PROGRAMIRANJE

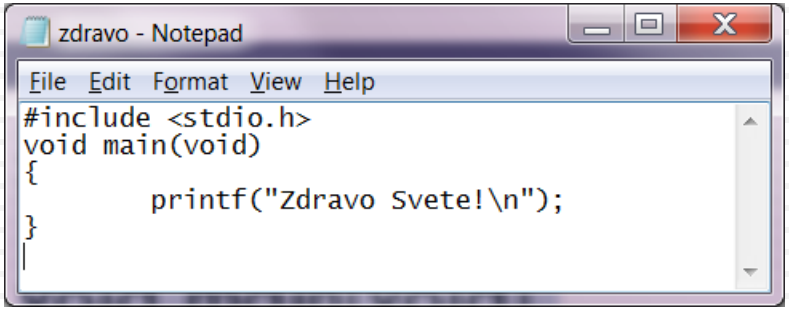
PROGRAMSKI JEZIK C

Ulaz/Izlaz
Operatori i Izrazi

Istorijat jezika C

- Razvijen 70-ih godina (Dennis M. Ritchie, Bell Labs)
- Osnov za razvoj UNIX operativnog sistema
- 1978, Brian Kernighan, Dennis Ritchie “The C Programming Language” (K&R standard)
- 1989, ANSI standard jezika C (C89, odnosno C90)
- 1999, ANSI standard jezika C (C99)

