Символы и строки в Pascal

Определения

- ***Символ*** — это буква, цифра или какой-либо знак.
- ***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

Тип CHAR

Символьный или строковый или литерный.

Его значениями являются отдельные символы: буквы, цифры, знаки. Символьные константы заключаются в кавычки, например, ‘A’, ’B’, ’C’, ‘4’,‘7’, ‘ ‘(пробел).

Символьные переменные описываются предложением:

Var имя переменной: char;

Символ может быть записан, ограниченный апострофами ***(‘a’)*** или записан в коде ***(#97)***

Стандартные символьные функции

В Паскале имеются стандартные символьные функции:

- CHR(N) – возвращает в программу символ с кодом N,
- ORD(S) – возвращает код символа S,
- PRED(S) –возвращает предыдущий символ
- SUCC(S) – возвращает следующий символ

ПРИМЕРЫ:

- CHR(128) = Б
- ORD(‘:’) = 58
- PRED(‘Б’) = А
- SUCC(‘Г’) =Д

ASCII

Каждый символ имеет свой уникальный двоичный код. Коды всех символов сведены в таблицу. Первая половина таблицы стала международным стандартом, который называется ASCII – American Standard Code Information Interchange (читается «аски код») в ней кроме прочего содержится латинский алфавит, вторая имеет разные варианты для разных языков. Кириллица (русский алфавит) имеет несколько стандартов. В Паскале используется стандарт КОИ-8.

ПРИМЕР 1 Составить программу, по которой компьютер многократно вычисляет сумму A+B при различных значениях A и B. в конце каждого этапа появляется запрос о продолжении или прекращении вычислений: «Завершить программу? (Д/Н)»

```
Var A,B:real; C : char;
Begin
    repeat
        Write('Введите два числа'); Readln(a,b);
        Writeln(a+b:0:2);
        Writeln('Завершить программу?(Д/Н)');
        Readln(c);
    Until   c='Д'; {программа завершит работу если будет введено Д}
Readln;
End.
```

Тип STRING

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

К строке можно обратиться посимвольно.

i-й символ строки **s** записывается как **s[i]**.

Например, если s = ‘Мир',

то s[1] = это ‘М',

s[2] = это ‘и',

s[3] = это ‘р'

Операции со строками

1. **Операция объединения строк (конкатенация).** Обозначается эта операция знаком + (но это не сложение!)

st:= 'abcd ';

stl:='efk';

st2:=st+stl;

значение **st2 — 'abcdefk'**

Примечание: если длина результирующей строки превысит допустимую (255) длину, то "лишние" символы отбрасываются.

Аналогичную операцию выполняет функция concat(s1,s2,...,sn)

st:= 'abcd ';

stl:='efk';

st2:=concat(st+stl);

значение **st2 — 'abcdefk'**

Операции со строками

2. Операции сравнения =, >=, >, <>, <, <=

Сравнивать можно строки разной длины.

Две строки сравниваются посимвольно, слева направо, по кодам символов ASCII .

Если одна строка меньше другой по длине, при сравнении недостающие символы короткой строки заменяются символом с кодом 0.

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

Примеры:

‘строка’<>’строки’ ; // true

‘Abc’<’abc’; // true

‘год’<’век’; // false

Тип данных String используется при обработке текстов, а это означает, что нам необходимо уметь:

1. *копировать часть строки;*
2. *удалить часть строки;*
3. *вставлять подстроку (т.е. часть строки) в данную строку;*
4. *осуществлять поиск подстроки (т.е. часть строки) в данной строке.*

Для реализации этих операций в Паскале существуют стандартные процедуры и функции.

Стандартные процедуры и функции

1. **Копирование** части строки - **функция**

   copy (st, n, k);

где **st**—исходная строка,

n—начальная позиция,

k — количество символов, которые мы будем копировать.

Пример:

```
var
st:='abcdef;
stl:=copy(st,2,3);
```

Значение строки stl—**'bcd'**.

2. Удаление части строки — **процедура**

delete(st,n,k);

где st— исходная строка,

n—начальная позиция (с какого символа начинаем удалять),

k—количество удаляемых символов.

Пример:

var st:='abcdef' {st[0]=6;}

delete(st,2,3); . .

Значение строки st—**'aef'**.

Внимание! При удалении части строки ее длина автоматически уменьшается, **st[0]=3**.

3. Вставить подстроку в данную строку **процедура**
insert(stl,st,n);

где stl —подстрока,

st—исходная строка,

n—позиция символа, с которого начинается вставка.

Пример:

Пусть st—исходная строка, stl—подстрока.

st:='?*-+/ab'; stl:='!%';

insert(stl,st,2);

Значение строки st—"?**!%***-+/ab".

Можно не вводить имя переменной для подстроки, указав значение переменной явно, т.е. insert('!%', st, 2);

4. Поиск подстроки в строке (первое вхождение) - функция

k:=pos(stl,st);

где stl—подстрока,

st—строка,

k—позиция **первого** вхождения подстроки, либо 0, если подстрока отсутствует.

Пример:

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.	23.	24.	25.
st:=	**‘И**	**в**	**а**	**н**	**о**	**в**		**А**	**л**	**е**	**к**	**с**	**а**	**н**	**д**	**р**		**г**	**.**	**М**	**о**	**с**	**к**	**в**	**а’**

Будем искать подстроку "г.Москва" в строке st,

k:=pos('г.Москва',st);

Ответ: k=18.

Внимание! Повторим еще раз:

1. Значение функции pos—переменная целого типа.
2. Функция pos ищет **первое** вхождение подстроки в строку.

Строка типа string помимо буквенных и других символов может содержать и цифры.

Пример:

```
 var st, stl, st2: string;
      a, b, c: integer;
 st:='123';
stl:='254‘;
a:=123;
b:=254;
```

В st имеем набор символов, но не числа! Представление в памяти переменной st, объявленной как string, отличается от представления переменных a и b.

Числовые переменные переводятся в двоичную систему счисления и, в зависимости от типа (целого или вещественного), хранятся в памяти ЭВМ (2, 4 или больше байт). В строке символов каждому символу соответствует свой код. *Один символ занимает 1 байт.*

Сравните:

st2:=st+stl;

значение st2 - **'l23254'**

c:=a+b;

значение c=**377**;

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

1. 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, stl: string;
……
  a:=345;
  str(a,st);
  b:=l2.6789;
  str(b, stl); ⇒ значение stl=1.2678900000E+01'
  str(b:2:2,stl); ⇒ значение stl='12.67'
```

2. **Процедура VAL** преобразует строку, содержащую цифры в число, вещественное или целое.

val(st,a,code);

где: a — числовая переменная;

st — строка;

code — переменная целого типа, по значению которой можно определить, успешно или нет прошел перевод.

Если перевод из строки символов в число прошел успешно, значение переменной code равно 0.

```
var
 a: integer;
 b: real;
 code: integer;
 st: string;
 ……
 st:='345';
```

val(st,a,code); ⇒ значение переменной a равно 345.

st:='12.6789'; val(st,b,code); ⇒ значение переменной b равно 1.2678900000E+01.