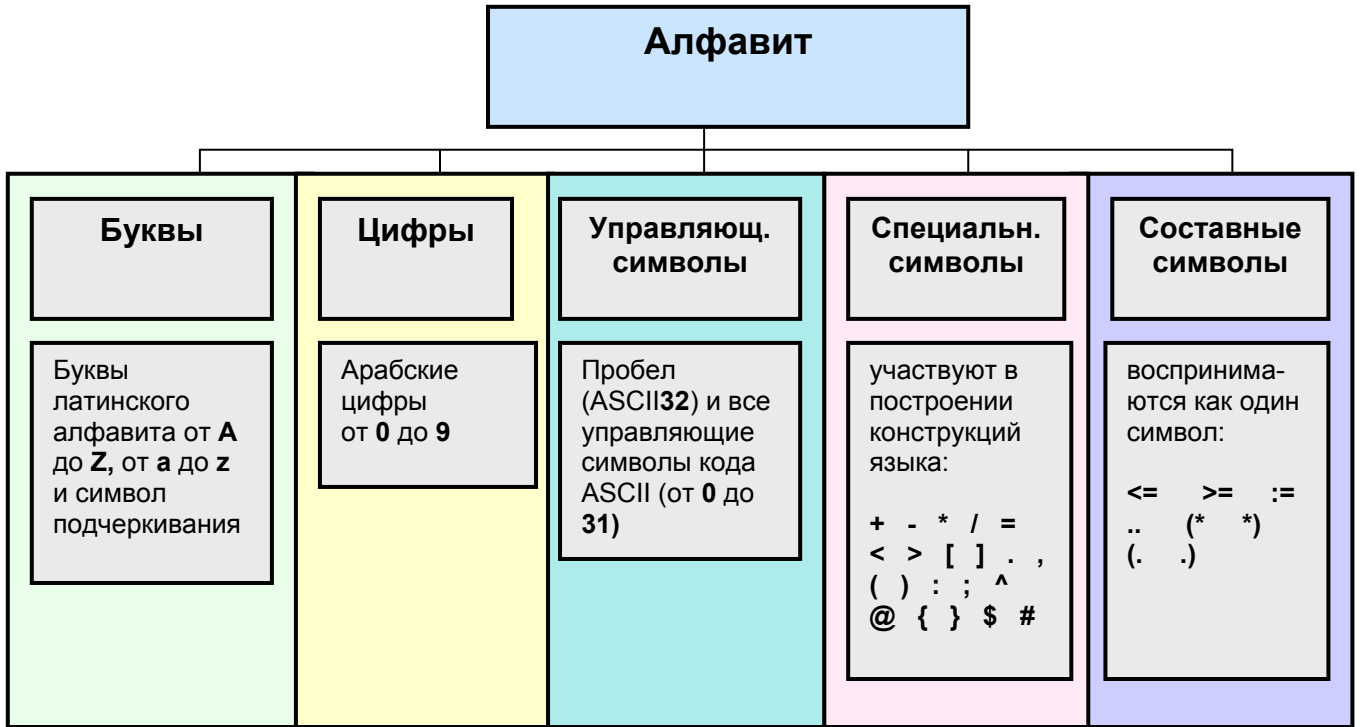
<code>program Project1;</code>	Заголовок программы (имя программы)
<code>{ \$APPTYPE CONSOLE }</code>	Признак консольного режима. Не удалять!!!
<code>uses SysUtils;</code>	Подсоединение системных утилит. Можно удалить.
<code>Begin</code>	Признак начала программы. Не удалять!!!
<code>{ TODO -oUser -cConsole Main : Insert code here }</code>	Здесь записывается сама программа.
<code>end.</code>	Признак конца программы. Не удалять!!!

3. Сохранить программу:
 - 1) Выполнить команду File-Save.
4. Запустить программу на выполнение:
 - 1) Выполнить команду Run-Run или нажать клавишу F9.

Алфавит языка PASCAL

Текст PASCAL-программы представляет собой последовательность *строк*, состоящих из символов, образующих *алфавит* языка.