Elementi C programa

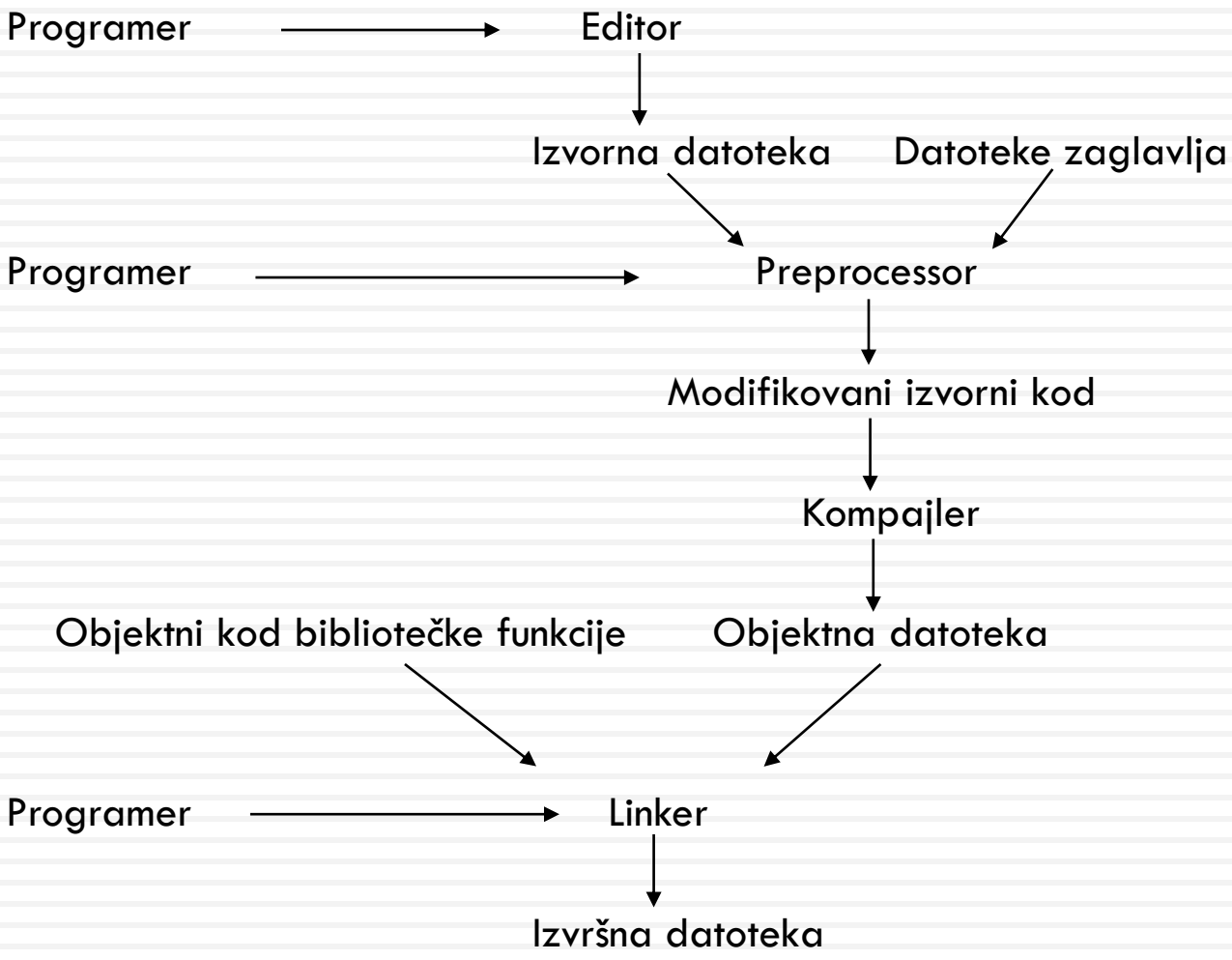


- U prvoj liniji se zahteva da se uključe informacije o standardnoj ulazno-izlaznoj biblioteci (*STandarD Input-Output library*).
- Izvršni kod programa je blok naredbi unutar vitičastih zagrada koji sledi posle zaglavlja *void main(void)*. Jedina linija bloka je poziv *printf* funkcije iz standardne ulazno-izlazne biblioteke koja vrši formatirani ispis.
- *\n* nalaže *printf* funkciji da se nakon ispisivanja poruke Zdravo, Svete! pređe u novi red. Naredni ispis bi se vršio u novom redu.

Elementi C programa

- Razvojno okruženje uključuje
  - Sistemske biblioteke i odgovarajuće datoteke zaglavlja
  - Izvorni kod i njegove datoteke zaglavlja
  - Kompajler generiše izvorni kod u mašinski-čitljiv kod (objektni kod)
    - Objektni kod koji generiše kompajler uglavnom ima “rupe” (nedostajuće delove)
  - Linker povezuje generisani objektni kod sa dodatnim objektnim kodovima i formira izvršni kod

C kompajler


Programer
Editor
Izvorna datoteka
Datoteke zaglavlja
Programer
Preprocessor
Modifikovani izvorni kod
Kompajler
Objektni kod bibliotečke funkcije
Objektna datoteka
Programer
Linker
Izvršna datoteka

Struktura C programa

- Programski jezik C
  - Jezik slobodne forme (free form language)
  - Dozvoljeno je umetanje neograničenog broj praznina i praznih linija u izvornom kodu
    - Više naredbi u jednoj liniji izvornog koda
    - Jedna naredba u više linija izornog koda
    - Kako ustanoviti gde je kraj jedne i pocetak druge naredbe?
      - Naredba C jezika se završava znakom ;
  - Poželjno je na taj način izvorni kod učiniti preglednim
- Forma prostog C programa

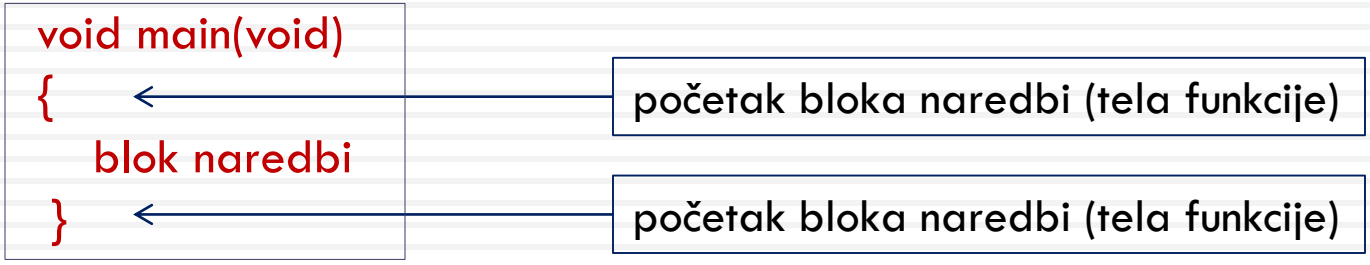
```
preprocesorske direktive
void main(void)
{
    blok naredbi
}
```

