8. СИМВОЛНИ НИЗОВЕ

В глава 1 беше показано как се дефинират константи и променливи от тип символен низ в Турбо Паскал. В глава 2 пък беше показано как се въвеждат и извеждат данни от този тип.

Тук ще разгледаме други основни операции със символни низове.

8.1. Операция присвояване

На променливите от тип *символен низ* могат да се присвояват изрази от същия тип. Нека в описателната част на дадена програма са декларирани следните променливи от тип символен низ:

```
Var
  Str1, Str2 : string[4];
  StrA : string[10];
  StrX : string;
```

И нека в изпълнимата част да присъстват следните оператори:

```
StrX := 'Русе''93';
StrA := StrX;
Str1 := StrX;
```

След изпълнение на първия изпълним оператор, променливата StrX ще има стойност Русе'93. След изпълнение на втория и StrA ще има същата стойност. След изпълнение на третия обаче Str1 ще има стойност Русе, защото тя е декларирана с дължина 4 символа и не разполага с място за символите след четвъртия.

8.2. Достъп до отделни символи

Символен низ и масив от символен тип, дефинирани например като

*StrY : **string**[20] u StrY : **array**[1..20] **of** char*

са аналогични по отношение на достъпа до отделните символи. Поради това i-ия символ от символния низ, респективно i-тия елемент на масива от символен тип, имат съставно име StrY[i] и може да участва във всички операции, разрешени за типа **char**.

```
Q := StrX[2];
Writeln(StrX[3]); . . .
```

8.3. Сравняване на символни низове

За величини от тип символен низ са допустими всички шест операции за сравняване. Сравняването е посимволно - сравняват се първите символи на двата низа, след това вторите, третите и т.н. до срещане на несъвпадащи символи или до изчерпване на някой от низовете. Резултатът от сравняването е:

- символните низове са равни (еднакви), ако двата низа са с еднаква дължина и всичките им символи съвпадат, например 'Иван' = 'Иван';
- левият символен низ е по-малък от десния, когато сравняването е прекратено поради изчерпване на левия низ без да са срещнати нееднакви символи или когато се срещнат нееднакви символи и този от нееднаквите символи, който принадлежи на левия низ е с по-малък код (независимо от дължините на двата низа), например 'Иван' < 'Иванка' и 'Иванка' < 'Иванчо';
- левият символен низ е по-голям от десния, когато сравняването е

прекратено поради изчерпване на десния низ без да са срещнати нееднакви символи или когато се срещнат нееднакви символи и този от нееднаквите символи, който принадлежи на левия низ е с по-голям код (независимо от дължините на двата низа), например 'Иванка' > 'Иван' и 'Иванчо' > 'Иванка';

8.4. Процедури и функции за операции със символни низове

В Турбо Паскал са включени следните процедури и функции за работа със символни низове:

а) Функция за сливане на символни низове

Function Concat(S1, S2, . . ., Sn : **string**) : **string**;

Тя създава нов низ, който се получава при последователното слепване на низовете S1, S2, . . ., Sn. Например след изпълнение на оператора

```
NovNiz := Concat('Turbo',' Pascal');
```

NovNiz ще има стойност Turbo Pascal.

Същият резултат може да се получи и чрез операцията слепване, за която се използва символът +. Например

```
NovNiz := 'Turbo' + ' Pascal';
```

б) Процедура за включване на символен низ в друг символен низ

Procedure Insert(S1 : **string**; **Var** S2 : **string**; N : integer);

Тя вмъква символния низ S1 между N-1 - ия и N - ия символи на символния низ S2.

След изпълнение на операторите

```
S1 := 'ПП';
S2 := 'ABCDEF;
Insert(S1,S2,3);
```

символният низ S2 е със стойност АВППCDEF.

в) Процедура за изтриване на част от символен низ

Procedure Delete(**Var** S1 : **string**; N,Br : integer);

Тя изтрива Br броя символи от символния низ S1, като започва от N-тия символ.

