

The background features a white surface with several large, flowing, semi-transparent shapes in shades of green, purple, and light blue. Interspersed among these shapes are numerous small, bright yellow triangles pointing in various directions, creating a festive or celebratory atmosphere.

СТРОКИ



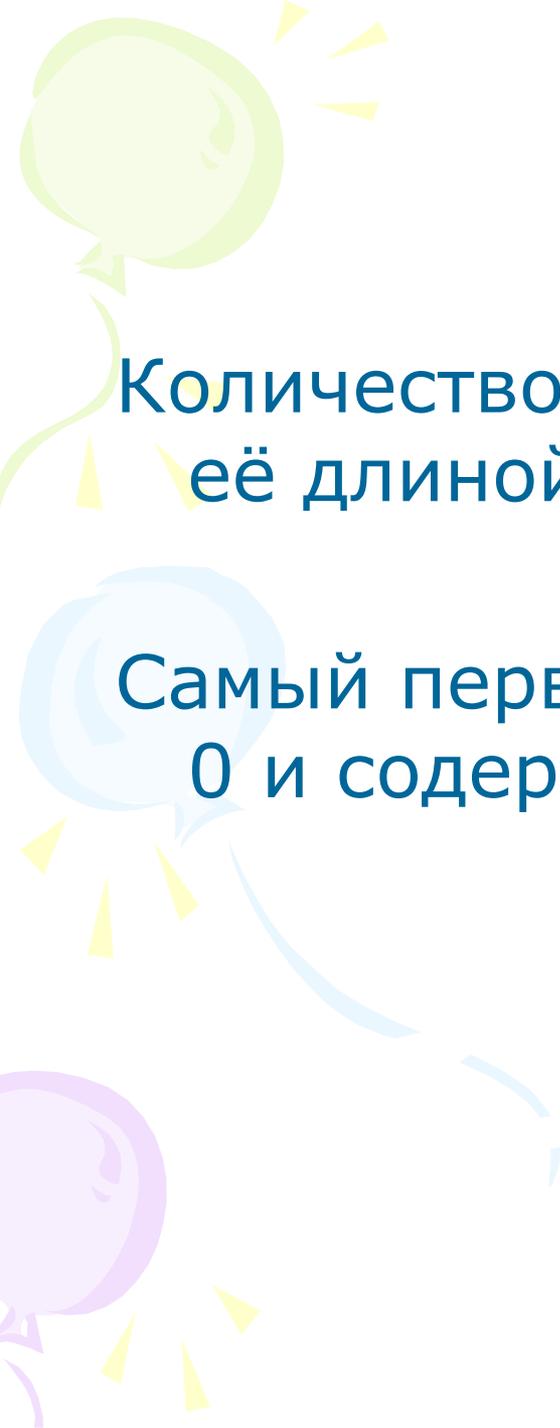
Строка

- это упорядоченная последовательность СИМВОЛОВ.



Каждый символ строки занимает 1 байт.





Строка

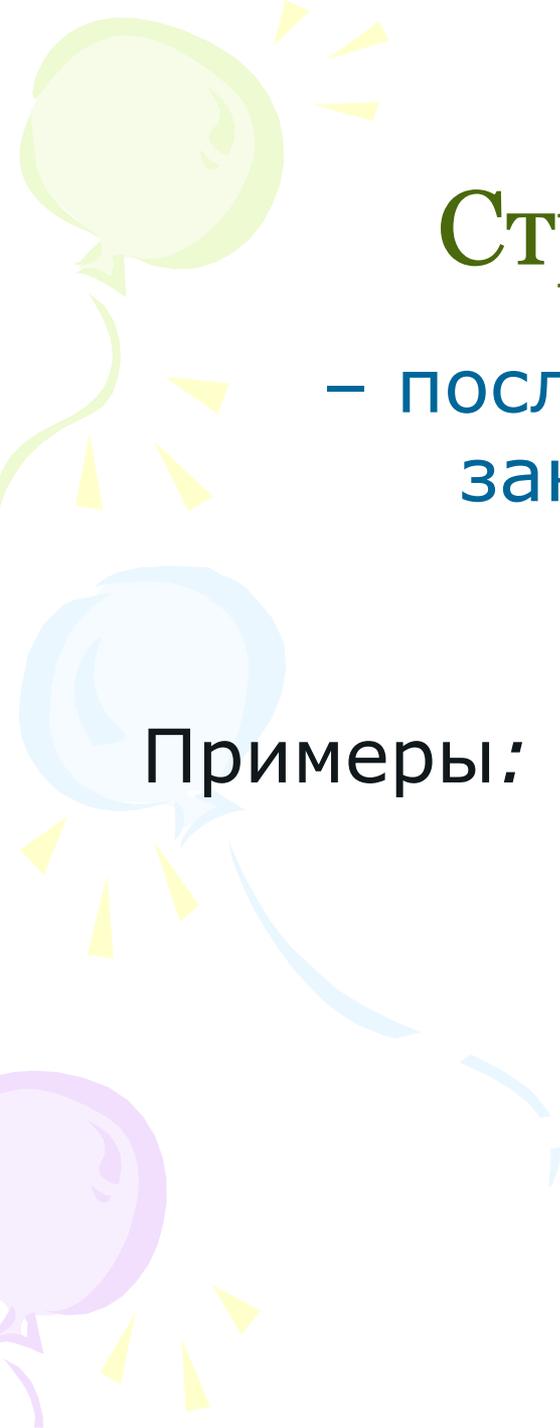
Количество символов в строке называется её длиной.