Struktura C programa

- Preprocesorske direktive
  - obezbeđuje se kontrolisana modifikacija izvornog koda.
  - počinju znakom # i smeštaju se u jednu liniju izvornog koda.
- Funkcija *main*
  - C program se sastoji iz funkcija
    - Procedure, potprogrami u drugim programskim jezicima
    - Funkcija sa imenom *main* je obavezna. Ona se poziva kada se pokrene izvršavanje programa
- Naredbe
  - Specificiraju akciju koju treba izvesti kada se pokrene izršavanje programa
  - Zaršavaju se znakom *;*

Struktura C programa

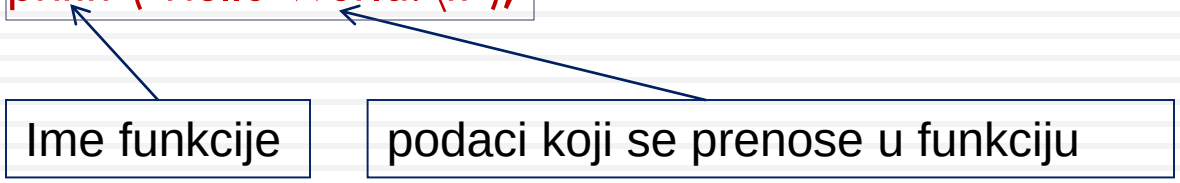
- Šta predstavlja void main(void)?
  - Zaglavlje funkcije main
    - Izvršavanje programa počinje od prve naredbe *main* funkcije
  - Prvo void: ne prenose se nikakve informacije iz funkcije *main* operativnom sistemu
  - Drugo void: ne prenose se nikakve informacije iz operativnog sistema u funkciju main
- Šta predstavljaju {,}?
  - Označavaju početak odnosno kraj bloka naredbi



Štampanje niza znakova

- Poziv *printf* funkcije u najprostijem obliku

```
printf (“Hello World!\n”);
```



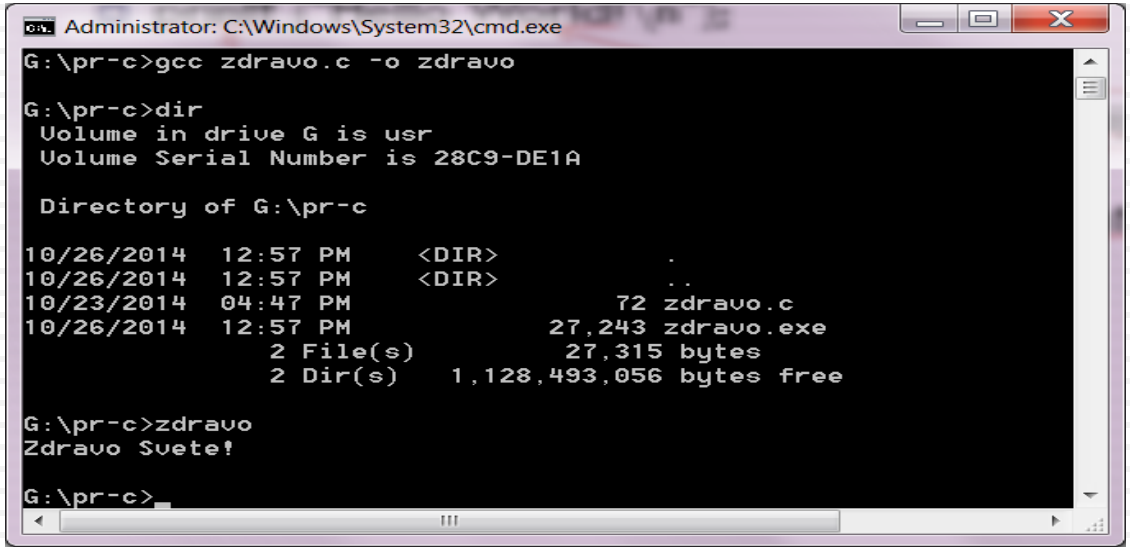
  - Prepisuje niz karaktera između dvostrukih navodnika na ekran
  - Prelazak u novi red se ne vrši automatski
  - Svaki sledeći poziv *printf* funkcije nastavlja ispis započet prethodnim pozivom. Sledeće ima isti efekat

```
printf (“Hello ”);
printf (“World!\n”);
```

