

PROGRAMIRANJE 1

UVOD U

PROGRAMSKI JEZIK

C

Staša Vujičić
Čas 6
V1.1

OSNOVNI TIPOVI PODATAKA U C-U

Tip	Opis	Uobičajena veličina
char	Mali ceo broj (ASCII kod karaktera)	1 bajt
int	Ceo broj	4 bajta
float	Realan broj	4 bajta
double	Realan broj	8 bajtova

OSNOVNI TIPOVI PODATAKA U C-U

- ◉ Tipovi iz tabele se mogu modifikovati ključnim rečima **short**, **long**, **signed** i **unsigned**.
- ◉ Na **int** se mogu primeniti **short** i **long** (u tom slučaju ključna reč **int** nije obavezna).
- ◉ Na **double** se može primeniti **long**.
- ◉ Na **float** se ne može primeniti nijedan modifikator.

- ◉ Modifikatori **unsigned** i **signed** se mogu primeniti na celobrojne tipove (char, int, short i long).
- ◉ Veličine tipova su platformski zavisne.
- ◉ Obavezna relacija za celobrojne tipove:
 $\text{short} \leq \text{int} \leq \text{long}$.

KONSTANTE

- **Karakterske konstante** se navode između jednostrukih navodnika. One su tipa **char**.
- **Celobrojne konstante** mogu biti dekadne, oktalne i heksadekadne. Sve ove konstante su tipa **int**.
 - Oktalne počinju nulom
 - Heksadekadne počinju sekvencom 0x ili 0X.
- **Sufiksi u ili U** daju neoznačenu konstantu.
- **Sufiksi l ili L** daju konstantu tipa **long**.

- ⦿ Konstante realnih brojeva sadrže decimalnu tačku(123.4) ili eksponent(1e-2) ili i jedno i drugo. Njihov tip je double osim ako:
- ⦿ se završavaju sufiksom **f ili F** – tada su tipa **float**.
- ⦿ se završavaju sufiksom **l ili L** – tada su tipa **long double**.

PRIMERI:

- 'a'
- 'ab'
- '\n'
- 34
- 034
- 093
- 0xa
- 2
- 321
- 0x12U
- 034ul
- 12Lu
- 4.2
- 2e-3
- 1.e-3
- .5e-1f
- 5.4e-3.4
- 5.4e-3L

OPERATORI

- ⊙ U C-u postoji veliki broj operatora. Mogu biti unarni i binarni.
- ⊙ **Unarni operatori** mogu biti prefiksni i sufiksni.
- ⊙ **Binarni operatori** su po pravilu infiksni.
- ⊙ Operatori imaju svoj prioritet i asocijativnost.

IZRAZI

- ◉ Kombinovanjem promenljivih, konstanti i operatora dobijamo **izraze**.
- ◉ Svaki izraz ima svoj **tip** i svoju **vrednost**.

TIP IZRAZA I KONVERZIJA

- ⦿ Tip izraza zavisi od tipova podizraza koji ga čine, kao i od operatora kojim se ti podizrazi povezuju.
- ⦿ Ako su **operandi** neodgovarajućeg tipa, tada se implicitno konvertuju u odgovarajući tip, ako je to moguće. Koja će se konverzija izvršiti zavisi od operatora.

PRIMER - KONVERZIJA

- Naredba dodele:

```
int i=5;
```

```
float f=2.3;
```

```
f=i; /* f ce imati vrednost 5.0*/
```

- obrnuto:

```
int i=5;
```

```
float f=2.3;
```

```
i=f; /* i ce imati vrednost 2*/
```

KONVERZIJE TIPOVA

- Tip izraza se može eksplicitno promeniti tzv. **cast** operatorom (ispred izraza se u zagradi navede ime tipa u koji želimo da konvertujemo izraz).

(tip)<izraz>

- float x;

x=2.3+4.2; /* x ce imati vrednost 6.5 */

x=(int)2.3+(int)4.2; /*x ce imati vrednost 6 */

x=(int)2.3*4.5; /* x ce imati vrednost 9.0 jer zbog prioriteta operatora konverzije prvo ce biti izvršena konverzija broja 2.3 u 2 pa tek onda izvršeno množenje. */

- x=(int)(2.3*4.5) /* x ce imati vrednost 10.0 */

PRIMER

- *Kako izbeći celobrojno deljenje*

- **int a,b;**

- float c;**

- a = 5;**

- b = 2;**

- c = a/b;** /* Celobrojno deljenje, c=2*/

- c = (1.0*a)/b;** /*Implicitna konverzija: 1.0*a je realan broj pa prilikom deljenja sa b dobija se realan rezultat c=2.5*/

- c = (0.0+a)/b;** /* Implicitna konverzija: (0.0+a) je realan broj pa prilikom deljenja sa b dobija se realan rezultat c=2.5*/

- c = (float)a/(float)b;** /* Eksplicitna konverzija*/

ARITMETIČKI OPERATORI

- Operatori **+**, **-**, *****, **/**, **%** nazivaju se aritmetički operatori.
- Ovi operatori se primenjuju na celobrojne i realne tipove.
- Operator **/** primenjen na cele brojeve daje celobrojni količnik.
- Operator **%** se primenjuje samo na cele brojeve i daje ostatak pri deljenju.

