



# Строки в Pascal. Строковые функции

Автор: преподаватель  
ГОУ ВО МО «ГГТУ»  
Промышленно-экономический колледж  
Романова Юлия Евгеньевна

г. Орехово-Зуево  
Московская область

# Определение

**Строка** – упорядоченная последовательность символов, заключенная в апострофы.

Количество символов в строке называется ее **длиной**.

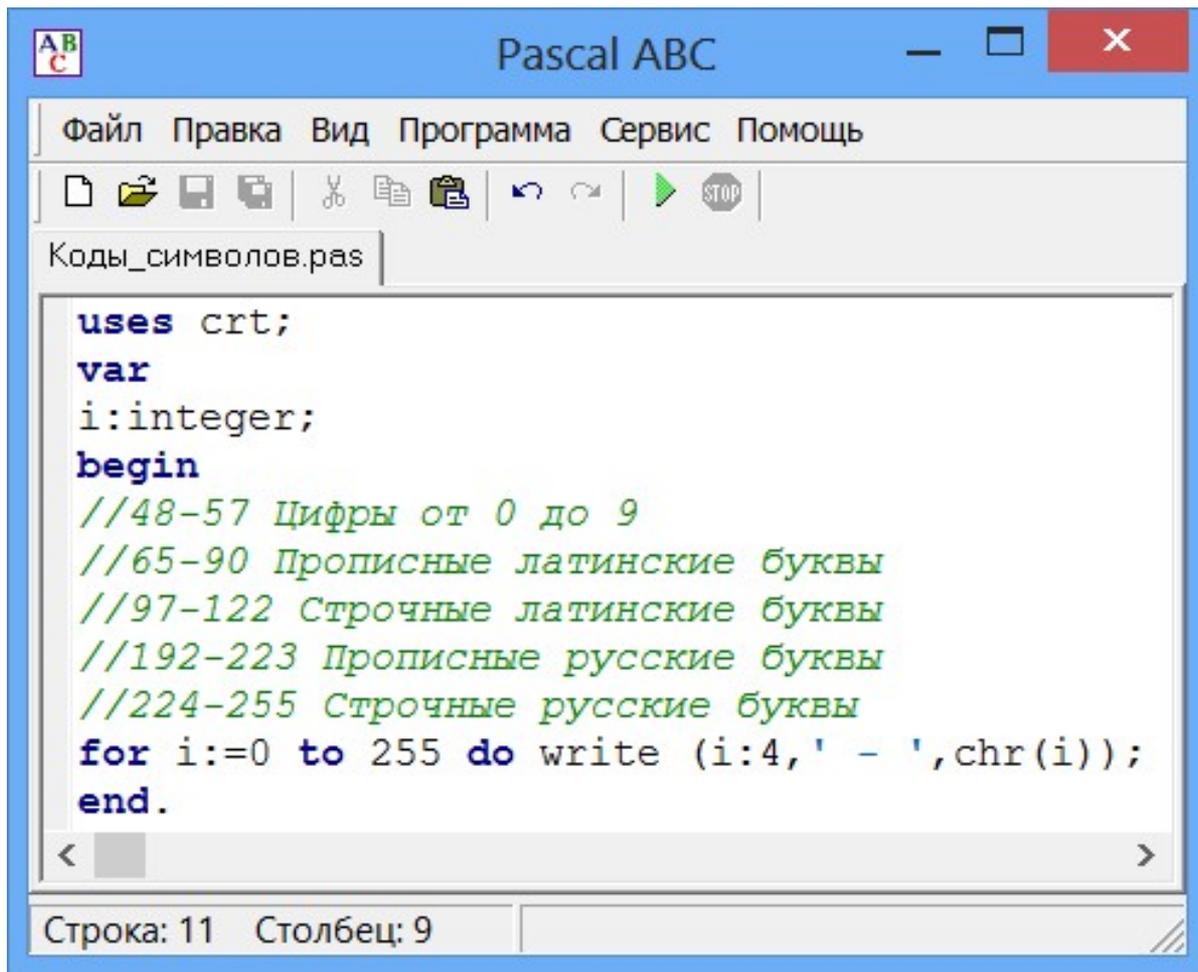
Длина строки может лежать в диапазоне от **0** до **255**.

Каждый символ строковой величины занимает **1 байт** памяти и имеет числовой код в соответствии с таблицей кодов ASCII

# Таблица кодов ASCII

- Символ может быть записан, ограниченный апострофами ('a') или записан в коде (#97)
- **chr(<код символа>)** – функция, возвращающая символ по его коду в кодовой таблице ASCII.
- **ord(<символ>)** – функция, возвращающая код по соответствующему символу.

# Вывод таблицы кодов



```
uses crt;
var
  i:integer;
begin
  //48-57 Цифры от 0 до 9
  //65-90 Прописные латинские буквы
  //97-122 Строчные латинские буквы
  //192-223 Прописные русские буквы
  //224-255 Строчные русские буквы
  for i:=0 to 255 do write (i:4, ' - ',chr(i));
end.
```

Строка: 11    Столбец: 9

# Описание строк

- **Var**

**<имя\_строки>:string;**

**<имя\_строки>:string[n];**

n - максимально возможная длина строки. Если параметр опущен, то по умолчанию он считается равным 255.