  - Ispis \n (new-line symbol) za efekat ima prelazak u novi red

Štampanje niza znakova

- Korisnik ⟶ OS ⟶ main ⟶ printf ⟶ OS ⟶ Ekran



Komentari

- Obezbeđuje se dokumentovanost programa
- Počinje sa /* i zavrsava se sa */

```
/* Ovo je komentar */
```

- Može se umetati svuda u programu
  - u više linija ili
  - u istoj liniji sa programskim tekstom
- Nisu dozvoljeni ugnježdeni komentari

```
/*
      /* GREŠKA */
*/
```

- Standard C99 uvodi novi tip komentara
  - počinje sa // i zavrsava se sa krajem linije u kojoj je započet

```
//Ovo je komentar
```

Elementi jezika C

- Azbuka
  - Mala i velika slova engleske abecede:
    - a b c … x y z A B C … X Y Z
  - cifre dekadnog brojnog sistema:
    - 0 1 2 … 9
  - znaci:
    - _ ! ? . , : ; ‘ = + - / * # % & \ | ( ) [ ] { } < >
- Leksički simbol (token)
  - Niz znakova azbuke koji ima značenje

Elementi jezika C

- printf("Zdravo, Svete!\n");
  1. printf
  2. (
  3. “Zdravo, Svete!\n”
  4.)
  5. ;
- Tokeni u C jeziku
  - identifikatori
  - službene reči
  - konstante
  - operatori
  - separatori

Elementi jezika C

- Identifikatori
  - Imenuju se pojedini objekti programa (promenljive, konstante, funkcije …)
  - Niz slova i cifara koji počinje slovom.
  - Znak _ se tretira kao slovo
  - Mala i velika slova se razlikuju
- Službene (ključne) reči
  - Standardom C jezika se utvrđuje skup identifikatora sa unapred definisanim značenjem

Podaci, tipovi podataka
Konstante i promenljive

- Podatak
  - Predmet obrade
  - Osobine pojedinog podatka određuju tip podatka
- Tip podataka
  - Skup vrednosti koje podaci tog tipa mogu imati
  - Skup operacija nad datim skupom vrednosti
- Konstanta
  - Podatak koji se predstavlja vrednošću odgovarajućeg tipa podataka
- Promenljiva
  - Imenovani podatak
  - Pridružena memorijska lokacija
  - Mogu menjati vrednost tokom izvršavanja programa
  - Ime promenljive je identifikator (niz slova i cifara ….)

Podaci, tipovi podataka
Konstante i promenljive

- Simbolička konstanta
  - Imenovani podatak koji ne može menjati svoju vrednost
  - Nema pridruženu memorijsku lokaciju
  - Uvodi se preprocesorskom direktivom *#define*

```
#define JEDAN 1
```

  - Preporuka
    - Ime simboličke konstante velikim slovima

Promenljive, deklaracija promenljive

- Promenljiva
  - Imenovani podatak koji može menjati svoju vrednost
  - Mora joj se pridružiti tip podataka
    - Pored skupa vrednosti i skupa operacija, tipom podataka se utvrđuje
      - Veličina memorijskog prostora za reprezentaciju vrednosti
      - Način reprezentacije vrednosti
  - Neki od osnovnih tipova podataka jezika C
    - int: celobrojni tip podataka (integer)
    - *float*: razlomljeni tip podataka (floating point)
- Naredba deklaracije promenljive
  - Pridruživanje tipa podataka promenljivoj

```
int godina;
float prosek;
```

Promenljive, deklaracija promenljive

- Više deklaracija se može kombinovati u jednu

```
int i, j, k;
float p, q, r;
```

- Promenljive se pre korišćenja moraju deklarisati
  - Struktura tela funkcije

```
{
  deklaracije promenljivih
  naredbe
}
```

Naredba dodeljivanja

- Promenljivoj se dodeljuje vrednost naredbom dodeljivanja
  - Prvo se navodi ime promenljive zatim znak = i na kraju vrednost (izraz) odgovarajućeg tipa
    - Vrednost int tipa
      - Niz cifara dekadnog brojnog sistema
      - Vodeći karakter može biti znak vrednosti +,-
    - Vrednost float tipa
      - Niz cifara dekadnog brojnog sistema koji sadrži decimalnu tačku . i koji se završava slovom f (odnosno F)
      - Vodeći karakter može biti znak vrednosti +,-

```
godina=2014;
prosek=+9.42f;
```

