

# Programiranje 1

Programski jezik C  
1. čas

# Leksika programskog jezika C

- U upotrebi su samo simboli iz ASCII skupa
- Velika i mala slova se razlikuju
- Komentari se navode između `/*` i `*/` i mogu se prostirati u više redova, ne mogu biti ugneždeni
- `int a, A; /*` ovo su dve različite promenljive `*/`

# Tokeni

Postoji šest vrsta tokena:

1. identifikatori
  2. ključne reči
  3. operatori
  4. separatori
  5. stringovi
  6. konstante
- Tokeni se razdvajaju belinama, tabulatorima i novim redovima.

# Identifikatori

- Identifikatori se sastoje iz slova, cifara i znaka \_ pri čemu prvi karakter nije cifra.
- Koriste se za imena promenljivih, funkcija, itd.

# Ključne reči

- Ključne reči su rezervisane reči koje imaju posebnu ulogu, i ne mogu se koristiti kao identifikatori.
- Ključne reči se koriste za:
  - definisanje jezičkih konstrukcija (***if, while, for***),
  - imena tipova (***int, float, char***), itd.

# Osnovni tipovi podataka u C-u

- char - mali ceo broj ili ASCII kod karaktera, uobicajena velicina mu je 1 bajt
- int – ceo broj, 4 bajta
- float – realan broj, 4 bajta
- double – realan broj u dvostrukoj preciznosti, 8 bajtova
- **Velicine tipova se razlikuju od platforme do platforme**

# Osnovni tipovi podataka u C-u

- Ovi tipovi se mogu modifikovati ključnim rečima short, long, signed i unsigned. (na int se mogu primeniti sve četiri, na double long, a na float nijedna)
- Sa unsigned i signed se modifikuju samo celobrojni tipovi (char, int, short i long)
- Obavezna relacija koja mora da važi za celobrojne tipove je:  
short < int < long

# Konstante

- Karakterske konstante su tipa char i navode u obliku 'a', 'A'
- Celobrojni tip se može navoditi u dekadnom sistemu, oktalnom (prefiks 0) i heksadekadnom (prefiks 0x)
- Realne konstante se navode sa ili bez eksponencijalnog dela i kao takve su double tipa (ako imaju sufiks f ili F onda su float tipa, a sa sufiksom l ili L su tipa long double)



# Promenljive

- Sve promenljive u C-u se moraju deklarirati. Time se za promenljivu u memoriji rezerviše potreban prostor, a ostatak programa postaje svestan postojanja promenljive i njenog tipa.
- Deklaracija se sastoji iz imena tipa za kojim slede imena promenljivih (identifikatori) koje se deklarišu, i koja su razdvojena zarezima.
- Deklaracija se, kao i svaka naredba u C-u završava sa ;
- Svako ime promenljive u deklaraciji može biti praćeno inicijalizatorom koji se sastoji iz karaktera = za kojim sledi inicijalna vrednost.

# Promenljive

Da bi se bilo koja promenljiva mogla upotrebljavati u programu, ona mora biti deklarirana obavezno na početku programa

**int a, b = 5;** /\* promenljiva a nije inicijalizovana, a vrednost promenljive b je postavljena na 5 \*/

**const double e = 2.728282;** /\* kvalifikator const služi da obeleži promenljivu koja se u programu neće menjati \*/

# Program u C-u

Izvorni program (source code) u C-u je običan tekstualni fajl, kreiran u bilo kom editoru teksta (npr. Notepad, pico, emacs, kate).

- Program u C-u sastoji se iz definicija funkcija.
- Funkcija koja mora da postoji u svakom programu je **main()**. Izvršavanje programa se svodi na izvršavanje tela ove funkcije.
- Telo funkcije se navodi iza zaglavlja funkcije `int main()`, između vitičastih zagrada `{ i }`.

# Program u C-u

U telu funkcije se na početku navode deklaracije pomenljivih, nakon čega sledi proizvoljan niz naredbi.

```
int main() /* zaglavlje funkcije main */  
{ /* ovde pocinje telo */  
/* deklaracije promenljivih */  
/* naredbe */  
} /* kraj tela */
```

# Prvi primeri u C-u

Napisati program koji na standardnom izlazu štampa **Zdravo, svete!**

```
#include <stdio.h>
main()
{
    printf("Zdravo, svete!\n");
    return 0;
}
```

- Izlaz iz programaće biti:

**Zdravo, svete!**

# Prvi primeri u C-u

Šta je izlaz iz sledećeg programa?

```
#include <stdio.h>  
int main()  
{  
printf("Zdravo, ");  
printf("svete!");  
printf("\n");  
return 0;  
}
```

# Uvođenje promenljivih u program

```
#include <stdio.h>
main()
{
/* Deklaracija vise promenljivih istog tipa */
int rez,pom1,pom2;
pom1=20; pom2=15;
rez=pom1-pom2;

/* Ispisivanje rezultata */
printf("Rezultat je %d - %d = %d\n", pom1, pom2, rez);
return 0;
}
```

- Izlaz iz programa je:

Rezultat je 20-15=5

# Funkcije ulaza i izlaza

Ulaz i izlaz ostvaruju se posredstvom funkcija koje su definisane u standardnoj biblioteci **stdio.h**

- Ove funkcije su uobičajene C funkcije, koje se služe direktno servisima operativnog sistema (sistemskim pozivima) prilikom svog rada.



