



Osnovi programiranja – C

Operatori. Naredbe grananja



Teme

- Operator dodele
- Aritmetički operatori
- Relacioni i logički operatori
- Operator nabiranja (razdvajanja)
- Operator *sizeof*
- Blokovi naredbi
- Naredba if
- Naredba if-else-if
- Naredba switch
- Uslovni operator



Operator dodele

vrednost = izraz;

- Moguća je i lančana dodela

int x,y,z;

x=y=z=10,

- Izraz tipa **vrednost = vrednost + izraz** može se napisati kao **vrednost += izraz;**
- Treba zapamtiti suštinsku razliku između jednostrukog znaka **jednakosti (=)** koji je simbol za dodelu, i dvostrukog znaka jednakosti **(==)** koji je **simbol za poređenje**

Aritmetički operatori

- Operator modulo (modulo) (%) može se primenjivati isključivo na celobrojne promenljive

Operator	Značenje
+, -, *, /	sabiranje, oduzimanje, množenje, deljenje
%	modulo (ostatak celobrojnog deljenja)
++	inkrement (uvećava za 1)
--	dekrement (umanjuje za 1)

Operacija deljenja

- Da bi se izbegle nedoumice oko tipa rezultata, preporučljivo je brojeve definirati sa **float**, čak i kada **nemaju decimalnih mesta**

float rezultat = 3 / 4; //rezultat je nula.

rezultat =(float) 3 / 4 ; //rezultat je 0.75



Inkrement i dekrement

- Ove operatore treba koristiti kad god je moguće (generišu efikasan kod)

Primjer

```
int brojac;
```

```
brojac = 0;
```

```
//sledeće tri naredbe su identične
```

```
brojac = brojac + 1;
```

```
brojac++;
```

```
++brojac;
```



Inkrement i dekrement (2)

- Razlikujemo prefiksnu (++i) i postfiksnu (i++) notaciju

```
#include <stdio.h>
```

```
int main() {
```

```
    int i=100;
```

```
    int j=++i;
```

```
    printf("t=0, j =%d\n", j);
```

```
    int k=100;
```

```
    j =k++;
```

```
    printf("t=1, j =%d\n", j);
```

```
    return 0;
```

```
}
```

t=0, j =101

t=1, j =100

Relacioni i logički operatori

Relacioni operator	Značenje
>, >=, <, <=	veće, veće ili jednako, manje, manje ili jednako
==	operator jednakosti
!=	operator nejednakosti

Logički operator	Značenje
&&	AND (i)
	OR (ili)
!	NOT (negacija)

Operatori sa bitima

Logički operator	Značenje
&	AND (i)
	OR (ili)
^	XOR (EXILI)
~	Komplement (jedinični)
>>	Pomeranje bita udesno (right shift)
<<	Pomeranje bita ulevo (left shift)

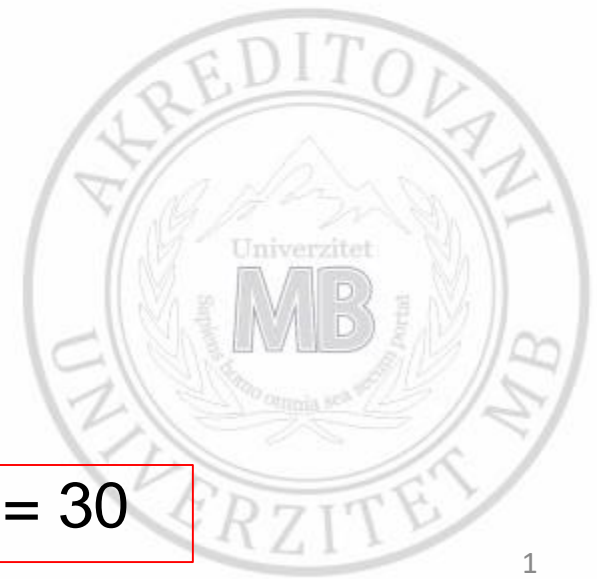
Operator razdvajanja ili nabiranja (,)