Štampanje vrednosti promenljive

- Ispisati vrednost celobrojne promenljive godina u sledećem formatu:

  Godina: ___

  ```
  printf("Godina: %d\n", godina);
  ```

  - sekvenca *%d* rezerviše prostor za celobrojnu vrednost u poruci koja se štampa
  - vrednost kojom se popunjava rezervisani prostor se navodi kao sledeći argument funkcije printf.
  - Sekvenca %f se koristi za ispis vrednosti razlomljenog tipa

    ```
    printf("Prosek: %f\n", prosek);
    ```

    - Razlomljena vrednost se standardno ispisuje sa šest decimala
    - Ispis razlomljene vrednosti sa npr. dve decimale se specificira sa *%.2f*

      ```
      printf("Prosek: %.2f\n", prosek);
      ```

Inicijalizacija promenljive

- Pre korišćenja promenljivoj se mora dodeliti vrednost
  - Naredbom dodeljivanja
  - Zajedno sa deklaracijom

```
int godina=2014;
```

    - Slično kao u slučaju deklaracije, inicijalizacija više promenljivih istog tipa se može kombinovati

```
int i=1,j,k=0;
```

Učitavanje ulaznih podataka

- Standardna ulazno-izlazna biblioteka (stdio)
  - printf: formatirani izlaz
  - scanf: formatirani ulaz
    - Zadaje se tip vrednosti koji se učitava
      - %d: učitava se celobrojna vrednost
      - %f: učitava se razlomljena vrednost
    - Memorijska lokacija u koju se upisuje učitana vrednost
      - Navođenjem znaka & pre imena promenljive specificira se memorijska lokacija pridružena promenljivoj sa datim imenom

```
scanf("%d", &godina);
scanf("%f", &prosek);
```

printf funkcija

- printf: formatirani ispis niza znakova

```
printf("niz_znakova", vrednost1, vrednost2, ...)
```

  - vrednost1, vrednost2, … se prema zadanom formatu umeću u niz znakova koji se ispisuje
  - niz_znakova čine
    - Obični karakteri: samo se prepisuju
    - Specifikatori konverzije
      - Počinju znakom %
      - Rezerviše se prostor za ispis odgovarajuće vrednosti

Specifikator konverzije printf funkcije

- %[širina][.preciznost]tip_konverzije
  - tip_konverzije
    - Specificira kako konvertovati vrednost iz interne forme (binarni zapis) u izlaznu formu (tekst)
    - %d konverzija celobrojne vrednosti iz binarnog zapisa u decimalni zapis
    - %f konverzija razlomljene vrednosti iz binarnog zapisa u decimalni zapis
  - širina
    - minimalni broj karaktera koji će se na datoj poziciji rezervisati
      - Ako konvertovana vrednost ima više karaktera odgovarajuće polje se automatski proširuje
      - Ako konvertovana vrednost ima manje karaktera odgovarajuće polje se dopunjuje prazninama sleva (desno poravnanje)
  - preciznost
    - U slučaju celobrojne konverzije: minimalni broj cifara koji se ispisuje (podrazumevano 1)
    - U slučaju razlomljene konverzije: broj cifara iza decimalne tačke (podrazumevano 6)

Specifikator konverzije printf funkcije

- Specifikator konverzije može kao vodeći imati još jedan parameter, koji je kombinacija sledećih znakova:
  - -: izvršiće se levo poravnanje (podrazumevano je desno poravnanje)
  - +: pozitivne brojne rednosti se ispisuju sa znakom + (podrazumevano je da se znak pozitivnih brojnih vrednosti ne ispisuje)
  - □ (praznina): pozitivne brojne vrednosti se ispisuju sa znakom □ (podrazumevano je da se znak pozitivnih brojnih vrednosti ne ispisuje). Ignoriše se ako je zadan i + znak
  - 0: u slučaju desnog poravnanja odgovarajuće polje se dopunjava znakovima 0 sleva (podrazumevano je dopunjavanje prazninama), ako se zahteva levo poravnanje ignoriše se (nikad se ne dodaju 0 sdesna).

