

Reťazce znakov

Reťazec znakov je postupnosť znakov, ktorá tvorí jeden celok, napr. veta „Ahoj, svet!“

Vlastnosti:

- Ø správa sa ako pole znakov
- Ø sústavne udržuje počet znakov vložených do reťazca, tzv. aktuálnu veľkosť reťazca
- Ø je deklarované množstvo funkcií a procedúr pre prácu s reťazcami
- Ø procedúry `read`, `readln`, `write` a `writeln` umožňujú priamo vstupné a výstupné operácie

Deklarácia:

(zoznam premenných) : **string**;

napr.: `ret,text1,názov : string`;

Práca s reťazcom:

`ret:=’Turbo’`;

`text1:=’’`; {do premennej `text1` priradzujeme prázdny reťazec}

`nazov:=ret + text1`; {spojenie reťazcov do jedného reťazca}

Operácie pre prácu s reťazcami:

- Ø funkcia **Concat(reťazec1, reťazec2, ..., reťazecN)** – spojí reťazce do jedného celku, výsledkom je hodnota typu `string`. Ak je súčet dĺžok všetkých reťazcov väčší ako deklarovaná dĺžka premennej, do ktorej výsledok ukladáme, reťazec sa automaticky skráti na túto dĺžku.
- Ø funkcia **Length(reťazec)** – zistí aktuálnu dĺžku reťazca, výsledkom je hodnota typu `byte`.
- Ø funkcia **Copy(reťazec, od_pozicie, pocet_znakov)** – vyberie z reťazca časť dĺžky `pocet_znakov` od stanovenej pozície. Výsledkom je hodnota typu `string`.
- Ø funkcia **Pos(hľadaný_podreťazec, reťazec)** – vráti pozíciu, od ktorej sa v reťazci nachádza `hľadaný_podreťazec`. Táto hodnota je typu `byte`. Ak sa hľadaný podreťazec v reťazci nenachádza, funkcia vráti 0.
- Ø procedúra **Insert(vkladany_reťazec, reťazec, kam)** – vloží `vkladany_reťazec` do druhého reťazca od pozície `kam`. Výsledná hodnota

je uložená v premennej *reťazec*. Ak je v parametri *kam* hodnota väčšia ako je aktuálna dĺžka *reťazca* vloží sa *vkladaný_reťazec* za *reťazec*.

- procedúra **Delete(*reťazec*, *odkial'*, *koľko_znakov*)** – odstráni časť reťazca od pozície *odkial'* o dĺžke *koľko_znakov*. Výsledná hodnota je uložená v premennej *reťazec*.
- procedúra **Str(*číselný_výraz*, *reťazec*)** – prevedie výraz ľubovoľného číselného údajového typu na reťazec. Požadovanú dĺžku reťazca a počet desatinných miest určíme podobným spôsobom ako u procedúr *write* a *writeln*.
- procedúra **Val(*reťazec*, *číslo*, *pozícia_chyby*)** – z reťazca vytvorí číselnú hodnotu. Rovnaký proces prevodu vykonávajú procedúry *read* a *readln*. Medzery pred začiatkom zápisu čísla nevadia, za zápisom čísla nesmú byť žiadne ďalšie znaky. Ak *reťazec* neodpovedá zápisu čísla, resp. prevedená číselná hodnota nie je kompatibilná s údajovým typom skutočnej premennej *číslo*, prevod sa neuskutoční a do parametra *pozícia_chyby* sa vloží číslo, ktoré odpovedá prvej pozícii reťazca, kde sa nachádza neprípustný znak v číselnom zápise. Ak je hodnota parametra *pozícia_chyby* 0, prevod prebehol bezchybne.

Príklad 1:

Zostavte program, ktorý načíta vetu a vypíše počet znakov vety a tiež ju vypíše odzadu dopredu.

Riešenie:

```
uses Crt;
var r:string;
    n,i:integer;
begin
  clrscr;
  writeln('Zadaj reťazec znakov');
  readln(r);
  n:=length(r);
  writeln;
  writeln('Pocet znakov reťazca je ',n);
  writeln('Tento reťazec odzadu dopredu je ');
  for i:=n downto 1 do write(r[i]);
  repeat until keypressed;
end.
```