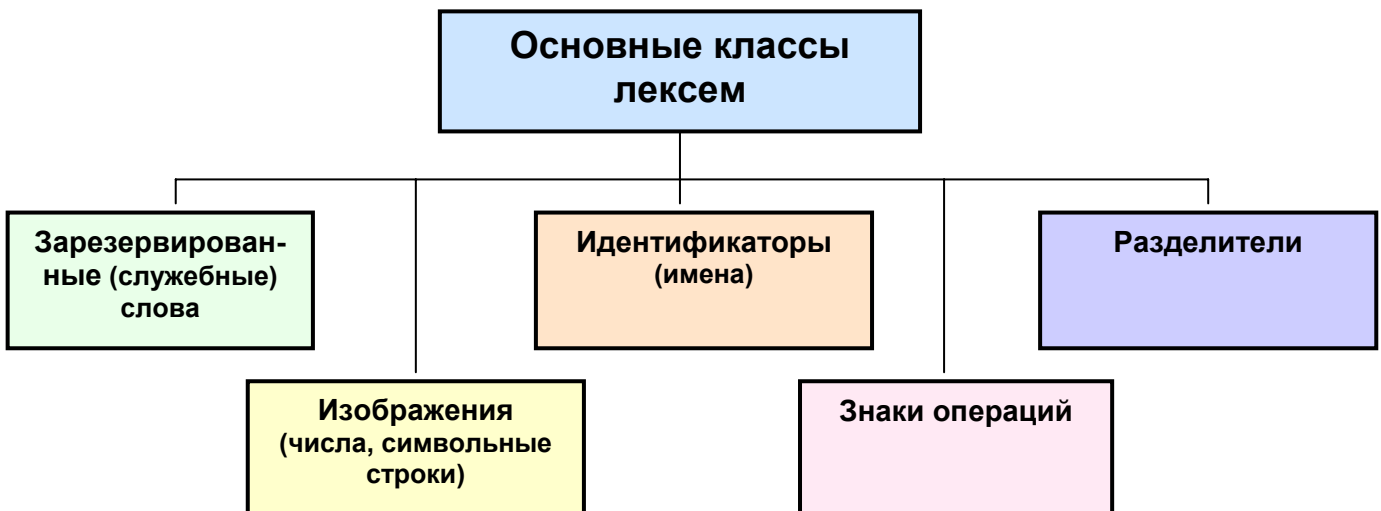
Максимальная длина строки - 126 символов.



Лексическая структура языка

Символы из алфавита языка используются для построения базовых элементов PASCAL-программ - *лексем*.

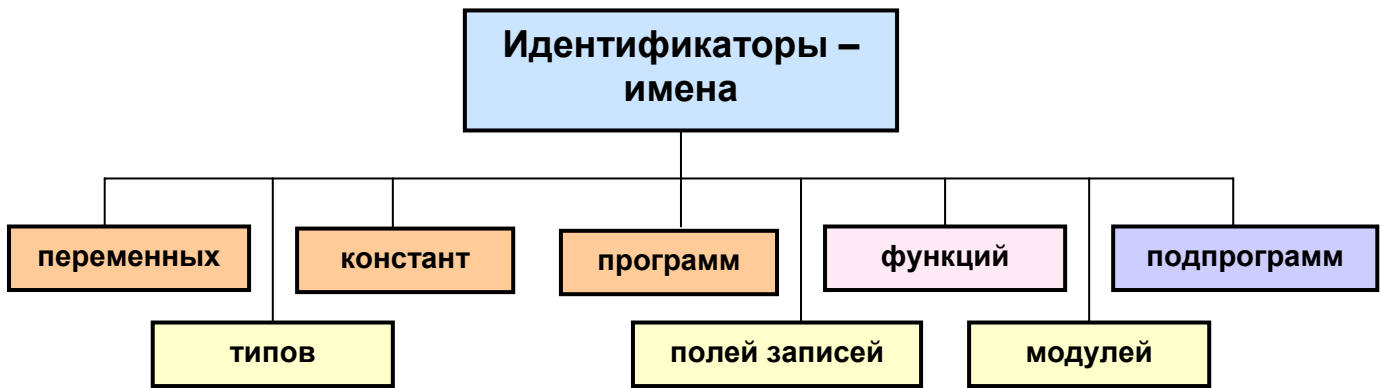
Лексема - минимальная единица языка, имеющая самостоятельный смысл.