- **Var**

**name : string[15];**

# Элемент строки

- К строке можно обратиться посимвольно.
- $i$ -й символ строки **s** записывается как **s[i]**.
- Например, если  $s = \text{'Мир'}$ ,  
то  $s[1] = \text{'М'}$ ,  
 $s[2] = \text{'и'}$ ,  
 $s[3] = \text{'р'}$

# Сравнение строк

- Над строками определены операции сравнения  $<$ ,  $>$ ,  $=$ ,  $<>$ ,  $<=$ ,  $>=$ , результат выполнения операции логический.
- Две строки сравниваются посимвольно, слева направо, по кодам символов. Если одна строка меньше другой по длине, при сравнении недостающие символы короткой строки заменяются символом с кодом 0.

# Сравнение строк

- В результате сравнения двух строк получается логическое значение (**true** или **false**).

Примеры:

- 'строка' <> 'строки' (true);
- 'Аbc' < 'abc' (true);
- 'год' > 'век'; (true);

# 1. Удаление в строке

**delete(s, poz, n);**

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

Пример:

- `s := 'алгоритм';`
- `delete(s,1,4);` // s будет равно ''

## 2. Вставка строки в строку

**insert(s1, s2, poz);**

вставляет строку **s1** в строку **s2**,  
начиная с позиции **poz** строки **s2**.

Пример:

- `s1 := 'до';`
- `s2 := 'ил';`
- `insert(s1,s2, 2);` //s будет равно ''

## 3. Копирование в строке

- **copy(s, poz, n);**

возвращает подстроку, выделенную из строки **s**, длиной **n** символов, начиная с позиции **poz**.

Пример:

- `s := 'корабль';`

- `s1 := copy(s, 3, 3); //s1 будет ' '`

## 4. Конкатенация (слияние)

- **concat(s1,s2,...,sn)**

выполняет сцепление строк в том порядке, в каком они указаны в списке параметров

```
c:=concat('кол', 'о' , 'бок');
```

результат c= 'колобок'

## 5. Длина строки

- **length(s);**

определяет текущую длину строки **s**.

Пример:

- `s='Программирование';`
- `length(s) = 16;`

## 6. Позиция символа в строке

- **pos(s1,s);**

возвращает номер первой позиции подстроки **s1** в исходной строке **s**. Если подстроки нет, то выдает 0.

Пример:

- `s := 'способ хранения';`
- `x1 := pos('ранения', s); //x1 будет 9`
- `x2 := pos('рана', s); //x2 будет 0`

## 7. Преобразования

**Str (x,s);** - преобразует числовое значение **x** в строковое **s**.

**Val (s,v,error);** - преобразует строковое значение **s** в числовое **v**, где значение переменной **error** равно позиции первого по порядку неправильного символа в **s**, если преобразование не произошло, или 0 в случае успешного преобразования.

# Проверка усвоения материала

Дана строка **s** = 'Удачного дня, ребята!'

- 1) `length(s)` = ?
- 2) `s1='же'+copy(s,15,6); s1` = ?
- 3) `delete(s,13,8); s` = ?
- 4) `insert(' вам',s,9); s` = ?

# Пример:

- В введенной строке подсчитать количество букв «О»,
- заменить букву «М» на «П»,
- удалить запятое.

```
var s:string;
kol,i:integer;
begin
writeln ('Введите строку: ');
readln(s);
kol:=0;
for i :=1 to length(s) do
begin
if s[i]='o' then inc(kol);
if s[i]='M' then s[i]:='П';
if s[i]=',' then delete(s,i,1);
end;
writeln('Кол-во букв o=',kol);
writeln('Получилась строка: ',s);
end.
```

# Домашнее задание

Добавить  
комментарии В  
программе,  
демонстрирующей  
основные функции  
и процедуры  
обработки строк

```
var s1, s2, s3, s4, str:string;  
l, k:integer;  
begin  
s1:='Ivanov';  
s2:='Ivan';  
str:= s1+' '+s2; //  
writeln(str); // str=  
l:=length(str); //  
writeln(l); // l=  
k:=pos('v', s1); //  
writeln(k); // k=  
s3:=copy(s1, 1, 4); //  
writeln(s3); // s3=  
s4:=copy(str, 1, pos(' ', str)-1); //  
writeln('Первое слово - ', s4); // s4=  
delete (str, 5, 2); //  
writeln(str); // str=  
end. |
```

# Задача.

Составить программу, вставляющую перед каждой цифрой 4 в исходной строке слово «да».  
(добавить комментарии)

```
uses crt;
var m,k,i:integer;
s,s1:string;
begin
Writeln('Введите строку: ->');
readln(s);
m:=0; //
for i:=1 to length(s)*2 do //
begin
if copy(s,i,1)='4' then //
begin
insert('da',s,i); //
i:=i+2; //
m:=m+1; //
end; end;
writeln;
write('Количество замен: ',m);
write('Новая строка: ',s);
end.
```

# Задача для самостоятельного решения

Подсчитать сумму всех цифр, которые встречаются в строке. (Указания: рассмотреть строку посимвольно, построить условие, проверяющее, является ли очередной символ числовым, при истинности данного условия преобразовать символ в число и добавить в сумму)