

Funkcie a procedúry definované pre reťazcové konštanty

Length – používa sa na zistenie dĺžky reťazca.

Deklarácia: **Length(s:string):integer;**

Funkcia vráti skutočnú dĺžku reťazca (počet znakov) **s**.

Príklad:

s := slovo;

d := Length (s);

Výsledok: v premennej **d** bude hodnota **5**.

Copy – používa sa na kopírovanie reťazca.

Deklarácia: **Copy (s:string;cislo,pocet:integer):string;**

Funkcia vráti podreťazec z reťazca **s**. parameter **cislo** udáva pozíciu znaku, od ktorého sa začne kopírovať. Parameter **pocet** udáva počet znakov, čiže koľko sa ich skopíruje z reťazca **s**, počínajúc znakom, ktorý bol udaný ako **cislo**. Ak je hodnota **pocet** väčšia ako zostávajúca dĺžka reťazca, skopíruje sa len zvyšok reťazca.

Príklad:

s := ABCDE;

z := Copy (s,2,3);

Výsledok: v premennej **z** bude uložený reťazec **BCD**.

Concat – používa sa na spájanie reťazcov.

Deklarácia: **Concat(s1,s2,...,sN:string):string;**

Funkcia spojí reťazce s1, s2, ..., sN a vráti reťazec, ktorý je výsledkom tohto spojenia. Ak je výsledok dlhší ako 255 znakov, reťazec skráti na dĺžku 255 znakov.

Príklad:

s1 := ABC;

s2 := DEF;

z := Concat (s1,s2);

Výsledok: v premennej **z** bude reťazec **ABCDEF**.

Pos – používa sa na vyhľadávanie podreťazca v inom reťazci.

Deklarácia: **Pos(podr:string;s:string):byte;**

Funkcia vyhľadá podreťazec **podr** v reťazci **s**. funkcia vráti celočíselnú hodnotu, ktorá udáva, na ktorej pozícii sa nachádza prvý znak podreťazca **podr** v reťazci **s**. Ak sa podreťazec nenájde, funkcia vráti hodnotu „nula“.

Príklad:

ss := BCD;

s := ABCDE;

m := Pos(ss, s);

Výsledok: v premennej **m** bude číslo **2**.

Reťazcové premenné

Delete – používa sa na mazanie znakov v reťazci.

Deklarácia: **Delete**(var s:string; cislo, pocet:integer);

Procedúra vymaže podreťazec z reťazca **s**. Hodnota parametra **cislo** udáva, od ktorého znaku (včítane) sa začne mazanie. Parameter **pocet** udáva, koľko znakov sa má vymazať.

Príklad:

```
s := ABCDE;
```

```
Delete(s,2,3);
```

Výsledok: v premennej **s** bude reťazec **AE**.

Insert – používa sa na vkladanie reťazca do iného existujúceho reťazca.

Deklarácia: **Insert**(ret:string; var s:string; cislo:integer);

Procedúra vkladá reťazec **ret** do reťazca **s**. hodnota parametra **cislo** udáva pozíciu, od ktorej sa začne vkladanie reťazca **ret**.

Príklad:

```
s :=AB;
```

```
ss := XY;
```

```
Insert (ss, s, 2);
```

Výsledok: v premennej **s** bude reťazec **AXYB**.

Str - používa sa na konvertovanie numerickej hodnoty na reťazec.

Deklarácia: **Str**(x[:sirka[:pdm]];var s:string);

Procedúra konvertuje (mení) numerickú hodnotu **x** na reťazec znakov **s**.

Možno voliť parameter **sirka**, ktorým sa špecifikuje celkový počet znakov, ktorý zaberie výsledný reťazec. Parameter **pdm** špecifikuje počet desatinných miest vyhradených pre desatinnú časť numerickej hodnoty **x**. Ak sú voliteľné parametre zadané, ale numerická hodnota vo svojej numerickej reprezentácii zaberie väčší počet miest ako sú zadané voliteľnými parametrami, hodnoty voliteľných parametrov sú ignorované.

Príklad:

```
i := 10.3;
```

```
Str(i,s); { konvertuje sa číslo i na reťazec s }
```

```
OutTextXY(50, 50, s);
```

Výsledok: na grafickú obrazovku sa vypíše číslo **i** ako **reťazec**.

Val – používa sa na konvertovanie hodnoty typu string na zodpovedajúcu numerickú hodnotu.

Deklarácia: **Val**(s:string; var v,c:integer);

Procedúra konvertuje hodnotu typu **string** na zodpovedajúcu numerickú hodnotu. Syntax reťazca **s** musí byť celočíselný, alebo reálny typ (pred číslom môžu byť uvedené aj medzery). Premenná **v** musí byť celočíselného, alebo reálneho typu. Premenná **c** je pri úspešnom konvertovaní nastavená na nulu, v opačnom prípade ukazuje index znaku, pri ktorom došlo ku chybe.

Príklad:

```
s := 102;
```

```
Val(s,v,c);
```

```
Writeln(v);
```

Výsledok: v premennej **v** je číslo **102** a je typu **integer**.