Например след изпълнение на операторите

```
S1 := 'ABCDEF';
Delete(S1,3,2);
```

новата стойност на S1 ще бъде ABEF.

г) Функция за копиране на част от символен низ

Function Copy(S1 : **string**; N, Br : integer) : **string**;

Тя връща като резултат нов низ, който включва Br броя последователни символи от низа S1, започвайки от N-тия символ.

Например след изпълнение на операторите

```
EGN := '8105245366';
Copy(S1,3,2);
```

където EGN е единен граждански номер на дадено лице, функцията Copy ще върне символния низ ‘05’.

д) Функция за проверка за присъствие на символен низ в друг символен низ

Function Pos(S1,S2 : **string**) : integer;

Тя проверява дали низа S1 присъства в низа S2 и връща:

- когато S1 присъства в S2 - цяло число, което показва номера на символа в S2, от който започва присъствието;

- когато S1 не присъства в S2 - нула.

Например след изпълнение на операторите-обръщения

Pos(’524’, EGN) и Pos(’123’ ,EGN),

функцията Pos ще върне съответно 4 и 0, ако EGN има стойност '8105245366';

е) Функция за определяне дължината (броя на символите) на символен низ

Function Length(S : **string**) : integer;

Тя връща като целочислена стойност дължината на символния низ S. Например след изпълнение на оператора-обръщение

Length(EGN),

функцията Length ще върне числото 10.

ж) Процедура за преобразуване на символен низ в числена стойност

Procedure Val(S : **string**; **Var** X : *числов тип*; **Var** K : integer);

Тя превръща символния низ S в числена стойност и я присвоява на променливата X, а на K присвоява стойност нула при следните условия:

- когато X е от целочислен тип и S съдържа само цифри, евентуално предшествани от знак + или -;

- когато X е от реален тип и S съдържа символите, допустими при изписване на реална константа, например когато S е '1.234', '1234', '1234E2' или '1.234E4'.

Ако тези условия са нарушени, на K се присвоява номера на първия елемент от низа, който ги нарушава.

В следващите две таблици са дадени примери за превръщане символен низ съответно в цяло число и реално число.

Оператор-обръщение	Стойност на M след обръщението	k
Val (‘125’,M,k)	125	0
Val (‘-125’,M,k)	-125	0
Val (‘12.5’,M,k)	Неопределена	3

Оператор-обръщение	Стойност на R след обръщението	k
Val(‘1.234’,R,k)	1.2345	0
Val(‘-1.234’,R,k)	-1.234	0
Val(‘1234’,R,k)	1234.0	0
Val(‘1234E-2’,R,k)	12.34	0
Val(‘-1.234E+4’,R,k)	-12340.0	0
Val(‘123,4E-2’,R,k)	Неопределена	4

Например, ako i и k са целочислени, то след изпълнение на оператора Val(‘125’,i,k), i и k ще имат съответно стойности 125 и 0, а след изпълнение на оператора Val(‘12.5’,i,k), i ще има неопределена стойност, а стойността на k ще е 3, защото третият символ в низа е недопустим за целочислен тип.

Ako P е от реален тип, а k от целочислен, то след изпълнение на оператора Val(‘12.5’,P,k), i и k ще имат съответно стойности 12.5 и 0, а след изпълнение на оператора Val(‘12,5’,P,k), i ще има неопределена стойност, а стойността на k ще е 3, защото третият символ в низа е недопустим за реален тип.

з) Процедура за преобразуване на число в символен низ

Procedure Str(X : *числов тип*; **Var** S : **string**);

Тя преобразува стойността на X в символен низ при следните условия (аналогично на операторите Write и Writeln):

- ако X е от целочислен тип, цифрите му се разполагат в първите позиции на низа, но ако е указана дължина на полето, например Str(X:8,S), цифрите се разполагат в дясната част на участъка от първите 8 позиции от низа.

