

Строковые данные

Язык Pascal поддерживает стандартный символьный тип **Char** и динамические строки, описываемые типом **String** или **String[n]**.

Строковый тип

Строки называются *динамическими*, т.к. могут иметь различные длины в пределах объявленных границ.

Var

```
S32 :String[32]; { строчные значения, длиной не более 32}
S255 :String[255]; { строчные значения, длиной не более 255}
S :String; { без указания длины - String[255]}
```

Значением строки может быть любая последовательность символов, заключенная в одиночные кавычки:

```
'abcde - абвгд'
'123 ? 321'
' '
''''
```

Если включаются в строку управляющие коды, то строка состоит из склеенных кусков, при этом в такой записи не должно быть пробелов вне кавычек.

^G После сигнала нажмите **^J** клавишу пробел **^M^J**

'Номер п/п' **#179** 'Ф И О' **#179** 'Класс' **#179**

#7#32#179#32#32#179 (то же, что ^G' | |')

^G – звуковой сигнал (код 7)
 ^J – LF перевод строки (код 10)
 ^M – CR возврат каретки (код 13)

В Pascal строки можно рассматривать как массивы, и любой символ в строке можно изъять по его номеру. Отдельный символ совместим по типу со значением Char.

Var

```
CH:Char;
ST:String;
```

Begin

```
ST := 'Love' ;
CH := ST[1] ;      {CH=' L' }
ST[2] := 'i' ;    {ST=' Live' }
CH := 'X' ;
ST := CH ;        {ST=' X' }
```

End.

Длина строки

Для хранения любой строки отводится на 1 байт больше, чем указано при описании (256 байт для строк, размерность которых не указана явно). Вспомогательный байт располагается в самом начале строки – нулевой байт, в нем хранится текущая реальная длина строки. Поэтому максимальная длина строки может составлять 255 символов, т.к. в одном байте может быть представлено число не превышающее 255.

Значение этого байта автоматически изменяется программой при выполнении стандартных операций над строками.

Операция	Содержимое памяти							Значение S
Var S: String[6];	S[0]	S[1]	S[2]	S[3]	S[4]	S[5]	S[6]	
S:= '';	0							''
S:= 'Yes';	3	Y	e	s				'Yes'
S[4]:= '!';	3	Y	e	s	!			'Yes'
S:=S + 'no'	5	Y	e	s	n	o		'Yesno'
S[4]:= '!'	5	Y	e	s	!	o		'Yes'

Использование в процедурах и функциях параметров-массивов и параметров-строк

Основное правило:

Типом любого параметра в списке формальных параметров может быть только стандартный или ранее объявленный тип.

Типом любого параметра в списке формальных параметров может быть только стандартный или ранее объявленный тип.

```
Function XSTR(S:String[32]):String[6];
```

Корректно это выглядит таким образом:

```
Type STRING32=String[32];  
    STRING6 =String[6];  
    . . .  
Function XSTR(S:STRING32):STRING6;
```

Операции над строками

1. Присваивание, слияние

Слияние строк записывается в естественном виде. Если сумма получается длиннее, чем описанная длина левой части оператора присваивания, излишек отсекается.

```

Var
    S1,S2,S3:String;
Begin
    S1 := ' Вам ' ;
    S2 := ' привет' ;
    S3 :=S1+S2;      { S3 = ' Вам привет' }
    S3 :=S3+' !' ;   { S3 = ' Вам привет !' }
End.

```

2. Сравнение

Сравнение строк происходит посимвольно, начиная от первого символа в строке.

```

'abcd'='abcd'      →   True
'abcd'<>'abcde'   →   True
'abcd'<>'  abcd'   →   True

'abcd'>'abcD'     { 'd' >'D' }
'abcd'>'abc'      { 'd' >' ' }
'aBcd'<'ab'      { 'B' <'b' }
' '>' '          { #32 >' ' }

```

Пример программы, использующей процедуры и функции операций над строками**Пример**

Перевод символьной строки в числовой тип Word.

```
{R-}  
Var LongV:LongInt;  
    WordV:Word;  
    S:String;  
.  
.  
.  
WordV:=0;  
Val (S,LongV,ErrCode) ;  
  
If ErrCode=0  
  then  
    Begin  
      If (LongV>=0) and (LongV<=65535)  
        then WordV:=LongV  
        else  
WriteLn('Ошибка диапазона при преобразовании ',LongV)  
      End  
    else  
      WriteLn('Ошибка в строке ',S,' в символе',S[ErrCode]);
```