Escape sekvence

- Sekvence koje obezbeđuju poseban efekat prilikom njihovog ispisa
  - počinju karakterom \
  - Tretiraju se kao jedan karakter
    - \a: Alert
    - \b: Backspace
    - \n: New line
    - \t: Horizontal tab
    - \\: Znak \
    - \': Znak ‘
    - \": Znak “

scanf funkcija

- scanf: formatirani ulaz podataka

```
scanf("niz_znakova", &promenljiva1, &promenljiva2, ...)
```

  - niz_znakova: uparuje se sa znakovima ulaznog teksta i čine ga:
    - Obični karakteri: uparuju se sa odgovarajućim karakterima ulaznog teksta
      - □: belina se uparuje sa 0 ili više belina u ulaznom tekstu
      - Karakter koji nije belina: uparuje se sa istim karakterom u ulaznom tekstu
    - Specifikatori konverzije
      - Počinju znakom %
      - Uparuju se sa ulaznim podacima odgovarajućeg tipa
        - Uparena vrednost se upisuje u memorijsku lokaciju pridruženu odgovarajućoj promenljivoj iz liste (vrši se konverzija u interni zapis)
  - U slučaju neuspešnog uparivanja, prekida se izvršavanje scanf funkcije

Specifikator konverzije scanf funkcije

- %[*][širina]tip_konverzije
  - tip_konverzije
    - Za svaki specifikator konverzije *scanf* funkcija pokušava da u ulaznom tekstu pronađe podatak koji je zahtevanog tipa i pri tome preskače beline
    - Podatak se formira sa svim karakterima koji su prihvatljivi za zahtevani tip podataka
    - %d: uparuje se sa nizom cifara koji opciono počinje znakom +,-
    - %f: uparuje se sa nizom cifara koji
      - opciono počinje znakom +,-
      - može sadržati decimalnu tačku
      - opciono se završava eksponentom koji čine
        - slovo e (E)
        - niz cifara koji opciono počinje znakom +,-

Specifikator konverzije scanf funkcije

- *: izvršiće se traženo uparivanje, ali se odgovarajuća vrednost odbacuje
- širina: maksimalni broj karaktera koji će se koristiti za zadati tip konverzije

Operatori i izrazi

- Izraz
  - formula kojom se zadaje način kako izračunati neku vrednost
  - Najprostiji izrazi su promenljive i konstante
  - Složeni izrazi se dobijaju primenom operacija nad operandima koji su i sami izrazi
- Operatori
  - Aritmetički
  - Relacioni
  - Logički

Aritmetički operatori

- Unarni: +, -
- Binarni: +, -, *, /, %
  - U slučaju binarnih operatora +, -, *, /
    - Odgovarajući operandi (podizrazi) mogu biti
      - celobrojni: dobijeni izraz je celobrojni
      - razlomljeni: dobijeni izraz je razlomljen
      - Kombinacija celobrojnog i razlomljenog podizraza: dobijeni izraz je razlomljen
  - Binarni operatori /, %
    - % zahteva celobrojne operande
    - Desni operand mora biti različit od 0
    - Vrednost izraza u slučaju negativnih operanada nije očigledan
      - -9/7 može biti
        - -2 (zaokruživanje prema -∞) odnosno
        - -1 (zaokruzivanje prema 0)
      - -9%7 može biti
        - -2 (-9 = 7*(-1) + (-2)) odnosno
        - 5 (-9 = 7*(-2) + 5)

Prioritet operatora

- Kako izračunati vrednost izraza 2+3*4?
  - (2+3)*4
  - 2+(3*4)
- Korišćenjem zagrada
- Uvodi se prioritet operatora
  - Najvišeg prioriteta su unarni operatori + i –, zatim
  - operatori *, /, % i najnižeg prioriteta su
  - binarni operatori + i –

i+j*k → i+(j*k)
-i*-j → (-i)*(-j)
+i+j/k → (+i)+(j/k)

Asocijativnost operatora

- Kako izračunati vrednost izraza kada su operatori istog prioriteta 2-3-4?
  - (2-3)-4
  - 2-(3-4)
- Uvodi se asocijativnost operatora
  - unarni operatori + i – su desno asocijativni
  - Binarni operatori +, -, *, /, % su levo asocijativni

i-j-k ⟶ (i-j)-k
i*j/k ⟶ (i*j)/k
-+i ⟶ -(+i)