Самый первый байт в строке имеет индекс 0 и содержит текущую длину строки.



Строка

Первый значащий символ строки занимает второй байт и имеет индекс 1.



Строковая константа

– последовательность символов,
заключенных в апострофы.

Примеры:

```
'12+6'
```

```
' это строковая константа '
```

```
'3,1415'
```

```
''
```



Строковая переменная

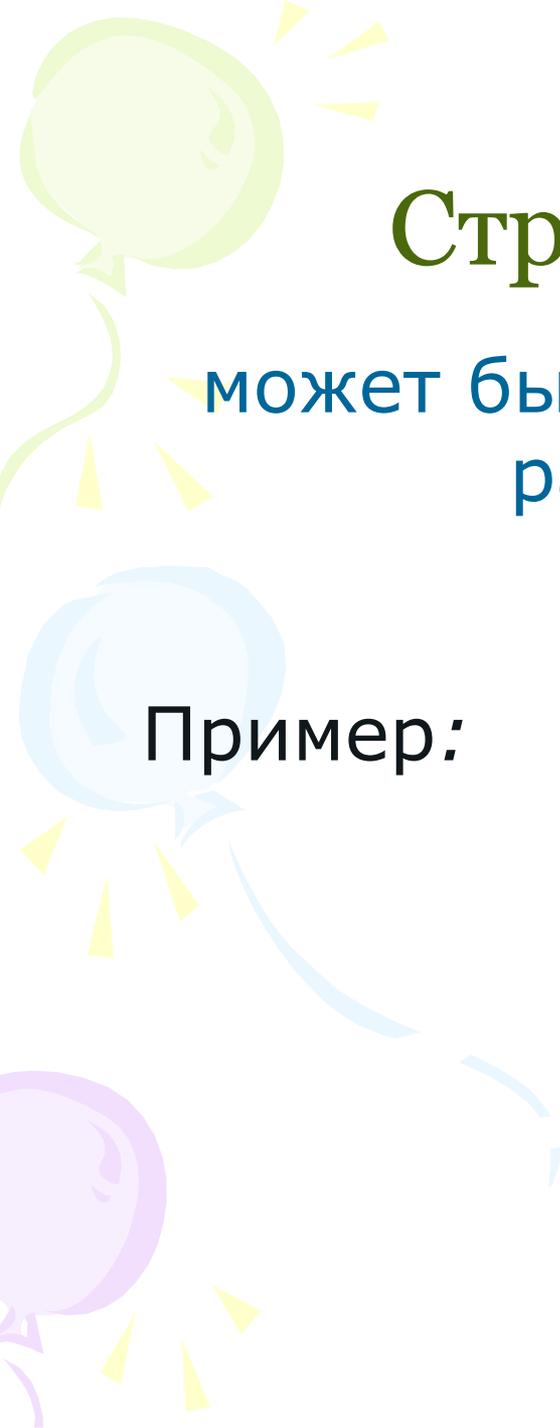
описывается в разделе описания переменных:

var

<идентификатор>:string[<максимальная длина строки>];