- ⦿ Rezultat aritmetičkih operacija je izraz istog tipa kao i operandi.
- ⦿ Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip.

RELACIONI OPERATORI

- Operatori `==`, `!=`, `<`, `>`, `<=`, `>=` nazivaju se relacioni operatori.
- Ovi operatori se primenjuju na celobrojne i realne tipove.
- Rezultat relacionih operacija je izraz celobrojnog tipa sa vrednošću 1 ako je relacija tačna, 0 u suprotnom.
- Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip.

LOGIČKI OPERATORI

- ◉ Logički operatori su:
- ◉ **!** – unarna negacija
- ◉ **&&** – logičko i
- ◉ **||** – logičko ili.

LOGIČKI OPERATORI

- ◉ U C-u ne postoji logički tip.
- ◉ Svaki izraz koji se može porediti na jednakost sa nulom se u C-u smatra logičkim izrazom.
- ◉ Ako je vrednost izraza različita od 0, tada se on smatra **logički tačnim**. Izraz jednak nuli se smatra **logički netačnim**.
- ◉ Rezultat logičkih operacija je izraz celobrojnog tipa, jednak 0 ako je rezultat logičke operacije netačno, odnosno 1, ako je rezultat logičke operacije tačno.

PRIMERI - LOGIČKI OPERATORI

- ◉ $5 \ \&\& \ 4$ – vrednost je tačno,
- ◉ $10 \ || \ 0$ – vrednost je tačno,
- ◉ $0 \ \&\& \ 5$ – vrednost je 0,
- ◉ $!1$ – vrednost je 0,
- ◉ $!9$ – vrednost je 0,
- ◉ $!0$ – vrednost je 1,
- ◉ $!(2>3)$ – vrednost je 1.

OPERATORI SA BOČNIM EFEKTIMA

- ⦿ Pojava da se prilikom izračunavanja nekog izraza menja vrednost neke promenljive naziva se **bočni efekat** (eng. side effect).
- ⦿ Operatori koji imaju bočni efekat su operatori dodele i operatori uvećanja i umanjenja.
- ⦿ Naredba dodele se u C-u predstavlja izrazom dodele, a njeno izvršavanje se zasniva na bočnim efektima operatora dodele.

OPERATORI DODELE

- ◉ Operator proste dodele je `=`. Ne treba ga mešati sa relacionim operatorom `==`.
- ◉ Levi operand ovog operatora je leva vrednost (ime promenljive). Desni operand je proizvoljni izraz.
- ◉ Najpre se izračunava izraz na desnoj strani. Njegova vrednost se zatim po potrebi konvertuje u tip promenljive na levoj strani. Nakon toga se dobijena vrednost upisuje u memorijsku lokaciju koja je rezervisana za čuvanje te promenljive.

- ◉ Tip izraza dodele je tip promenljive na levoj strani. Vrednost izraza dodele je vrednost koja je dodeljena promenljivoj.
- ◉ Operatori složene dodele su $+=$, $-=$, $*=$, $/=$, $\%=$.
- ◉ Izraz oblika $a+=E$ je ekvivalentan izrazu $a=a+(E)$.
- ◉ Slično je sa ostalim operatorima.

OPERATORI UMANJENJA I UVEĆANJA

- ◉ Unarni operatori **++** i **--** nazivaju se operatori **uvećanja** (inkrementacije) i **umanjenja** (dekrementacije) respektivno.
- ◉ Mogu biti prefiksni i sufiksni.
- ◉ U oba slučaja vrši se uvećanje promenljive za 1 ali izraz **++n** uvećava promenljivu *n* *pre* nego što se njena vrednost koristi, dok **n++** uvećava *n* *nakon što se njena vrednost koristi*. Tako se `x=++n;` razlikuje od `x=n++;`.
- ◉ Slično za operator **--**, s tim što je u pitanju umanjeње za jedan.

