

0

1 курс

**ПЛАН – КОНСПЕКТ**  
проведения занятия по дисциплине «Информатика»

**Раздел 4. «Основы алгоритмизации и программирования.»**

**Тема 4.2.: «Структурированные типы данных.»**

**часть 6**

Подготовил: преподаватель  
В.Н. Борисов

Рязань 2023

## Вопросы занятия:

1. Строковые переменные и функции.
2. Составление программ с использованием строковых переменных и функций.
3. Взаимное преобразование символьных и числовых типов данных.

Время проведения занятия – 2 часа.

Первый вопрос: Строковые переменные и функции.

## Строковые константы и переменные

**Строка** – это последовательность символов. Каждый символ занимает 1 байт памяти (код ASCII). Количество символов в строке называется её длиной. Длина строки может находиться в диапазоне от 0 до 255.

**Строковая константа** – это последовательность символов, заключённая в апострофы.

**Пример:** 'Язык программирования Паскаль'

### Описание строковой переменной

**Var <имя>: string[N];**

**Пример:** Var name: string[20];

Длина строки

adr: string; {параметр длины может не указываться}

MyShared

**Строка** - это массив символов, т.е. элементов типа **char**. В Паскале строке соответствует тип данных **String**.

```
var Имя : string [Длина];
```

Если длина не указана, выделяется память под строку до 255 символов.

```
var s1:string;
```

строка 255  
СИМВОЛОВ

```
var s2:string[20];
```

строка 20  
СИМВОЛОВ

## Функции и процедуры работы со строками

Название функции.	Действие.	Тип аргумента (ов).	Тип результата.
<i>str(x, s)</i>	преобразует число $x$ в строку цифр $s$	$x$ – любой числовой тип, $s$ – string	Процедура, результат сохраняется в переменной $s$
<i>val(s,x,c)</i>	преобразует строку $s$ из цифр в число $x$ и возвращает код ошибки $c$	$s$ – string, $x$ – соответствующий числовой тип,	Процедура, результат сохраняется в переменных $x$ и $c$ .
<i>StrToInt(s)</i>	преобразует строку $s$ из цифр в целое число	$s$ – string	integer
<i>IntToStr(x)</i>	преобразует целое число $x$ в строку из цифр	$x$ – integer	string[10]
<i>StrToFloat(s)</i>	преобразует строку $s$ из цифр в действительное число	$s$ – string	real
<i>FloatToStr(x)</i>	Преобразование вещественного числа в строку	$x$ - real	string
<i>FloatToStrf(x,ffFixed,5,2)</i>	Преобразование вещественного числа в строку по заданному формату.	$x$ – real, выведет число занимающее 5 позиций, из которых 2 отводятся для дробной части числа	string

## В Паскале существуют специальные процедуры и функции:

**Length (st)** — значением функции является длина строковой переменной *st*.

**Copy (st , m, n)** — значением функции является подстрока из *n* символов, вырезанных из строки *st*, начиная с позиции, указанной параметром *m*.

**Delete (st , m , n)** — данная процедура удаляет *n* символов из строки *st*, начиная с позиции, указанной параметром *m*.

**Concat (st1 , st2, ... stn)** — соединение строк. Можно использовать конструкцию *st1+st2,+...+stn*.

## Строковые процедуры

**val(s,n,i)** — переводит строку *s* в число (вещественное или целое, согласно типу переменной *n*), если строка *s* не является изображением числа соответствующего типа по правилам Паскаля, то значение переменной *i* будет отлично от нуля, при удачной конвертации значение *i* равно нулю;

**str(i,s)** — переводит число в его строковое представление;

**delete(s,i,n)** — удаляет из строки *s* *n* символов, начиная с символа с номером *i*;

**insert(s1,s,i)** — вставляет в строку *s* подстроку *s1*, перед символом с номером *i*

## Удаление в строке

**delete(s,p,k);**

удаляет из строки **s** подстроку длиной **k** символов, начиная с символа под номером **p**.

Пример:

- `s := 'Система Turbo Pascal';`  
`delete(s,8,6);` {s будет равно 'Система Pascal'}

### Удаление и вставка

#### Удаление части строки:

```
s := '123456789';
Delete ( s, 3, 6 );
```

6 штук

```
'123456789'
```

```
'129'
```

строка  
меняется!

с 3-его символа

#### Вставка в строку:

```
s := '123456789';
Insert ( 'ABC', s, 3 );
```

начиная с 3-его символа

```
'12ABC3456789'
```

что  
вставляем

куда  
вставляем

```
Insert ( 'Q', s, 5 );
```

```
'12ABQC3456789'
```

## Операции со строками

```
var s, s1, s2: string;
```

### Запись нового значения:

```
s := 'Вася';
```

**Объединение:** добавить одну строку в конец другой.

```
s1 := 'Привет';
s2 := 'Вася';
s := s1 + ', ' + s2 + '!';
```

'Привет, Вася!'