Operator dodeljivanja

- Konstrukcija v=e (v promenljiva, e izraz)
  - Izraz dodeljivanja u C jeziku (u drugim programskim jezicima to je naredba dodeljivanja)
  - =: operator (prostog) dodeljivanja
  - Vrednost od e se dodeljuje promenljivoj v
    - Ako promenljiva v i izraz e nisu istog tipa izvršiće se konverzija vrednosti izraza e u odgovarajuću vrednost tipa promenljive v
  - Vrednost izraza dodeljivanja v=e je vrednost koja je dodeljena promenljivoj v

```
int i;
i=1.2345f
```

Vrednost izraza je 1 (to je vrednost koja će se dodeliti promenljivoj i)

Bočni efekat

- Nije uobičajeno da operator menja vrednost svojih operanada
  - i+j: operator + ne menja vrednost promenljivih i, j
  - Operator = menja vrednost svog levog operanda
    - Izraz i=0 ima vrednost 0, ali samo izračunavanje vrednosti izraza ima bočni (dodatni) efekat
      - nova vrednost promenljive i je 0

Operator dodeljivanja

- = je operator
  - i=j=k=0 je složen izraz
  - Operator = je desno asocijativan

i=j=k=0 → i=(j=(k=0))

```
int i;
float f;
i=f=33.3f
```

f ← 33.3
i ← 33

```
f=i=33.3f
```

i ← 33.3
f ← 33

L-vrednost

- Operatori se primenjuju nad operandima koji su uglavnom
  - promenljive
  - konstante ili
  - Izrazi
- Operator dodeljivanja = zahteva takozvanu l-vrednost kao svoj levi operand
- L-vrednost: objekat koji može biti levi operand operatora dodeljivanja
  - objekat čija se vrednost čuva u memoriji
  - objekat čija se vrednost može menjati (promenljiva)
- Neispravni izrazi dodeljivanja

```
12=i
i+j=0
-i=j
```

Složeno dodeljivanje

- Čest slučaj
  - Postojeća vrednost promenljive se koristi da bi se formirala nova vrednost

    i=i+2 ⟶ i+=2
    - Uveden je složeni operator dodeljivanja +=
      - Vrednost levog operanda se uvećava za vrednost desnog operanda
    - Analogno su uvedeni -=, *=, /=, %=

      v op= e ⟶ v = v op e gde je op neki od +,-, *, /, % (ne uvek)
- Kao i operator prostog dodeljivanja = i operatori složenog dodeljivanja imaju desnu asocijativnost

  i+=j+=k ⟶ i+=(j+=k)

Inkrement i dekrement operatori

- Čest slučaj
  - Postojeća vrednost promenljive se uvećava (umanjuje) za 1

i=i+1 ⟶ i+=1

  - Za još kompaktniji zapis uvedeni su unarni operatori ++,--
    - Bočni efekat: vrednost operanda se uvaćava (umanjuje) za 1
    - Koriste se na dva načina
      - prefiksni
        - Vrednost pripadajućeg operanda se uveća odnosno umanji za 1 i tako izmenjena vrednost je vrednost izraza
        - Dakle, prvo se vrednost operanda menja, a zatim koristi
      - postfiksni
        - Obratno, prvo se vrednost pripadajućeg operanda koristi a zatim menja

Prioritet i asocijativnost operatora

1.	Postfiksni ++, --	leva asocijativnost
2.	Prefiksni ++, --	
	Unarni +,-	desna asocijativnost
3.	*,/,%	leva asocijativnost
4.	Binarni +,-	leva asocijativnost
5.	=, +=,-=,*=,/=,%=	desna asocijativnost

Redosled izračunavanja podizraza

- Redosled izračunavanja podizraza nije utvrđen standardom C jezika
  - (a+b)*(c-d)
    - Da li se prvo izračunava podizraz (a+b) ili podizraz (c-d)
  - Ako podizrazi imaju bočne efekte (operatori dodeljivanja inkrement i dekrement operatori) vrednost takvih izraza nije očigledna

```
int a=5, b, c;
c=(b=a+2)-(a=1)
```

    - Ako se prvo izračunava (a=1), c ⟵ 2
    - Ako se prvo izračunava (b=a+2), c ⟵ 6

Naredba izraza

- Svaki izraz postaje naredba C jezika ako se završava znakom ;

```
++i;
```

- Naredba izraza nema smisla ako izraz nema neki bočni efekat

```
i*j-1;
```