Примеры:

```
var slovo :string[30];  
stroka :string;
```

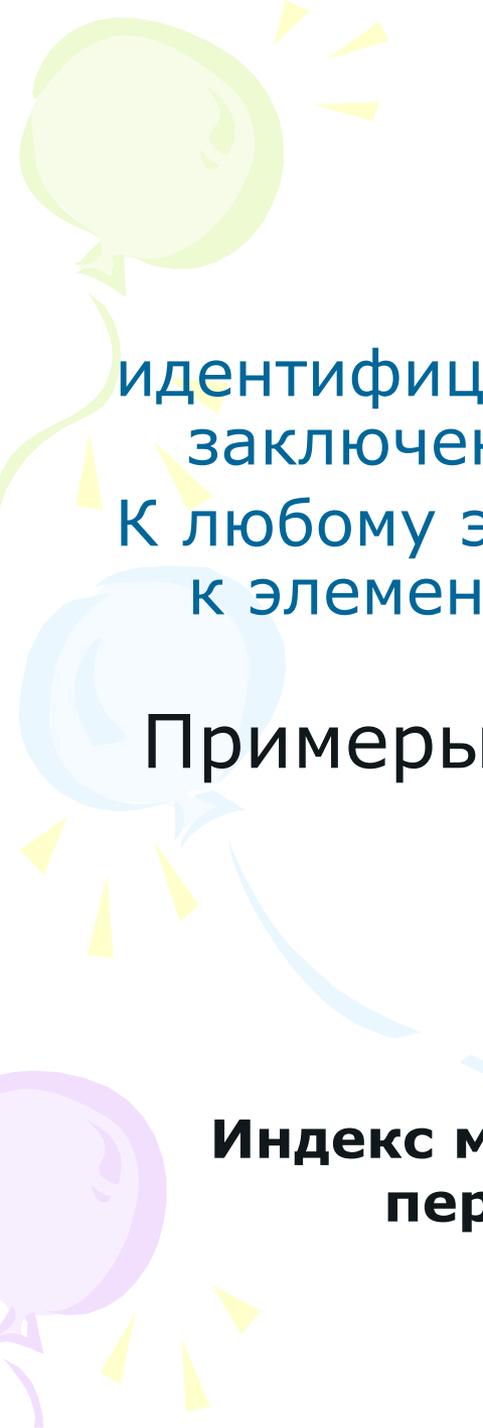


Строковая переменная

может быть описана с использованием раздела описания типов.

Пример:

```
type stroka = string[50];  
var s : stroka;
```



Элементы строки

идентифицируются именем строки с индексом, заключенным в квадратные скобки.

К любому элементу строки можно обратиться как к элементу массива **array [0..n] of char**

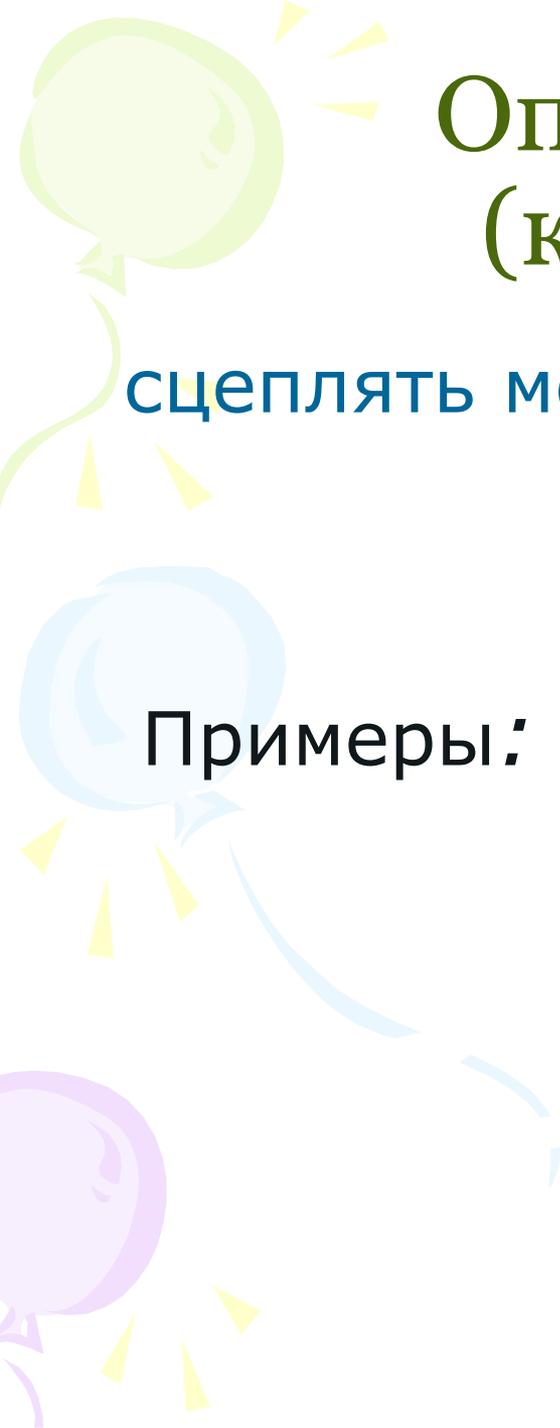
Примеры:

```
stroka[3], slovo[15]
```

Индекс может быть положительной константой, переменной, выражением целого типа.



ОПЕРАЦИИ НАД СТРОКАМИ



Операция сцепления (конкатенации) (+)

сцеплять можно как строковые константы,
так и переменные.

Примеры:

```
K := '45' + '12' + 'abc';
```

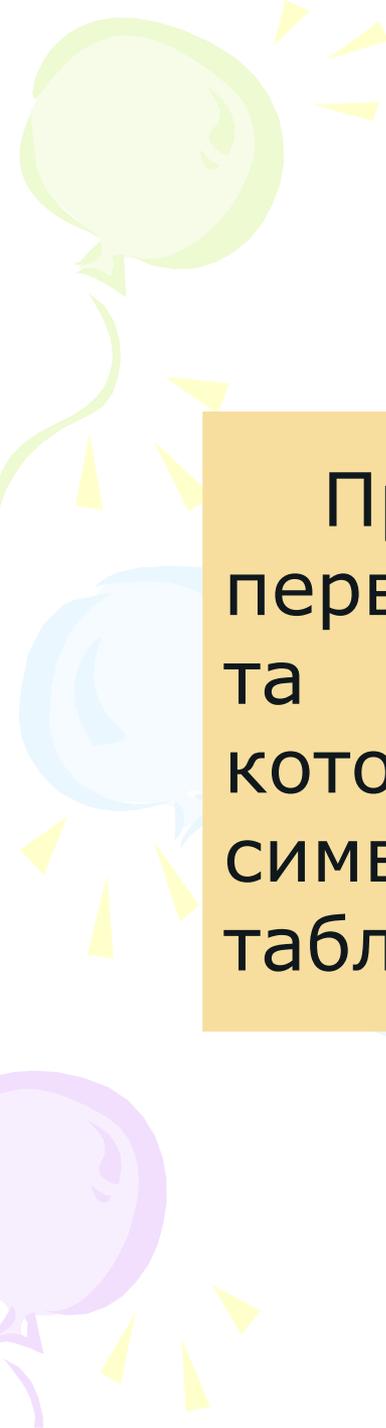
результат:

```
'4512abc'
```

Операции отношения:

=, <, >, <=, >=, <>

Выражение	Результат
'True1' < 'True2'	true
'Mother' > 'MOTHER'	true
'145' <> ' 145'	true
'Car' = 'Car'	true



Сравнение строк

Производится слева направо до первого несовпадающего символа, и та строка считается больше, в которой первый несовпадающей символ имеет больший номер в таблице символьной кодировки.

Функция COPY (s, p, n)

выделяет из строки **S** подстроку длиной **n** символов, начиная с позиции **p**.

n и **p** – целочисленные выражения

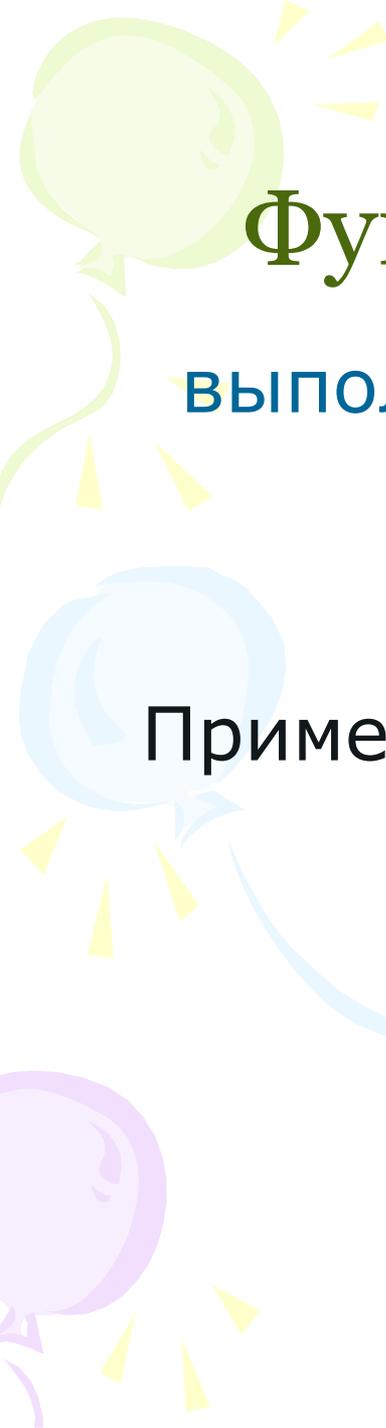
Примеры:

```
h := copy ('123456789', 2, 4);
```

результат: '2345'

```
t := copy (' водопровод', 6, 6);
```

результат: 'провод'

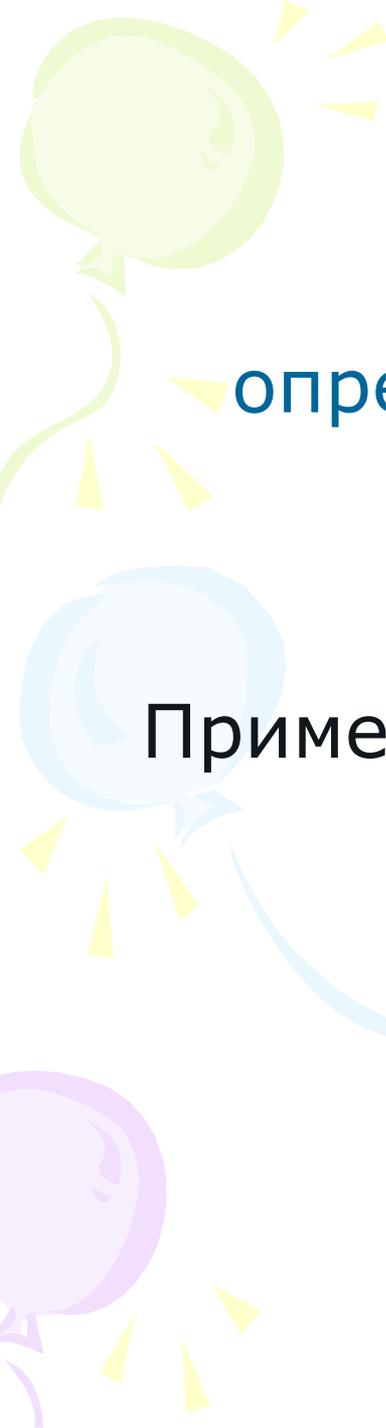


Функция $\text{CONCAT}(s_1, s_2, \dots, s_n)$

выполняет сцепление строк s_1, s_2, \dots, s_n в одну строку.

