

# **Строковый тип данных**

Для обработки строковой информации в Турбо Паскаль введен строковый тип данных.

***Строка*** - последовательность из определенного количества символов.

Количество символов последовательности называется ***длиной строки***.

Описание:

***var s: string[n];***

***var s: string;***

n - максимально возможная длина строки - целое число в диапазоне 1..255.

Если этот параметр опущен, то по умолчанию он принимается равным 255.

Пример:

**‘Текстовая строка’**

**#54#32#61**

‘ ‘

определена операция слияния  
(конкантенации), обозначаемая  
знаком +.

Например:

```
a := 'Turbo';  
b := 'Pascal';  
c := a + b;
```

В этом примере  
переменная c приобретет  
значение

**'TurboPascal'**

# Кроме слияния над строками определены операции сравнения

$\langle, \rangle, =, \langle \rangle, \langle =, \rangle =.$

В результате сравнения двух строк получается логическое значение (true или false). Сравнение строк производится слева направо посимвольно до первого несовпадающего символа, большей считается та строка, в которой первый несовпадающий символ имеет больший код в таблице кодировки. Если строки имеют различную длину, но в общей части символы совпадают, считается, что короткая строка меньше. Строки равны, если они имеют равную длину и соответствующие символы совпадают.

**‘строка’  $\langle \rangle$  ‘строки’** (верно, т.к. не совпадают последние символы);

**‘Аbc’  $\langle$  ‘abc’** (отношение истинно, т.к. код символа ‘А’ равен 65 в десятичной системе

счисления, а код символа ‘а’ – 97);

**‘год’  $\rangle$  ‘век’** (отношение верно, т.к. буква ‘г’ в алфавите стоит после буквы ‘в’, а, следовательно

# **Процедуры и функции для работы со строками**

**Delete(str,n,m)**-процедура удаляет из строки `str` `m` символов начиная с `n`-го, таким образом, сама строка изменяется.

*Пример.*

```
s := 'Система Turbo Pascal';
```

```
delete(s,8,6);
```

```
{s будет равно 'Система Pascal'}
```

**Insert(str1,str2,n)** - процедура вставляет строку str1 в строку str2 начиная с n-го символа, при этом первая строка остаётся такой же, как и была, а вторая получает новое значение.

Пример.

```
s := 'Система Pascal';
```

```
insert('Turbo ',s,9);
```

```
{s будет равно 'Система Turbo Pascal'}
```

**Copy(str,n,m)** - функция копирует m символов строки str начиная с n-го символа, при этом исходная строка не меняется.

Пример.

```
s := 'Система Turbo Pascal';
```

```
s2 := copy(s, 1, 7);
```

```
{s2 будет равно 'Система'}
```

```
s3 := copy(s, 9, 5); {s3 будет равно 'Turbo'}
```

```
s4 := copy(s, 15, 6);
```

```
{s4 будет равно 'Pascal'}
```

**Length(str)** - функция возвращает в качестве результата значение текущей длины строки-параметра, результат целое число.

Пример.

```
n := length('Pascal'); {n будет равно 6}
```

**Str(n, str)** - процедура преобразует численное выражение n в его строковое представление и помещает результат в str.

Пример.

```
Str (1234,str);  
{str будет равно '1234'}
```

**Pos(str1,str2)** - результат функции  
целое число, и оно определяет номер  
элемента, которого начинается первое  
вхождение str1 в строку str2, то  
значение функции равно 0.

Пример.

```
s := 'Система Turbo Pascal';
```

```
x1 := pos('Pascal', s);
```

```
{x1 будет равно 15}
```

```
x2 := pos('Basic', s); {x2 будет равно 0}
```

**Val(str,n,k)** - процедура переводит строковое значение в числовое. Если данная строка действительно является записью числа (целого или вещественного), то  $k=0$ , а  $n$  – это искомое число. Иначе  $k$  будет равно номеру первого символа, с которым процедура Val «не справилась».

Пример

Val('1234',n,k)  $n=1234$   $k=0$

Val('12-34',n,k)  $k=3$

Дано название футбольного клуба. Определить количество СИМВОЛОВ В НЕМ.

```
program rr;
```

```
var
```

```
st:string;
```

```
n:integer;
```

```
begin
```

```
write('задайте название клуба - ');
```

```
readln(st);
```

```
n:=length(st);
```

```
writeln('n=',n);
```

```
end.
```