



RAČUNARSTVO I INFORMATIKA – ISM

ČAS 2



- Promenljive ili varijable su objekti čija vrednost može biti promenjena tokom izvršavanja programa.
- Svaka promenljiva koja se koristi u programu mora biti prethodno deklarisanana.
- Deklaracijom promenljive ne određuje se njena vrednost, već samo tip podatka koji će u njoj biti smešten.
- Odeljak za deklarisanje promenljive počinje rezervisanom reči **var** (od eng. Reči variable=promenljiva) iza koje se navode nazivi promenljivih i njihovi tipovi. Promenljive istog tipa mogu biti odvojene zarezom, iza čega sledi dvotačka i tip podatka.
- Prilikom deklarisanja promenljive tipa **string** moguće je u uglastim zagradama iza rezervisane reči string definisati i maksimalnu dužinu stringa. Ukoliko prilikom deklarisanja nije naveden ovaj podatak, podrazumeva se da je maksimalna dužina stringa 255.



Primer

```
Var  x,y,z:real;  
     ime:string;  
     adresa:string[30];  
     i,j:integer;  
     pritisak:real;  
     iskaz1, iskaz2:boolean;  
     slovo:char;
```



- Izrazi su kombinacije simbola koje definišu poredak i način izračunavanja neke vrednosti korišćenjem:
 - Operanda (konstante, promenljive, funkcije)
 - Operatora
 - Oblih zagrada
- Operatori definišu koje je operacije potrebno izvršiti nad operandima, a zagrade definišu redosled vršenja tih operacija. U slučajevima kada redosled vršenja operacija nije određen zagradama, primenjuju se pravila o prioritetu operacija

Operatori	Prioriteti
not	1 (najveći priproitet)
and * / div mod	2
or + -	3
in = > < >= <= <>	4 (najmanji prioritet)

OPERACIJE NAD CELOBROJNIM (INTEGER) TIPOM PODATAKA



- Nad celobrojnim operandima se mogu izvoditi operacije koje daju celobrojni rezultat:

Operatori	Prioriteti
*	množenje
div	celobrojno deljenje
mod	ostatak pri celobrojnim deljenju
+	sabiranje
-	oduzimanje

- Primeri:

$$3 * 7 = 21$$

$$15 \text{ div } 2 = 7$$

$$15 \text{ mod } 2 = 1$$

$$4 + 9 = 13$$

$$3 * 7 \text{ div } 5 \text{ mod } 3 = 1$$

$$3 + 2 * 4 - 2 * 5 = 1$$

OPERACIJE NAD REALNIM (REAL) TIPOM PODATAKA



- Nad realnim operandima se mogu izvoditi operacije koje daju realni rezultat:

Operatori	Prioriteti
*	množenje
/	deljenje
+	sabiranje
-	oduzimanje

- Primeri:

$$3.47 * 7.21 = 25.0187$$

$$8.0 / 2.0 = 4.0$$

$$1.0 / 3.0 * 3.0 = 0.999999$$

$$6.17 + 2.14 * 3.81 - 4.5 = 9.8234$$

OPERACIJE NAD LOGIČKIM (BOOLEAN) TIPOM PODATAKA



- Nad logičkim operandima se mogu izvoditi operacije koje daju logički rezultat:

Operatori	Prioriteti
not	negacija
and	konjunkcija
or	disjunkcija
xor	eksluzivna disjunkcija

- Operator **not** ima viši prioritet od operacija **and** i **or**. Operacije istog prioriteta se izvršavaju sa leva na desno.

p	q	not p	p and q	p or q	p xor q
false	false	true	false	false	false
true	false	false	false	true	true
false	true	true	false	true	true
true	true	false	true	true	false



- Relacijski operatori omogućavaju poređenje dva operatora istog tipa, a dobijeni rezultat je logički tip. Iako su celi i realni brojevi formalno različiti tipovi podataka, korišćenjem relacijskih operatora moguće ih je međusobno porediti.

Operatori	Prioriteti
=	jednako
<>	različito
>	veće
<	manje
>=	veće ili jednako
<=	manje ili jednako



- Operator dodele je operator koji određenoj promenljivoj sa leve strane dodeljuje vrednost izraza sa desne strane.
- prvo izračuna vrednost izraza sa desne strane operatora, a zatim se ta vrednost upisuje u memorijsku lokaciju promenljive sa leve strane operatora.
- Tip vrednosti izraza na desnoj strani mora odgovarati tipu promenljive kojoj se vrednost dodeljuje.
- Jedini izuzetak je dodeljivanje celobrojnog izraza realnoj promenljivoj. U tom slučaju dobijena vrednost izraza se pretvara u realan broj i kao takva se dodeljuje realnoj promenljivoj sa leve strane.