Пример:

```
W := concat('абра', 'када', 'бра');  
результат: 'абракадабра'
```



Функция LENGTH(s)

определяет текущую длину строки **s**.

Примеры:

```
S := length ('test#3');
```

результат: 6

```
G := length ('(x+y)/2');
```

результат: 7

Результат – значение целого типа.

Функция POS (s_1, s_2)

обнаруживает первое появление в строке s_2 подстроки s_1 .

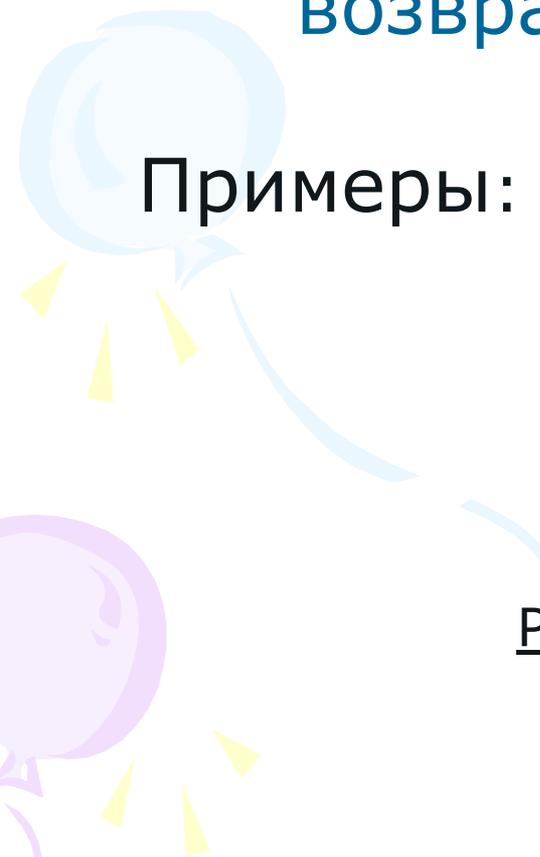
Примерь `A := pos ('12', '12abc12');`
результат: 1
`B := pos ('t', 'my name');`
результат: 0

Результат – целое число, равное номеру позиции, где находится первый символ подстроки s_1 .



Функция `UPCASE(C: char)`

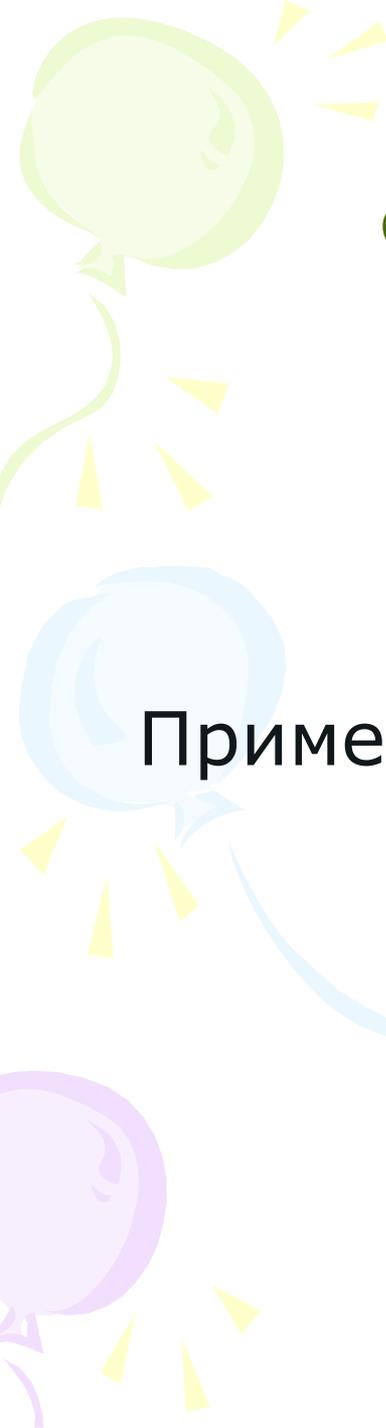
преобразует строчную латинскую букву в прописную. Любые другие символы возвращаются без преобразования.



Примеры:

```
f := 'pascal';  
f[1] := upcase (s[1]);  
результат: f = 'Pascal'
```

Результат – значение типа char.



Функция ORD(ch: char)

возвращает код символа ch.

Примеры:

```
T := ord ('M');
```

результат: 77

```
E := ord ('=');
```

результат: 61

Результат – значение типа byte.