1. Зарезервированные (служебные) слова:

absolute	div	goto	nil	repeat	var
and	do	if	not	set	virtual
array	downto	implementation	object	shl	while
asm	else	in	of	shr	with
assembler	end	inline	ot	string	xor
begin	external	interface	packed	then	
case	file		private	to	
const	for	interrupt	procedure	type	
constructor	forward	label	program	unit	
destructor	function	mod	record	uses	

2. Идентификаторы (имена)



Идентификатор – последовательность букв алфавита Pascal и цифр, начинающаяся с буквы. Значимыми являются только первые **63** символа.
Не допускается использование пробелов.

Примеры:

X	_Beta	программа
Y22M	5ABC	Figure/Fer
RAZMER	Pascal_Basic	INd67
index	_123	Const
Gamma	Y.22	ABCD_1234_EFG
Alfa_Beta	CH Gamma	Dos.Exec

3. Знаки операций

Формируются из одного или нескольких специальных символов и предназначены для задания действий по преобразованию данных и вычислений.

Условное обозначение операции	Наименование операции	Пример	Результат
+	сложение	$1 + 3$	4
-	вычитание	$5 - 2$	3
*	умножение	$4 * 8$	32
/	деление	$12 / 5$	2.4
div	деление целочисленное	$10 \text{ div } 3$	3
mod	остаток от целочисленного деления	$14 \text{ mod } 5$	4
:=	присваивание	$a := 3 + b$	

4. Разделители

Формируются из специальных символов и в основном используются для повышения наглядности текстов программ.

Примечание

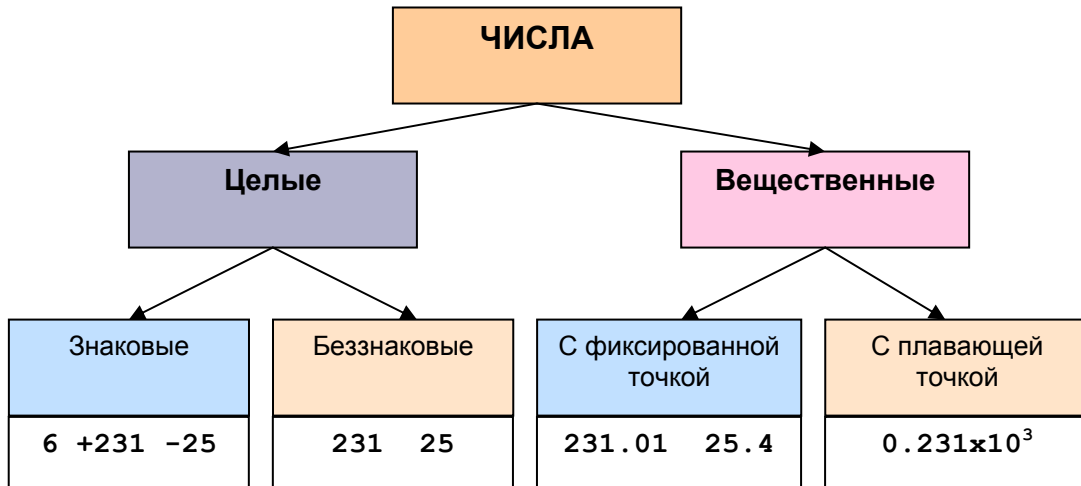
Символ пробела (ASCII, код 32).