PRIMERI

- ⦿ $a=5$; nakon $a++$, $a=6$
- ⦿ $a=5$; nakon $++a$, $a=6$
- ⦿ $a=5$; nakon $b=a++$, $a=6$, $b=5$
- ⦿ $a=5$; nakon $b=++a$, $a=6$, $b=6$

PRIORITET I ASOCIJATIVNOST OPERATORA

- ◉ Unarni operatori imaju viši prioritet od binarnih.
- ◉ Aritmetički operatori su višeg prioriteta od relacionih, a ovi višeg od logičkih.
- ◉ Operatori dodele su najnižeg prioriteta.
- ◉ Ako dva operatora imaju isti prioritet, onda se u obzir uzima asocijativnost, koja može biti s leva na desno, ili s desna na levo.
- ◉ Prioritet operatora se može promeniti korišćenjem zagrada ().

PRIMERI

- ◉ unarni operatori su višeg prioriteta od binarnih: $-1+2=1$
- ◉ aritmetički operatori su višeg prioriteta od relacionih: $3+5<6=0$
 $3+(5<6)=4$
- ◉ relacioni operatori su višeg prioriteta od logičkih: $0<-1 || 0=0$
 $0<(-1 || 0)=1$

- ◉ Binarni operatori su leve asocijativnosti, osim operatora dodele koji imaju desnu asocijativnost. Unarni operatori su mahom desne asocijativnosti, ali ima izuzetaka.

$x=y==z$ je isto što i $x=(y==z)$

$-a*b+c$ je isto što i $((-a)*b)+c$

$a=b=c=d$ je isto što i $(a=(b=(c=d)))$

PROSTA NAREDBA

- ◉ **Prosta naredba** se sastoji iz izraza za kojim sledi karakter ';'.
- ◉ Naredba se izvršava tako što se izračunava izraz iz koga se sastoji uz sve bočne efekte koji se pri tom javljaju.
- ◉ Jasno je da prosta naredba ima smisla samo ako sadrži bar jedan bočni efekat.

PETLJA DO-WHILE

- Slična je repeat-until izrazu u Pascal-u.
- **do naredba**
while (izraz);

PRIMER - PETLJA DO-WHILE

○ **#include <stdio.h>**

int main()

{

int x;

x = 1;

do

{

printf("x = %d\n",x);

x++; /* x++ je isto kao i x=x+1 */

}

while (x<10);

}

⊙ Izlaz:

$$x = 1$$

$$x = 2$$

$$x = 3$$

$$x = 4$$

$$x = 5$$

$$x = 6$$

$$x = 7$$

$$x = 8$$

$$x = 9$$

FOR PETLJA

- ⦿ **for (izraz1; izraz2; izraz3) naredba**
- ⦿ Ovo je ekvivalentno kodu:

```
izraz1;  
while (izraz2)  
{  
    naredba  
    izraz3;  
}
```

PRIMER - FOR PETLJA

◎ **#include <stdio.h>**

```
int main()
```

```
{
```

```
    int x;
```

```
    for (x = 1; x < 10; x++)
```

```
        printf("x = %d\n",x);
```

```
}
```

⊙ Izlaz:

$$x = 1$$

$$x = 2$$

$$x = 3$$

$$x = 4$$

$$x = 5$$

$$x = 6$$

$$x = 7$$

$$x = 8$$

$$x = 9$$

- ⦿ Napomena: izraz1, izraz2, izraz3 i naredba mogu biti izostavljeni. Ako je izraz2 izostavljen podrazumeva se da je stalno tačan.
- ⦿ **for(; ;);** predstavlja "beskonačnu" for petlju.

NAREDBE BREAK I CONTINUE

- Naredba **break** omogućava prevremeni izlazak iz petlje a **continue** omogućava izlazak iz tekuće iteracije u petlji i nastavak izvršenja petlje počev od sledeće iteracije.

PRIMER - BREAK

◎ `#include <stdio.h>`

```
int main()
```

```
{
```

```
    int i;
```

```
    for(i=1; i<=5; i++)
```

```
    {
```

```
        if(i==3) break;
```

```
        printf("i = %d\n", i);
```

```
    }
```

```
}
```

⦿ Izlaz:

$i = 1$

$i = 2$

PRIMER - CONTINUE

⦿ `#include <stdio.h>`

```
int main()
```

```
{
```

```
    int i;
```

```
    for(i=1; i<=5; i++)
```

```
    {
```

```
        if(i==3) continue;
```

```
        printf("i = %d\n", i);
```

```
    }
```

```
}
```

⊙ Izlaz:

$i = 1$

$i = 2$

$i = 4$

$i = 5$