Функция CHR(n: byte)

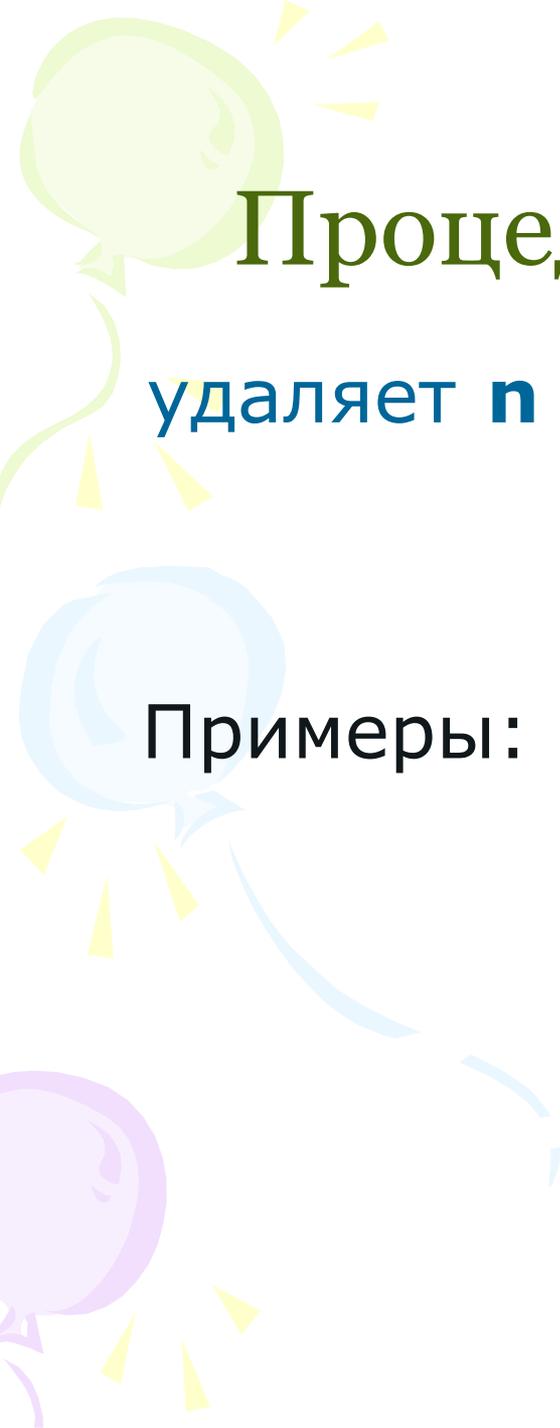
возвращает символ по коду, равному значению выражения n.

Если n можно представить как числовую константу, то можно также пользоваться записью #n.

Примеры:

```
w := chr (73);  
результат: 'I'  
d := chr (100 + 5);  
результат: 'i'
```

Результат – значение типа char.



Процедура DELETE(s, poz, n)

удаляет **n** символов из строки **s**, начиная с позиции **poz**.

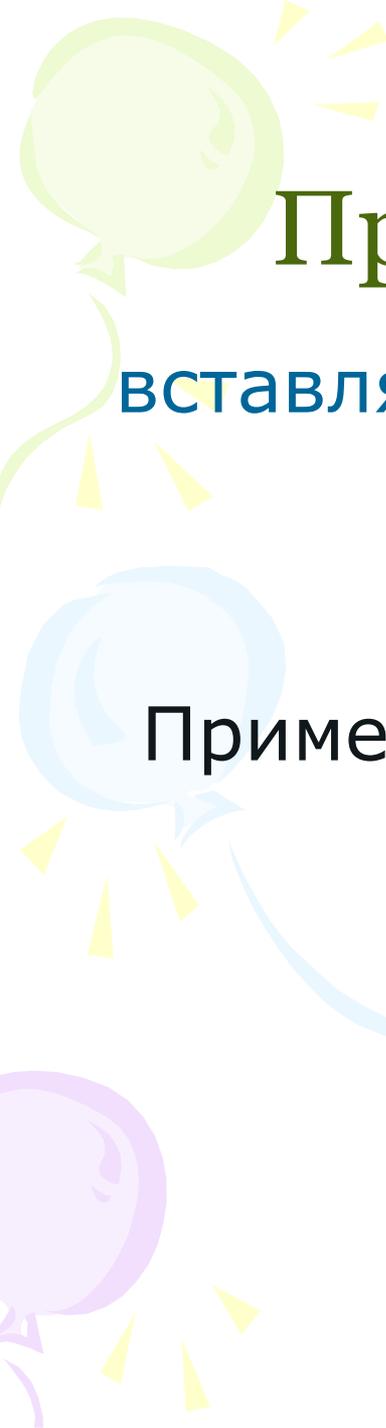
Примеры:

```
delete ('мармелад', 1, 5);
```

результат: 'лад'

```
delete ('abcdefg', 2, 6);
```

результат: 'а'



Процедура `INSERT(s1, s2, p)`

вставляет строку **s₁** в строку **s₂**, начиная с позиции **p**.