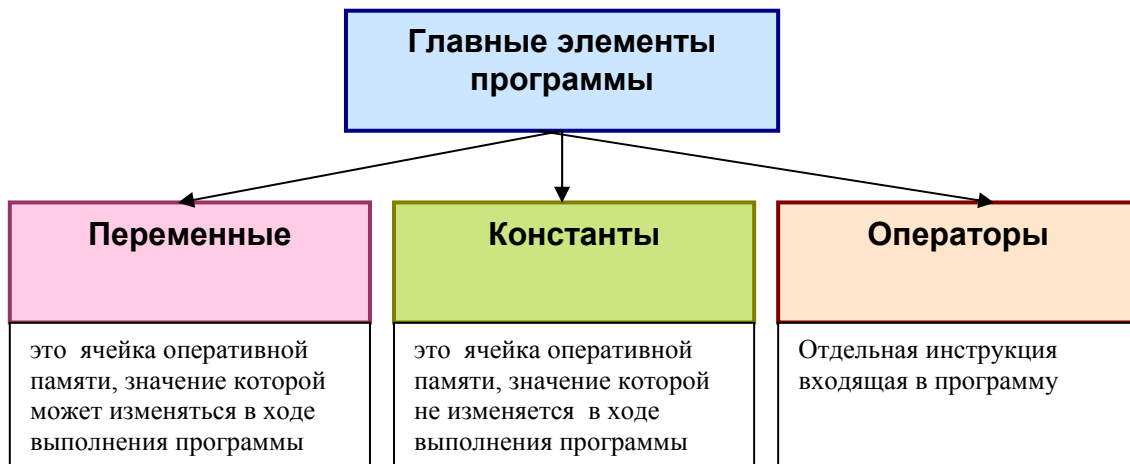
Управляющие символы (имеют ASCII-коды от 0 до 31). Эти символы могут применяться при описании строчных и символьных констант. Управляющие символы с ASCII-кодом 9 (табуляция), а также 10 и 13 (закрывающие строку).

5. Изображения

Группа лексем обозначает числа, символьные строки и некоторые другие значения.

Представление чисел





Основные характеристики переменной

Имя (идентификатор)	Alpha
Тип переменной	Integer
Значение	345
Адрес	\$1000 : \$2A34

Имя служит для того, чтобы в программе можно было обратиться к этому объекту (т.е. его идентифицировать).

Задание типа определяет область значений и вид внутреннего представления.

Значение переменной присваивается в программе путем считывания соответствующей константы, либо с помощью оператора присваивания.

Адрес является адресом в памяти, начиная с которого значение переменной записывается в память.

Объявление в программе переменных и констант

```
program Project1;  
  
{ $APPTYPE CONSOLE }  
  
uses  
    SysUtils;
```

```
Const Time=30;  
       Sum=10;
```

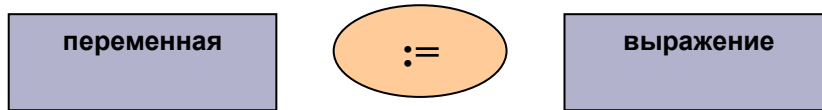
Раздел описания констант

```
Var    A, B: Integer;  
       C: Real;
```

Раздел описания переменных

```
begin  
    { TODO -oUser -cConsole Main :  
    Insert code here }  
end.
```

Оператор присваивания



Правила выполнения оператора присваивания

1. Вычисляется выражение в правой части оператора присваивания.
2. Переменной, указанной в левой части оператора присваивания присваивается вычисленное значение.
3. Переменная и выражение должны быть совместимы по типу.

```
Const A = 15;
Var
    I, J      : Integer;
    X, Y      : Real;
```

Правильные записи	Неправильные записи
<pre>I := 121; X := I + J mod A; X := X + 2; Y := 275;</pre>	<pre>I := 3.375; X + 3 := Y; A := 10;</pre>

Оператор вызова процедуры вывода Write, WriteLn

Оператор **Write** выдает на экран информацию, состоящую из букв (символов) и чисел.

Оператор **WriteLn** выдает на экран информацию, состоящую из букв (символов) и чисел и в заключение выполняет перевод курсора в первую позицию новой строки.