**Подстрока:** выделить часть строки в другую строку.

```
s := '123456789';
s1 := Copy ( s, 3, 6 );
s2 := Copy ( s1, 2, 3 );
```

с 3-его символа

6 штук

'345678'

'456'

## Операции над строками

1. Присваивание строк:

```
s1 := 'ABCD';
s1 := s2;
s1 := 'A';
s1 := ''; {пустая строка}
```

2. Обращение к элементу:

```
s1[5] - прямое
s1[i] - косвенное
```

3. Конкатенация (сцепление) строк:

```
st := st + 'A';
st := 'A' + 'B';
```

4. Операции отношения – выполняется попарным сравнением кодов символов, результат определяется по отношению кодов первых различных символов:

```
b := s1 > s2;
'T' < 'Ta'
```

5. Ввод-вывод строк:

```
ReadLn(s1);
{строка вводится до Enter
```

или указанной длины}

```
WriteLn(s1);
```

## Стандартные функции для работы с символьными величинами.

Существует целый ряд задач, при решении которых необходимо уметь представить числовое данное в виде цепочки символов и, соответственно, выполнить обратное действие. В Турбо-Паскале существуют две процедуры, которые позволяют решить эти проблемы.

- Процедура `STR` преобразует число любого вещественного или целого типов в строку символов. `str(a, st)`; где:  
a — числовая переменная; st — строка символов. Пример:

```
var
  a: integer;
  b: real;
  st, st1: string;
.....
  a:=345;
  str(a, st);
  b:=12.6789;
  str(b, st1);  ⇒ значение st1=1.2678900000E+01 '
  str(b:2:2, st1); ⇒ значение st1='12.67'
```

### ***1. Операция сложения символьных величин.***

Операция сложения позволяет строить из двух символьных строк третью, состоящих из символов первой, за которой следуют символы второй. Обозначается эта операция знаком "+".

#### Пример 1.

Описываем строковые переменные.

```
var s1, s2, s3: string;
```

Присваиваемое значение строки заключается в апострофы. Присвоим первым двум следующие значения, а третья будет равна их склеиванию:

```
s1:= 'Тише воды, ';
```

```
s2:= 'ниже травы';
```

```
s3:=s1+''+s2;
```

Строка `s3` имеет значение 'Тише воды, ниже травы'.

## 2. Длина строки

Под длиной строки понимается количество введенных символов, но она не может превышать максимально возможной длины (в описательной части). Это значение можно определить при помощи функции, результат которой целое число, равное количеству символов.

### Пример 2.

```
s1:='12345';
s2:='Семеро одного не ждут';
k1:=Length(s1);
k2:=Length(s2).
```

В результате значения целых переменных будут равны: k1=5, k2=21.

## 3. Копирование

Функция `copy(str,n,m)` в *Turbo Pascal 7.0* – копируют *m* символов строки **str**, начиная с *n*-го символа, при этом исходная строка не меняется. Можно результат этой функции присваивать другой строке или сразу выводить его на экран.

### Пример 3.

```
s1:='паровоз';
s2:='123456';
s3:=copy(s1, 5, 3);
writeln(s3);
writeln(copy(s2, 3, 2));
```

Значения переменной s1='воз'. А на экране будут выведены следующие строки: воз и 34.

## 4. Удаление

В *Turbo Pascal 7.0* для этого используется процедура **Delete(str, n,m)**, которая вырезает из строки **str** *m* символов, начиная с *n*-го. таким образом сама строка изменяется.

### Пример 4.

Дан фрагмент программы:

```
s:='123456';
delete(s, 3, 2);
writeln(s);
```

После выполнения этих операторов из строки будут удалены два символа, начиная с третьего, то есть строка будет такой: *s = '1256'*.

## 5. Замена (Вставка)

В *Turbo Pascal 7.0* это можно сделать, применяя процедуру **Insert(s1,s2,n)** – вставка строки *s1* в строку *s2*, начиная с *n*-го символа, при этом первая строка остается такой же, как и была, а вторая получает новое значение.

### Пример 5.



```
s1:='34';
s2:='1256';
insert (s1, s2, 3);
```

В результате выполнения данной процедуры строка будет такой  $s2='123456'$ .

## 6. Числа и строки

Надо заметить, что число 25 и строка 25 – это не одно и то же. Для работы с числами и строками в Turbo Pascal 7.0 применяются две процедуры.

$Str(n,s1)$  – переводит числовое значение  $n$  в строковое и присваивает результат строке  $s1$ , причем можно переводить как целые числа, так и вещественные.

### Пример 6.1

```
n:=12;
str(n,s1);
- после выполнения s1='12';
```

Существует обратная операция, переводящая строковое значение в числовое.

Функция  $val(s, n, k)$  – переводит строковое значение в числовое, если данная строка действительно является записью числа (целого или вещественного), то значение  $k=0$ , а  $n$  – это число, иначе  $k$  будет равно номеру символа, в котором встречается первое нарушение записи числа  $n$ .