# Funkcije ulaza i izlaza

Za korišćenje ovih funkcija neophodno je uključiti zaglavlje `stdio.h` navođenjem direktive **`#include<stdio.h>`** pre definicije funkcije `main()`.

- Ovo zaglavlje je tekstualni fajl u kome su navedene deklaracije funkcija ulaza i izlaza.
- Direktiva `#include` na mestu poziva uključuje kompletan sadržaj fajla koji je naveden, čime funkcije i podaci deklarirani u njemu postaju dostupni funkciji `main()`.

# Funkcija printf()

- Ovom funkcijom se ispisuje poruka zadata format stringom na standardni izlaz.

```
printf("%d - %d = %d\n", pom1, pom2, rez);
```

- Eventualni konverzioni specifikatori se zamenjuju vrednostima izraza koji u tom slučaju slede nakon format stringa, kao argumenti funkcije printf(), razdvojeni zarezima i u onom poretku u kome su odgovaraju ći konverzioni specifikatori navedeni.
- Tipovi izraza moraju biti u skladu sa tipovima koje određuju konverzioni specifikatori.

# Konverziji specifikatori za printf()

Specifikator	Tip	Napomena
%d	int	dekadni ispis
%f	float, double	ispis bez eksponenta
%Lf	long double	ispis bez eksponenta
%hd	short	dekadni ispis
%ld	long	dekadni ispis
%c	char	ispis ASCII karaktera

# printf()

Funkcija printf je bibliotečka funkcija koja prikazuje izlazne podatke u određenom formatu.

- Primer korišćenja funkcije printf je:

```
printf("%d\t%d\n", broj1, broj2);
```

- Prvi argument ove funkcije je uvek između " " i određuje format u kome će se podaci ispisati na izlaz.
- Ova funkcija vraća kao vrednosti broj upisanih znakova na izlazu.

# printf()

Sekvenca **\n** u okviru prvog argumenta funkcije printf je C oznaka za prelazak u novi red, **\t** je oznaka za tabulator, dok **%d** označava da će na tom mestu biti ispisana celobrojna vrednost argumenta koji je sa njim u paru. Svaka % konstrukcija je u paru sa odgovarajućim argumentom koji sledi.

**%%** koristi se za ispis znaka %

**\\** koristi se za ispis znaka \

**\"** koristi se za ispis znaka ”

# printf()

Postoji mogućnost da se precizira i širina polja u kome će se ispisati odgovarajuće vrednosti.

Na primer, koristimo `%3c` za štampanje karaktera na tri pozicije poravnato udesno.

Koristimo `%3d` za štampanje broja na tri pozicije ili `%6d` za štampanje broja na 6 pozicija.

**`%f`** — štampaj kao realan broj

**`%6f`** — štampaj kao realan broj širok najviše 6 znakova

**`%.2f`** — štampaj kao realan broj sa dve decimale

**`%6.2f`** — štampaj kao realan broj širok najviše 6 znakova pri čemu su 2 iza decimalne tačke.

- Da bi se izvršilo levo poravnanje, između `%` i odgovarajućeg karaktera dodaje se znak `-`

# Primer sa printf()-om

```
#include <stdio.h>
main()
{
printf("Slova:\n%3c\n%5c\n", 'z' , 'Z');
return 0;
}
```

Izlaz iz programa će biti:

Slova:

z

Z

# Funkcija scanf()

Ovom funkcijom se učitavaju podaci sa standardnog ulaza (tastature).

- Prvi argument je format string u kome se navode konverzioni specifikatori kojima se definiše tip podatka koji se očekuje, tj. koje se učitava.
- Nakon format stringa slede adrese promenljivih, razdvojene zarezima, u koje treba upisati vrednosti učitane sa ulaza. Adresa promenljive a navodi se sa **&a**
- Adrese se navode u onom poretku u kom su odgovarajući konverzioni specifikatori navedeni u format stringu.
- Tipovi promenljivih moraju biti u skladu sa tipovima koje određuju konverzioni specifikatori.



## Konverziji specifikatori za scanf()

Specifikator	Tip	Napomena
%d	int	opciono označeni dekadni broj
%f	float	realni broj sa opcionim eksponentom
%Lf	long double	realni broj sa opcionim eksponentom
%hd	short	opciono označeni dekadni broj
%ld	long	opciono označeni dekadni broj
%c	char	ASCII karakter
%lf	double	realni broj sa opcionim eksponentom

# Primer korišćenja scanf()

**scanf("%d %d, &a, &b);**

- Ova funkcija čita sa ulaza dva cela broja i smešta ih na adresu promenljivih a i b, redom.
- Kao rezultat, ova funkcija vraća broj uspešno dodeljenih ulaznih vrednosti, tj. u ovom primeru 2.
- Naredni poziv funkcije scanf() nastavlja čitanje neposredno iza poslednjeg znaka koji je već pročitano.

# Primer korišćenja scanf()

Program prikazuje unos celog broja koristeći funkciju scanf()

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

```
main()
```

```
{ int x;
```

```
printf("Unesi ceo broj : ");
```

```
/* Obratiti paznju na znak & pre imena  
promenljive u funkciji scanf. */
```

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

```
/* U funkciji printf nije potrebno stavljati &. */
```

```
printf("Uneli ste broj %d\n", x);
```

```
return 0;}
```