Примеры:

```
insert ('IBM-', 'ЭВМ РС', 5);  
результат: 'ЭВМ IBM-PC'  
insert('#', 'map2', 4);  
результат: 'map#2'
```

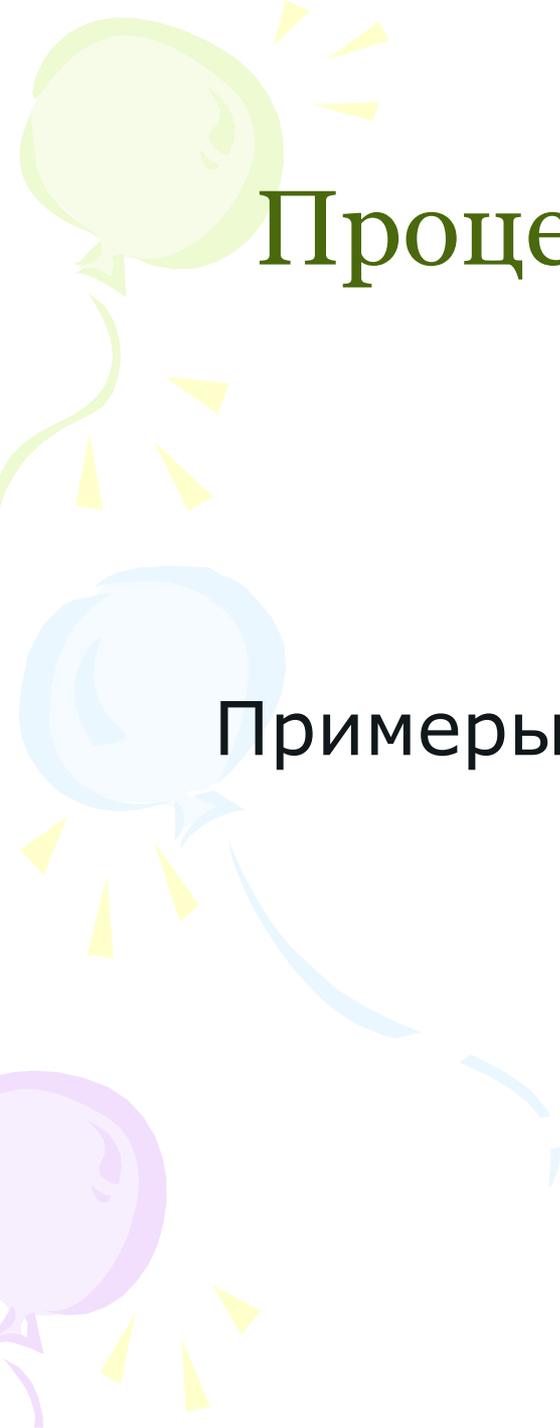
Процедура STR(x:a:b,stroka)

преобразует число **x** любого вещественного или целого типов в строку СИМВОЛОВ **stroka**

a и **b** (если они присутствуют) - задают формат преобразования.

a – определяет общую ширину поля, выделенного под соответствующее символьное представление вещественного или целого числа **x**

b – количество символов в дробной части (этот параметр имеет смысл только в том случае, когда **x** – вещественное число)



Процедура STR(x:a:b,stroka)

```
str (1234, t);
```

результат: t = '1234'

```
str (452.567, t);
```

Примеры: *результат:* t = '452.567'

```
str (4.52567e+2, t);
```

результат: t = '4.52567e+2'

Процедура VAL (s, x, code)

преобразует строковое значение S в его численное представление X типа real или integer.

Примеры

```
val ('-1268', i, k);
```

результат: i=-1268, k=0

```
val ('12-45', x, k);
```

результат: x = 0, k = 3

Параметр Code содержит признак ошибки преобразования (0 - нет ошибки)

Задания на закрепление

1. Запишите результат выполнения следующих стандартных процедур:

```
str (365.874, st)           str (2.89784e+4, st)
val ('9876', x, k)         val ('1.0098e+6', y, k)
val ('679-8', y, k)       val ('2,567', y, k)
```

2. Определите результат выполнения программы:

```
var  s: string;    k, sum, d, i: integer;
begin
  sum := 0; s := '12r345ty';
  for i := 1 to length (s) do
    begin
      val (s[i], d, k);
      if k = 0 then sum := sum + d
    end;
  writeln (sum:6)
end.
```

Задания на закрепление

3. Поставьте знак сравнения (>, <, =) между парами строк и объясните свой ответ.

'Компьютер' _____ 'Комп'

'Stroka' _____ 'stroka'