### Пример 6.2

```
val('1234',n,k) n=1234, k=0;
```

## 7. Функции преобразования типов

Иногда в программах возникает необходимость по коду определить символ и, наоборот, по символу определить его код. Для этого используют функцию: **CHR(x)**.

Эта функция возвращает символ, соответствующий ASCII-коду числа  $x$ .

### Пример 7.1

```
for i = 0 to 255 do
writeln( i, ' ', chr(i));
```

Для определения кода по символу используют функцию **ORD**.

### Пример 7.2

```
readln(s);
writeln(ord(s));
```

## Второй вопрос: Составление программ с использованием строковых переменных и функций.

### Пример 1.

Составить программу, определяющую по введенному с клавиатуры символу его код.

```

Programm prim1;
Var s: char;
Begin
Writeln('введите символ с клавиатуры');
Readln(s);
Writeln('код символа ',s,'=',ord(s));
Readln;
End.

```

### Пример 2.

В три символьные переменные F, I, O ввести свои фамилию, имя, отчество.  
Сформировать из этих данных строку S, содержащую ваши фамилию и инициалы.

```

Program prim2;
Var F, I, O, S : string;
Begin
Writeln('введите вашу фамилию');
Readln(F);
Writeln('введите ваше имя');
Readln(I);
Writeln('введите ваше отчество');
Readln(O);
S:=F+' '+copy(I,1,1)+'.'+copy(O,1,1)+'.';
Writeln('ваши реквизиты: ', S);
Readln;
End.

```

### Пример 3.

Определить сколько цифр содержится в записи произвольного натурального числа.

```

Program prim3;
Var s: string;
x, k: integer;
Begin
Writeln('введите число');
Readln(x);
Str( x, s);
k:=length(s);
Writeln('в числе ',k,' цифр');

```

```
Readln;
End.
```

#### Пример 4.

Переменные A и B содержат строки цифр. Найти сумму соответствующих чисел.

```
Program prim4;
Var A, B: string;
S, x, y, n, k: integer;
Begin
Writeln('введите первое число');
Readln(A);
Writeln('введите второе число');
Readln(B);
Val(A, x, n);
Val(B, y, k);
S:=x+y;
Writeln('сумма чисел равна ',S);
Readln;
End.
```

#### Пример 5.

Распечатать заданное слово в одной строке с разрядкой (пробел после каждой буквы).

```
Program prim5;
Var s, x: string;
i: integer;
Begin
Writeln('введите слово');
Readln(s);
x:='';
For i:=1 to length(s) do begin
x:=x+copy(s,i,1)+' ';
End;
Writeln('получилось слово ', x);
Readln;
End.
```

#### Пример 6.

Составить программу подсчета количества вхождений буквы "а" в заданном тексте.

```

Program prim6;
Var s: string;
i, k: integer;
Begin
Writeln('введите текст');
Readln(s);
k:=0;
for i:=1 to length(s) do begin
if copy(s, i, 1)='a' then k:=k+1
end;
Writeln('количество букв "а" в тексте равно ', k);
Readln;
End.

```

### Пример 7.

Определить, какое из двух исходных слов длиннее и насколько.

```

Program prim7;
Var s1, s2: string;
L1, l2: integer;
Begin
Writeln('введите первое слово');
Readln(s1);
Writeln('введите второе слово');
Readln(s2);
l1:=length(s1);
l2:=length(s2);
if l1>l2 then writeln('первое слово длиннее второго на ',l1-l2,' символов')
else
if l1=l2 then writeln('слова одинаковой длины')
else writeln('первое слово длиннее второго на ',l2-l1,' символов');
Readln;
End.

```

**Задача 2. Составить программу которая считает сколько русских и сколько английских букв в строке введенной с клавиатуры.**

```

write('введи строку'); readln(a);
s:=0;
s1:=0;
for i:=1 to length(a)
begin
  b:=ord(a[i]);
  if (b in [65..90])
  (b in [97..122]) then s:=s+1;
  if (a[i] in ['А'..'Я']) or
  (a[i] in ['а'..'я']) then s1:=s1+1;
end;
writeln(s, '-английских букв');
writeln(s1, '-русских букв');

```

Код символа как  
Символ, как  
элемент множества

MyShared

Получить из слова «информатика»  
слова «тик», «форма», «форт».

```

program zadacha1;
var p,slovo1,slovo2, slovo3:string;
begin
p:='Информатика';
slovo1:=copy (p,3,5);
slovo2:=copy (p,8,3);
slovo3:=copy(p,3,3)+ copy(p,8,1);
writeln (slovo1,', ', slovo2,', ', slovo3);
end.

```

форма, тик, форт

## 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.
```

Третий вопрос: Взаимное преобразование символьных и числовых типов данных.

## Преобразование типов

**Str(x)**

Преобразует  
численное  
значение x в его  
строковое  
представление

**ord(x)**

возвращает  
порядковый номер  
для параметра x  
порядкового типа