

Символы и строки

Описание строк

В разделе **var**:

var <имя_строки>: string[<длина>]

Максимальная длина строки - 255 символов. Нумерация - начиная с 0, нулевой байт хранит длину строки.

Примеры описаний:

var s1: string[10]; (*строка длиной 10 символов*)

s2: string; (*строка длиной 255 символов*)

var c: char;

s: string[1];

Неименованные константы (найти ошибку)

```
c:='z';      {c: char}  
s:='abc';{s: string}  
c1:='zzz';  {c1: char}
```

Пустая строка:

```
st:= "";
```

Апостроф:

```
s:='Don"t worry about the apostrophe!'; {Don't worry about the apostrophe!}
```

Нетипизированные константы

```
const c3 = ""'; {это один символ - апостроф!}  
s3 = 'This is a string';
```

Типизированные константы

```
const c4: char = ""'; {это один символ - апостроф!}  
c4: string[20] = 'This is a string';
```

Действия с символами

Операции

Унарная операция

`#<положительная_неименованная_константа_целого_типа>`

`#100 = 'd'`

`#39 = ''' {апостроф}`

`#232 = 'ш'`

`#1000 = 'ш' {почему?}`

Операции сравнения: `<`, `<>`, `>`, `=`.

Стандартные функции

Функция `chr(k:byte):char` преобразует номер символа в символ аналогично `#`:

`c := chr(48); {c: char}`

`{c = '0'}`

Обратная к функции `chr()` - `ord()`. `ord(c:char):byte;`

`i:=Ord('0'); {i=48}`

`ord(chr(k)) = k` и `chr(ord(c)) = c`

pred(), succ(), inc() и dec():

Например:

pred('[') = 'z'

succ('z') = '{'

inc('a') = 'b'

inc('c', 2) = 'e'

dec('z') = 'y'

dec(#0, 4) = '№' {#252}

uppercase(c: char):char преобразует строчную латинскую букву в прописную.

Стандартные функции и процедуры обработки строк

- **concat(s1,_,sN:string):string:**

```
concat('abc','3de',' ','X','yz') = 'abc3de Xyz'
```

```
S:string[10];
```

```
s1:= 'abc';
```

```
S:= concat(s1,'3de',' ','X','yzwww'); {s= 'abc3de Xyz'}
```

- **copy(s:string;i,k:byte):string:**

```
copy('abc3de Xyz',2,4) = 'bc3d' copy(s,i,length(s1)) = 'Xyz'
```

```
copy('abc3de Xyz',12,4) = ''
```

```
copy('abc3de Xyz',8,14) = 'Xyz'
```

```
S:= copy('abc3de Xyz',8,14);
```

- **delete(s:string;i,k:byte):**

```
{s = 'abc3de Xyz'}
```

```
delete(s,2,3);
```

```
{s = 'ade Xyz'}
```

```
{s = 'abc3de Xyz'}
```

```
delete(s,8,13);
```

```
{s = 'abc3de '}
```

- **insert(ss,s:string;i:byte):**

```
{s = 'abc3de Xyz'}
```

```
insert('xyz',s,2);
```

```
= 'axyzbc3de Xyz'}
```

```
{s = 'abc3dexyz'}
```

```
{s = 'abc3de'}
```

```
insert('xyz',s,12);
```

```
{s
```

- **length(s:string):byte:**

```
length('abc3de Xyz') = 10
```

- **pos(ss,s:string):byte:**

```
pos('X', 'abc3de Xyz') = 8;
```

```
pos('bc3', 'abc3de Xyz') = 2;
```

```
pos(' ', 'abc3de Xyz') = 7;
```

```
pos('bcd', 'abc3de Xyz') = 0;
```

```
pos('X', 'abc3de XyzXX') = 8;
```

- **str(x[:w[:d]],s:string):**

```
str(156.4:7:2,s); {s = ' 156.4 '}
```

- **val(s:string;i:<арифметический_тип>;err:byte):**

```
{s = '15.47'}
```

```
val(s,x,err); {x = 15.47}
```

Операции со строками

Сравнения (=, >, <)

'abc' < 'xyz'

'a' < 'abc'

'1200' < '45'

'Anny' < 'anny'

Обращение к компонентам строки

<имя_строки>[<индекс>]

Например:

```
{s = '15.47'}
```

```
c := s[2];
```

```
{c = '.'}
```

```
!!!s[i] := 'a';
```

Изменение символа в строке при помощи length(), concat() и copy().

```
1. s1 := copy(s, 1, k-1);  
2. s2 := new_char;  
3. s3 := copy(s, k+1, length(s) - k);  
4. s := concat(s1, s2, s3);
```

Или

```
s := concat(copy(s, 1, k-1), new_char,  
copy(s, k+1, length(s) - k));
```

Конкатенация

```
s := copy(s, 1, k-1) + new_char + copy(s, k+1, length(s) - k);
```