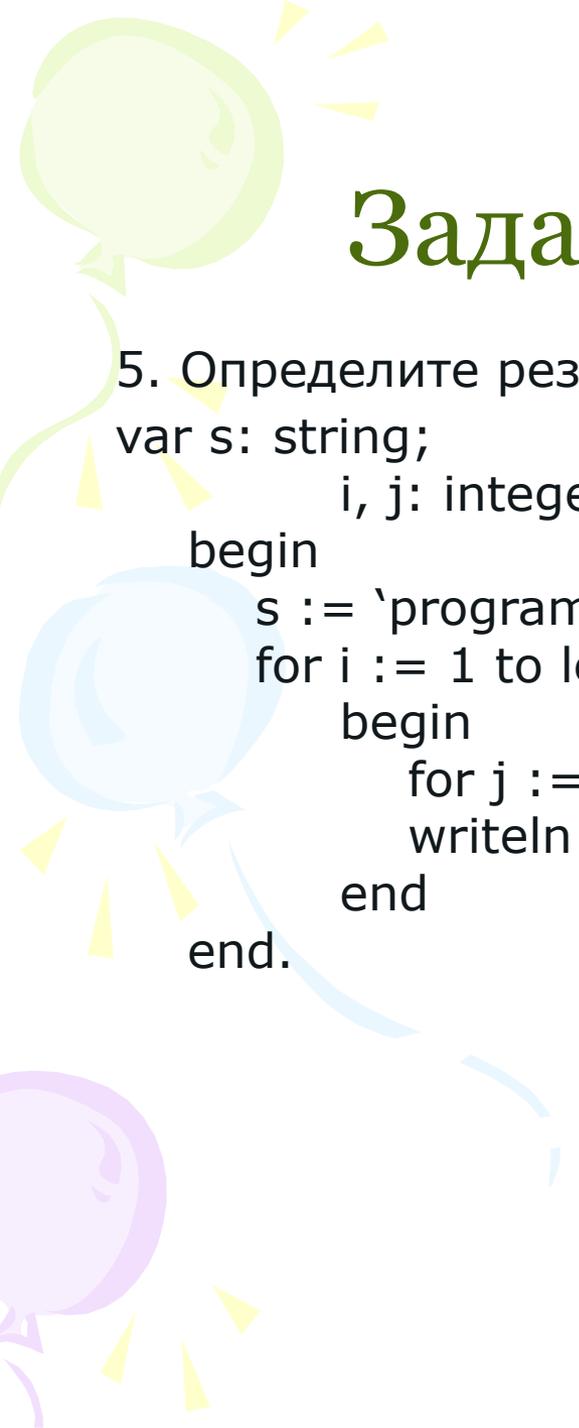
'кошка' _____ 'кошка'

'кот' _____ 'кот'

'муха' _____ 'слон'

4. Определите результат выполнения программы.

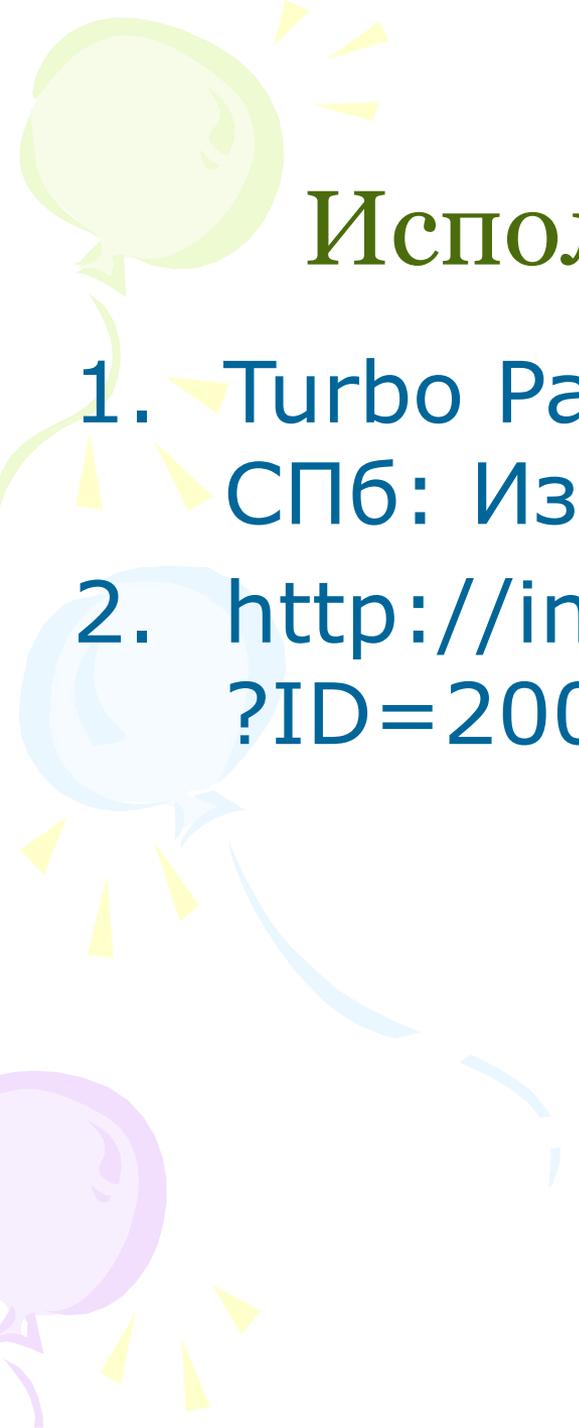
```
var s: string;  
    i, k: integer;  
begin  
    s := 'абракадабра';  
    k := 0;  
    for i := 1 to length (s) do  
        if s[i] = 'a' {if copy (s, i, 1) = 'a'}  
            then k := k + 1;  
    writeln (k: 5)  
end.
```



Задания на закрепление

5. Определите результат выполнения программы:

```
var s: string;  
    i, j: integer;  
begin  
    s := 'programm';  
    for i := 1 to length (s) do  
        begin  
            for j := 1 to i - 1 do write(' ');  
            writeln (s[i])  
        end  
    end.  
end.
```



Используемые материалы

1. Turbo Pascal. С. А. Немнюгин.-
СПб: Издательство "Питер", 2000.
2. <http://inf.1september.ru/article.php?ID=200800802>