- Vrijednost niza izraza je ona koja je posljednja u nizu.

```
#include <stdio.h>
int main()
{
    int l1=1, l2=2, l3=3, l4=4;
    l4= (l1=10, l2=20, l3=30);
    printf("l4 = %d\n", l4);

    return 0;
}
```

l4 = 30



Operator *sizeof*

```
#include <stdio.h>
```

```
int main() {
```

```
printf("The size of an int is: %d", sizeof(int));
```

```
printf(" bytes.\n");
```

```
printf("The size of a short int is: %d", sizeof(short));
```

```
printf(" bytes.\n");
```

```
printf("The size of a long int is: %d", sizeof(long));
```

```
printf(" bytes.\n");
```

```
return 0;
```

```
}
```

- Prikazuje dužinu određenog tipa u bajtovima

The size of an **int** is: 4 bytes.

The size of a **short int** is: 2 bytes.

The size of a **long int** is: 4 bytes.

Prioritet operatora

- Redosled izvršavanja operatora je definisan, ali se uvek može izmeniti pomoću zagrada
- Dobra je navika stavljati zagrade i razmake uvek kada postoji dilema o hijerarhiji operatora

Operator
()
!, + (unarni), - (unarni), ++, --
*, /, %
+, -
<, <=, >, >=
==, !=
&&
=, +=, -=, *=, /=

Konverzija tipova u izrazima i dodelama

- Kada u izrazu dodele učestvuju operatori različitih tipova, vrednost desne strane (izraza) biće konvertovana u tip leve strane (tip promenljive)
- Kada u izrazu učestvuju operandi različitih tipova, svi se konvertuju u tip najvećeg operanda (tzv. unapređivanje tipa)



Konverzija tipova

Prioritet operatora (konverzije)	Tipovi operatora
	long double
↑	double
↑	float
↑	unsigned long
↑	long
↑	unsigned int
↑	int
↑	unsigned short
↑	short
↑	unsigned char
↑	char

Konverzija tipa: cast operator

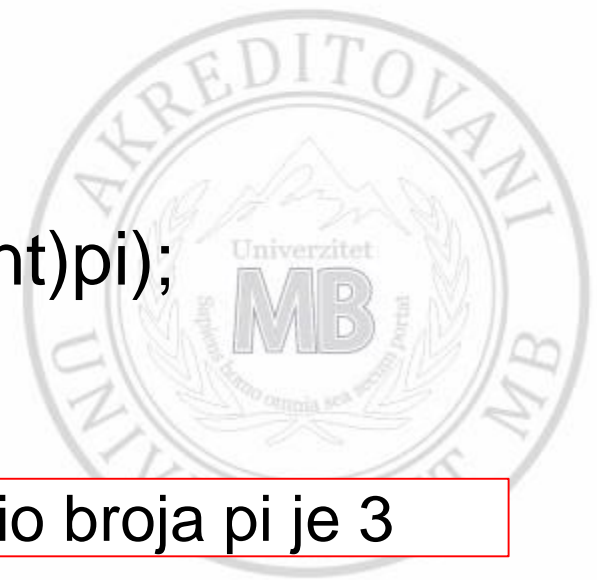
(tip)izraz

```
#include <stdio.h>
int main() {
```

```
float pi=3.1415926;
printf("Celi deo broja pi je %d\n", (int)pi);
```