- ако X е от реален тип, числото се представя с плаваща десетична точка и така се разполага в символния низ. Може да се укаже общата дължина на полето, т.е. Str(X:8,S) и тогава от нея се определя дължината на мантисата на числото. Може да се укаже обща дължина на полето и участъка от него за дробната част, т.е. Str(X:8:3,S) и тогава символният низ съдържа числото изписано с фиксирана десетична точка.

Няколко примера се съдържат в следната таблица:

Оператор-обръщение	Символен низ S след обръщението
Str(150,S)	‘150’
Str(150:8,S)	‘ 150’
Str(-150:8,S)	‘ -150’
Str(12.5,S)	‘ 1.2500000000E+1’
Str(12.5:10,S)	‘ 1.2500E+1’
Str(12.5:8:3,S)	‘ 12.500’
Str(-12.5:8:3,S)	‘ -12.500’
Str(-12.56E-2:8:3,S)	‘ -0.126’

Програма 8.1. Програма с подпрограма за търсене на подниз в даден низ и заместване на подниза с друг низ.

```
Var
  Str, StrTrs,StrZam:string;
Procedure TrsZamStr(Var S:string;S1,S2:string);
{ S -низ, в който се търси;
  S1-низ, който се търси като подниз в S;
  S2-низ, който замества намерения подниз}
  Var
    i,n:byte;
  Begin
    n:=Length(S1);
    Repeat
      i:=Pos(S1,S);
      If i>0
        then begin
               Delete(S,i,n);
               Insert(S2,S,i)
             end
    until i=0
  End;
Begin
  Repeat
    Write('Въведете низ или Enter за край: ');Readln(Str);
    If Str<>''
    then begin
```

```
    Write('Въведете търсен низ: ');Readln(StrTrs);
    Write('Въведете заместващ низ: ');Readln(StrZam);
    Writeln('Зададен низ: ', Str);
    TrsZamStr(Str,StrTrs,StrZam);
    Writeln('Коригиран низ: ',Str);
  end;
  Writeln
until Str='';
End.
```

Програма 8.2. Програма с подпрограма за анализ дали даден низ може да е име на обект от програма.

```
Var
  Str:string;
Function ImeObekt(S:string):boolean;
  Var
    i,L:byte;
    Ime:boolean;
  Begin
    L:=Length(S);
    i:=1;
    Ime:=(S[i]>='A') and (S[i]<='Z') or (S[i]>='a') and (S[i]<='z');
    While (i<L) and Ime do
    begin
      i:=i+1;
      Ime:=(S[i]>='A') and (S[i]<='Z') or (S[i]>='a') and (S[i]<='z') or
                                                    (S[i]>='0') and (S[i]<='9')
    end;
    ImeObekt:=Ime
  End;
Begin
  Repeat
    Write('Въведете низ или Enter за край: ');Readln(Str);
    If Str<>''
    then if ImeObekt(Str)
           then Writeln('Низът е име на обект')
           else Writeln('Низът не е име на обект');
    Writeln
  until Str='';
End.
```

Програма 8.3. Програма с подпрограма за анализ дали даден низ е цяло число.

```
Var
  Str:string;
{Подпрограма за анализ дали даден низ е цяло число}
Function Cialo(S:string):boolean;
  Var
    i,L:byte; Cl:boolean;
  Begin
    L:=Length(S); i:=0;
    Repeat          {Пропускане на празните символи в началото на низа }
      i:=i+1
    until (S[i]<>' ') or (i=L);
    If S[i] = ' '
```

```
    then Cialo:=false
    else begin
       If(S[i]='+') or (S[i]='-') then i:=i+1;
       If i > L
         then Cialo:=false
         else begin
            Repeat
              Cl:=(S[i]>='0') and (S[i]<='9');
              i:=i+1
            until not Cl or (i>L);
            Cialo:=Cl
         end
      end
  End;
Begin
  Repeat
    Write('Въведете низ или Enter за край: ');Readln(Str);
    If Str<>''
      then if Cialo(Str)
            then Writeln('Низът е цяло число')
            else Writeln('Низът не е цяло число');
    Writeln
  until Str='';
End.
```