Примеры

```
A := 1; B := 2; C := 3;
```

```
Write (A); Write (B); Write (C);
```

1	2	3
---	---	---

```
Write (A,B,C);
```

1	2	3
---	---	---

```
WriteLn (A); WriteLn (B); WriteLn (C);
```

1
2
3

```
WriteLn (A,B,C);
```

1	2	3
---	---	---

```
WriteLn (B, '@', A, 'summa', C+A);
```

2	@	1	s	u	m	m	a	4
---	---	---	---	---	---	---	---	---

Diagram illustrating the output of the statement `WriteLn (B, '@', A, 'summa', C+A);`. The output is displayed as a sequence of characters in a row: 2, @, 1, s, u, m, m, a, 4. Arrows point from the corresponding arguments in the code to their respective positions in the output: B to '2', '@' to '@', A to '1', 'summa' to 's', 'u', 'm', 'm', 'a', and C+A to '4'.

Оператор вызова процедуры ввода (чтения) данных

Задача. Сложить два числа – 3 и 12.

Сравним две программы решения этой задачи:

```

program Z_1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

Var A, B : Integer;

begin
  A := 3; B := 12;

  writeln(A + B);

  Readln;
end.

```

```

program Z_2;

{$APPTYPE CONSOLE}

uses
  SysUtils;

Var A, B : Integer;

begin
  ReadLn (A, B) ;

  writeln(A + B);

  Readln;
end.

```

Программы отличаются только одной строкой. Первая программа не требует пояснений – она выдает на экран число 15. Во второй программе нигде не сказано, чему равны переменные A и B, а вместо этого включен оператор **ReadLn**. Он приказывает компьютеру остановиться и ждать, когда человек введет с клавиатуры определенную информацию, после чего продолжить работу.

В частности, **ReadLn (A, B)** будет ждать ввода двух целых чисел.

Пояснения к программе Z_2.

Первая программа после запуска будет работать без остановки, а в конце выдаст результат. Вторая программа на операторе **ReadLn** остановится и будет ждать. Во время этого ожидания человек должен на клавиатуре набрать число 3 (так как первым в списке оператора **ReadLn** стоит A), затем нажать клавишу пробела, затем набрать 12 и нажать клавишу **Enter**. Pascal воспринимает нажатие пробела, как сигнал человека о том, что закончен набор на клавиатуре одного числа и сейчас начнется набор другого. После набора на клавиатуре последнего числа необходимо нажать клавишу **Enter** в знак того, что ввод чисел для данного оператора **ReadLn** закончен и компьютер может продолжать работу. В соответствии с этим компьютер сразу же после нажатия **Enter** прекращает ожидание и направляет число 3 в память, в ячейку A, число же 12 - в ячейку B. На этом он считает выполнение оператора **ReadLn** законченным и переходит к следующему оператору. В результате будет выведено на экран число 15.

ReadLn позволяет человеку общаться с компьютером, вести с ним диалог во время выполнения программы.

Правила выполнения оператора вызова процедуры **ReadLn**

- При вводе числовых значений, они набираются на клавиатуре минимум через один пробел.
- После набора данных для одного оператора **ReadLn** нажимается клавиша ввода **Enter**.

- Значения переменных должны вводиться в строгом соответствии с синтаксисом языка Pascal. Если соответствие нарушено, то возникают ошибки ввода-вывода.
- Строковые (String) значения при вводе лучше всего располагать в отдельной строке.

Пример

```
Var A, B : Integer;

Begin
  ReadLn (A, B);
End.
```

Набираем на клавиатуре:

1-ый способ

или

2-ый способ

234 <пробел> 15<Enter>

234 <Enter> 15<Enter>

В результате A=234, B=15.

Математические функции

Вызов функции	Назначение функций	Пример	Результат
Abs (X)	Абсолютное значение X	Abs (-13)	13
Sqrt (X)	Квадратный корень из X, X>0	Sqrt (36)	6
Sqr (X)	Значение квадрата X	Sqr (11)	121
Round (X)	«Правильное» округление X до ближайшего целого	Round (2.34) Round (2.54)	2 3
Pi	Число π		3.1415...