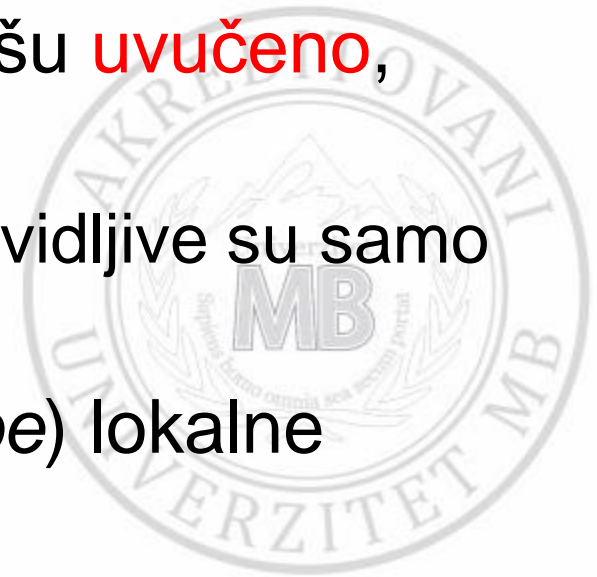
```
return 0;
}
```

Celi dio broja pi je 3



Blokovi naredbi

- Delovi programa koji se uslovno izvode ili čije se izvršavanje ponavlja grupišu se u blokove naredbi tj. u jednu ili više naredbi unutar paravitičastih zagrada { }
- **Blokovi naredbi** se uobičajeno pišu **uvučeno**, zbog preglednosti programa
 - promenljive deklarisanе u bloku vidljive su samo unutar njega
- Ograničenje važnosti (engl. *scope*) lokalne promenljive



Blokovi naredbi (2)

```
#include <stdio.h>
int main() {
{
int i=1;
}

printf("%d\n", i); //greška
return 0;
}
```

```
#include <stdio.h>
int main() {
int i=10;
{
int i=1;
printf("%d\n", i);
}

printf("%d\n", i);
return 0;
}
```

1
10

Grananje naredbom IF

- Opšta forma naredbe je:
if (izraz) naredba;
else naredba; //nije obavezno
- If petlju može da kontroliše bilo koji validni logički izraz



Grananje naredbom IF (2)

```
#include <stdio.h>
int main() {
int i=10;
if (i<0)
    printf("broj je negativan\n");
if (i % 2)
    printf("broj je neparan\n");
else
    printf("broj je paran\n");
return 0;
}
```



```
#include <stdio.h>
```

```
int main() {
```

```
int prviBroj, drugiBroj;
```

```
printf("Uneti veliki broj: ");
```

```
scanf ("%d", &prviBroj);
```

```
printf("\nUneti mali broj: ");
```

```
scanf ("%d", &drugiBroj);
```

```
if (prviBroj > drugiBroj)
```

```
    printf("\nHvala!\n");
```

```
else
```

```
    printf("\nGreska, drugi broj je veci!\n");
```

```
return 0;
```

```
}
```

Primer #1

Uneti veliki broj: 1000

Uneti mali broj 10

Hvala!

IF-ELSE-IF lestvica

```
if(uslov1)
    naredbe;
else if(uslov2)
    naredbe;
else if(uslov3)
    naredbe;
...
else naredbe;
```

- Moguće je pisati i ugnježdene IF blokove; u tom slučaju, blok ELSE uvek pripada najbližem IF-u
- Uslovni izrazi se proveravaju odozgo naniže; čim se pronađe tačan, izvršava se naredba povezana sa njim, a ostatak lestvice se preskače
- Poslednji else služi kao default izbor, ako ništa pre njega nije ispunjeno

```
#include <stdio.h>
```

```
int main() {
```

```
int prviBroj, drugiBroj;
```

```
printf("Uneti dva broja.\nPrvi: ");
```

```
scanf("%d", &prviBroj);
```

```
printf("\nDrugi: "); scanf("%d", &drugiBroj); printf("\n\n");
```

```
if (prviBroj >= drugiBroj) {
```

```
    if ((prviBroj % drugiBroj) == 0) { // parno deljivi?
```

```
        if (prviBroj == drugiBroj)
```

```
            printf("Brojevi su isti!\n");
```

```
            else printf("Parno su deljivi!\n");
```

```
        }
```

```
    else printf("Nisu parno deljivi!\n");
```

```
}
```

```
else printf("Drugi broj je veci\n");
```

```
return 0;
```

```
}
```

Primer #2

Uslovni operator ?

