

Delphi naredbe za rad sa string tipovima podataka

Da bi se pristigli rezultati merenja mogli obraditi, potrebno je iz primljenog stringa izdvojiti svaku od merenih veličina i pretvoriti je u realan broj. Za to se mogu iskoristiti sledeće Delphi naredbe za manipulaciju sa string tipovima podataka i konverziju u cele i realne brojeve:

```
function Pos(Substr: string; S: string): Integer;
```

funkcija Pos vraća poziciju prvog karaktera podstringa Substr unutar stringa S. Funkcija Pos je osetljiva na mala i velika slova. Ukoliko traženi podstring nije pronađen, Pos kao rezultat vraća nulu.

Nakon:

```
S := 'Bole je moj drug!';  
I := Pos('drug', S);
```

vrednost promenljive I biće 13.

```
function Copy(S: string; Index, Count: Integer): string;
```

funkcija Copy vraća podstring stringa S, koji počinje od pozicije Index i ima Count karaktera. Ako je Index veće od dužine stringa S, funkcija Copy vraća prazan string. Ako parametar Count zahteva više karaktera nego što je u stringu S dostupno, funkcija Copy vratiće samo podstring od pozicije Index do kraja stringa.

Nakon:

```
S := 'Bole je moj drug!';  
S1 := Copy(S, 1, 6);
```

vrednost promenljive S1 biće 'Bole j'.

```
function Trim(const S: string): string; overload;
```

funkcija Trim uklanja sve vodeće i završne prazne (space) karaktere, kao i kontrolne karaktere iz stringa S i vraća string koji je rezultat ove konverzije. Višestruki space karakteri koji se nalaze unutar stringa S neće biti uklonjeni.

Nakon:

```
S := '  Boles je moj drug!  ';  
S1 := Trim(S);
```

vrednost promenljive S1 biće 'Boles je moj drug!'.

```
procedure Delete(var S: string; Index, Count: Integer);
```

procedura Delete uklanja Count karaktera iz stringa Spočev od pozicije Index. Ako je parametar Index veći od dužine stringa S ili je manji od 1, iz stringa S neće biti obrisan nijedan karakter. Ako parametar Count zahteva uklanjanje više karaktera iz stringa S nego što ih preostaje počev od pozicije Index, biće obrisani svi karakteri od Index do kraja stringa. Ako je Count manje ili jednako od nule nijedan karakter neće biti obrisan..

Nakon:

```
S := 'Bole je moj drug!';  
Delete(S, 9, 4);
```

vrednost promenljive S biće 'Bole je drug!'.

```
procedure Insert(Source: string; var S: string; Index: Integer);
```

Insert ubacuje string Source u S na poziciju Index. Ako je Index manje od 1, Source se ubacuje na prvu poziciju. Ako je Index veće od dužine stringa S, Source se ubacuje na kraj stringa S. Ukoliko je Source prazan string Insert nema nikakvog efekta.

Nakon:

```
S := 'Bole je moj drug!';  
Insert('ni', S, 5);
```

vrednost promenljive S biće 'Bole nije moj drug!'.

```
function IntToStr(const I: Integer): String;
```

IntToStr konvertuje ceo broj u string koji sadrži decimalnu interpretaciju tog broja.

Nakon:

```
I := 15;  
S := IntToStr(I);
```

vrednost promenljive S biće '15'.

```
function StrToInt(const I: Integer): String;
```

StrToInt konvertuje string S u ceo broj. S mora predstavljati neki od važećih načina zapisivanja celih brojeva u decimalnoj ili heksadecimalnoj notaciji.

U slučaju da se string *S* ne može konvertovati u ceo broj, aktivira se greška koja prekida izvršavanje programa, pa se prilikom konverzije stringa u broj moraju koristiti mehanizmi za kontrolu obrade grešaka (**try**, **except**, **end**).

Nakon:

```
S := '15';  
I := StrToInt(S);
```

vrednost promenljive *I* biće 15.

```
function FloatToStr(Value: Extended): string;
```

`FloatToStr` konvertuje floating-point vrednost *Value* u odgovarajuću string reprezentaciju. Konverzija koristi opšti format broja sa 15 značajnih cifara. Za bolju kontrolu formatiranja broja preporučuje se upotreba funkcije `FloatToStrF`.

Nakon:

```
X := 0.00003457;  
S := FloatToStr(X);
```

vrednost promenljive *S* biće '3.457e-5'.

```
function FloatToStrF(Value: Extended; Format: TFloatFormat; Precision,  
                    Digits: Integer): string;
```

`FloatToStrF` konvertuje realan broj *Value* u njegovu string reprezentaciju. Parametar `Precision` određuje preciznost sa kojom je dat parametar *Value* i može imati vrednost 7 ili manju za tip `Single`, 15 ili manju za tip `Double` i 18 ili manju za tip `Extended`. Parametri `Digits` i `Format` zajedno kontrolišu kako će broj biti konvertovan u string. Najvažnija vrednost parametra `Format` je `ffFixed` za prikazivanje brojeva u fiksnom zarezu.

Za sve formate, karakteri koji će se koristiti za odvajanje hiljada i decimalnih mesta definisani su trenutnim vrednostima globalnih promenljivih `ThousandSeparator` i `DecimalSeparator`.

Ako data vrednost nije validan broj rezultat je string 'NAN' (not-a-number). ako je *Value* na maksimalan pozitivan broj rezultujući string je 'INF' (infinity), a koje maksimalan negativan broj, onda je rezultujući string '-INF'.

Nakon:

```
X := 3.1415296;  
S := FloatToStrF(X, ffFixed, 15, 5);
```

vrednost promenljive *S* biće '3.14153'.

```
function StrToFloat(const S: string): Extended;
```

Funkcija StrToFloat konvertuje string S u realan broj. String S se mora sastojati samo od opcionog predznaka (+ ili -), niza cifara sa opcionim decimalnim zarezom i opcionim eksponentom. Eksponent se označava malim ili velikim slovom "e" koje prati opcioni predznak eksponenta (+ or -) i celobrojna vrednost eksponenta.

Globalna Delphi promenljiva DecimalSeparator definiše karakter koji se koristi kao decimalni zarez. Kako većina instrumenata podrazumeva da je decimalni zarez karakter ".", na računarima na kojima je Local Settings u Windows-u podešen na Srpsku verziju, neophodno je na početku programa promeniti vrednost promenljive DecimalSeparator.

Nakon:

```
S := '3.14';  
X := StrToFloat(S);
```

vrednost promenljive X biće 3,14.