- Razmenu podataka između programa i standardnog ulaza (tastatura, fajl na disku, ...) i izlaza (ekran, štampač, disk, ...) možemo izvršiti korišćenjem ulazno-izlaznih naredbi.
- Ulazne naredbe programskog jezika Pascal su
 - read
 - readln
- Izlazne naredbe su:
 - write
 - writeln



- omogućavaju organizovanje naredbi u funkcionalne celine
- Svaki blok naredbi počinje rezervisanom reči **begin**, a završava se rezervisanom reči **end**.
- Sve naredbe u okviru jednog bloka čine celinu i ne mogu se izvršavati odvojeno.

```
begin
```

```
  [komanda_1;]
```

```
  [komanda_2;]
```

```
  ...
```

```
  [komanda_n;]
```

```
end;
```

```
begin
```

```
  b:=5;
```

```
  suma:= b+3;
```

```
end;
```

- Pišu se unutar vitičastih zagrada {komentar}



- **ABS** - Vraća apsolutnu vrednost izraza
- **ARCTAN** - Izračunava arkustangens vrednosti u radijanima
- **CHR** - Vraća ASCII vrednost znakovnog argumenta
- **COS** - Vraća kosinus ugla
- **DEC** - umanjuje vrednost promenljive za navedeni korak. `DEC(i,10);`
- **EXP** - Izračunava stepen sa bazom prirodnog logaritma $e=2.718281828459$
Može da se koristi za stepenovanje a^b
`Stepen := EXP(eksponent*LN(v));`
- **FRAC** - Vraća decimalni deo realnog broja
`Writeln(Frac(3,14):8:7);`
IZLAZ: 0.14
- **INC** - Uvećaj vrednost promenljive za navedeni korak.
`INC(i,3);`
- **LENGTH** - Vraća broj karaktera u stringu.
- **LN** - Vraća prirodni logaritam.



- **ODD** - Vraća TRUE ako je vrednost argumenta neparna ili FALSE ako je vrednost parna.
- **ORD** - Vraća redni broj tipa.

N a p o m e n a: Za ASCII vrednosti, funkcija ORD je inverzna od funkcije CHR. Naredba CHR daje ASCII kod za datu vrednost, dok ORD od ASCII koda daje znak eksterne azbuke. (ORD('A')=65, CHR(65)=A)

- **RANDOM** - Vraća slučajnu vrednost broja.

```
PROGRAM Slucajan;  
VAR i: INTEGER;  
BEGIN  
    RANDOMIZE;  
    WRITELN('INTEGER REAL');  
    FOR i := 1 TO 10 DO WRITELN(RANDOM(65535):5, RANDOM:15:5);  
END.
```

- **ROUND** - Zaokruži na najbliži celi broj. **TRUNC** - Vraća ceo deo realnog broja
- **SIN** - Funkcija vraća sinus ugla.
- **SQR** - Vraća kvadrat date vrednosti.
- **SQRT** - Vraća kvadratni koren date vrednosti.



- **LENGTH** (*Str*) – određuje dužinu datog stringa **Str**
Pr. $a := 'Kragujevac'$
 $i := \text{length}(a) \rightarrow i = 10$
- **CONCAT** (*St1, St2, ..., Stn*) – sabira stringove *St1, St2, ..., Stn*
Pr. $a := 'Novi'$
 $b := 'Sad'$
 $c := \text{concat}(a, b) \rightarrow c = 'Novi Sad'$
- **POS** (*Str1, Str2*) – određuje vrednost pozicije od koje se string *Str1* nalazi u stringu *Str2*
Pr. $i := \text{pos}('ra', 'Kragujevac') \rightarrow i = 2$
- **COPY** (*St, Poz, Br*) – kopira string *St* od pozicije *Poz* i kopira *Br* simbola
Pr. $a := 'Beograd'$
 $b := \text{copy}(a, 4, 2) \rightarrow b = 'gr'$



- **DELETE** (*St*, *Poz*, *Br*) – briše string **St** od pozicije **Poz** i briše **Br** simbola.
Pr. `a := 'Kragujevac'`
 `delete(a, 7, 4) → a = 'Kraguj'`
- **INSERT** (*StUb*, *StObj*, *Br*) – u string *StObj* ubacuje u string *StUb* od pozicije *Br*
Pr. `a := 'Beod'`
 `b := 'gra'`
 `insert(b, a, 4) → a = 'Beograd'`
- **STR** (*NumVr*, *Str*) – Pretvara broj *NumVr* u string *Str*
Pr. `str(567, s) → s = '567'`
- **VAL** (*Str*, *NumVr*, *Kod*) – pretvara string *Str* u broj *NumVr*, ako je string preved bez greške *Kod* dobija vrednost 0, u suprotnom vrednost pozicije na kojoj je greška nastala
Pr. `val('345', i, k) → i = 345`



1. Uneti dva cela broja x i y . Na standardnom izlazu odštampati vrednost celobrojnog deljenja broja x sa bojem y , i vrednost ostatka pri tom deljenju.
2. Napisati program koji za uneti trocifreni ceo broj ispisuje zbir cifara tog broja.
3. Napisati algoritam i program koji dvema celobrojnim promenljivim zamenjuje vrednosti.
4. Neka se sa tastature unose realni brojevi a_1 , a_2 , i a_3 . Napiši program kojim se izračunava zbir brojeva koji se nalaze u zatvorenom intervalu 5 do 9. Rezultat prikaži u formatu od 5 mesta sa dve decimale.
5. Napisati algoritam i program kojim se za date realne brojeve a i b izračunava z po formuli:

$$z = \frac{\min(a, b) + 0.5}{1 + a}$$