- Uslovni operator treba koristiti samo za jednostavna ispitivanja, kada naredba staje u jedan red; u suprotnom, kod postaje nečitljiv.

```
#include <stdio.h>
```

```
int main() {  
int x, y = 10;  
x = (y < 10) ? 30 : 40;  
printf("Vrednost za x: %d\n", x);  
  
return 0;  
}
```



Grananje toka naredbom switch

```
switch(izraz) {
```

```
  case (konstanta1):
```

```
    naredbe;
```

```
    break;
```

```
  case (konstanta2):
```

```
    naredbe;
```

```
    break;
```

```
  case(konstanta3):
```

```
    naredbe;
```

```
    break;
```

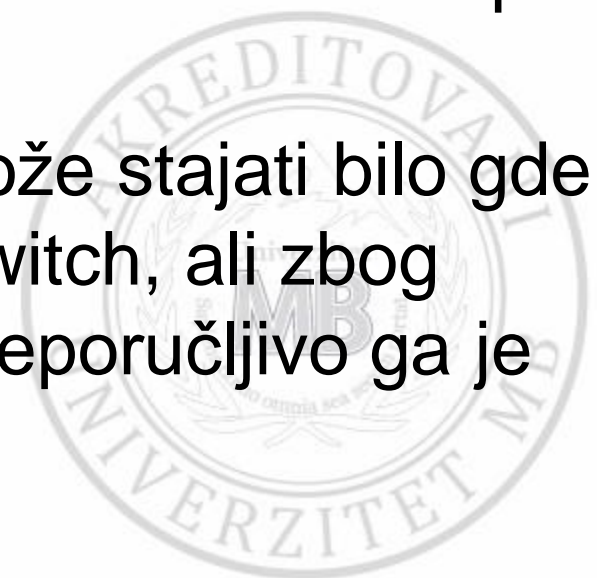
```
  ...
```

```
  default
```

```
    naredbe;
```

```
}
```

- Naredba switch se koristi umesto serije ugnježenih IF- ELSE naredbi, kada se kontrola naredbe može vršiti izrazom tipa char ili int
- Blok default može stajati bilo gde unutar bloka switch, ali zbog preglednosti preporučljivo ga je staviti na kraj



Izvršavanje naredbe switch

- Izračunava se izraz
- Izračunata vrednost se redom upoređuje sa vrednostima u case delovima
- Za prvu pronađenu istu vrednost se izvršava odgovarajuća naredba u nastavku
- Ako nije pronađena nijedna ista vrednost, izvršava se naredba iza reči default, ako postoji
- Naredba break nije obavezna, ali bez nje grananje naredbom switch nema mnogo smisla

```
#include <stdio.h>
```

```
int main() {  
int x;  
printf("Unesite celi broj: ");  
scanf("%d", &x);  
switch (x) {  
    case 1:  
        printf("x = 1");  
        break;  
    case 2:  
        printf("x = 2");  
        break;  
    default:  
        printf("x = %d\n\n", x);  
}  
return 0;  
}
```

Naredba switch: Primer

//if ekvivalent

```
if (x == 1) {  
    printf("x = 1");  
}  
else if (x == 2) {  
    printf("x = 2");  
}  
else {  
    printf("x = %d\n\n", x);  
}
```

Zaključak

- Operator dodele: **vrednost = izraz;**
- Aritmetički operatori: **+, -, *, /, %, ++, --**
- Relacioni operatori: **<, <=, >, >=, ==, !=**
- Logički operatori: **&&, ||, !**
- Operator nabiranja (razdvajanja)
- Operator *sizeof*: **prikazuje dužinu određenog tipa u bajtovima**
- Prioritet operatora
- Konverzija tipova u izrazima i dodelama



Zaključak (2)

- **Blokovi naredbi:** grupa naredbi unutar para { }
- **if-else naredba**
- **if-else-if:** ugnježdene IF naredbe
- **Uslovni operator ?:** koristi se samo za jednostavna ispitivanja kada naredba staje u jedan red
- **Naredba switch:** izračunata vrednost se redom upoređuje sa vrednostima u **case** delovima





Kraj prezentacije

HVALA NA PAŽNJI!

