

OSNOVI PROGRAMIRANJA

Dr Dinu Dragan, dinud@uns.ac.rs

Dr Dušan Gajić, dusan.gajic@uns.ac.rs



UPRAVLJAČKE STRUKTURE

Programske upravljačke strukture

- Programske upravljačke strukture su mehanizmi za definisanje redosleda izvođenja operatora i izraza nad datim tipovima podataka ...
- Rešenje bilo kog problema, koji je po svojoj prirodi rešiv pomoću računara, može se izraziti kao superpozicija sledećih struktura:
 - **sekvence,**
 - **selekcije i**
 - **iteracije**

Iskazi selekcije

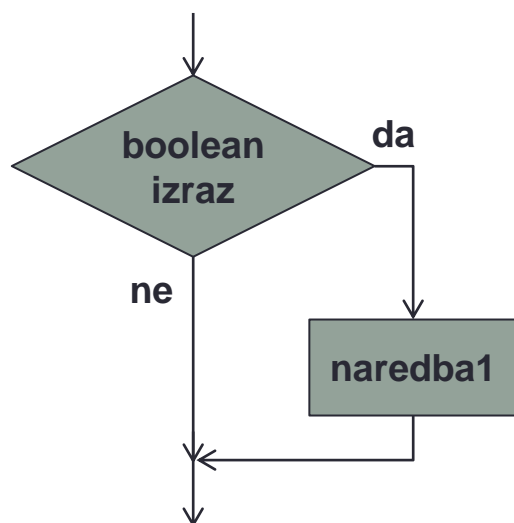
- Omogućava da se određene akcije naredbe izvrše uslovno u zavisnosti od vrednosti određenog iskaza
- Iskazi selekcije (grananje):
 - **if**
 - **switch**

if selekcija ...

- Koju granu (od moguće dve) izvršiti?

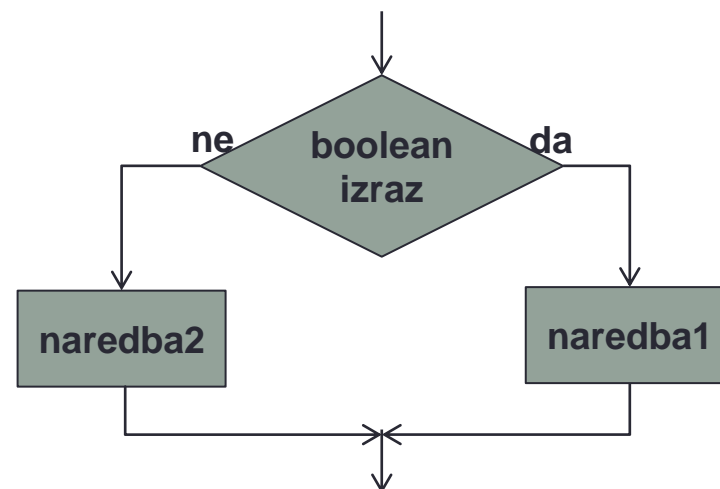
Nepotpuna if selekcija

Ako je **izraz** tačan
naredba1 se izvršava



Potpuna if selekcija

Ako je **izraz** tačan
naredba1 se izvršava
u suprotnom, ako je **izraz** netačan
naredba2 se izvršava



... if selekcija ...

- Kako to izgleda u Java kodu:

Nepotpuna if selekcija

```
if (boolean_izraz)  
    naredba1
```

Potpuna if selekcija

```
if (boolean_izraz)  
    naredba1  
else  
    naredba2
```

... if selekcija

- Ili blok naredbi:

Nepotpuna if selekcija

```
if (boolean_izraz)
{
    naredba11
    ...
    naredba1n
}
```

Potpuna if selekcija

```
if (boolean_izraz)
{
    naredba11
    ...
    naredba1n
}
else
{
    naredba21
    ...
    naredba2m
}
```


Ugnježdavanje if selekcije

- Kako rešiti sledeći problem:
Ako je $x > 3$ prikaži “gore” a ako je $x < 0$ prikaži “dole”

```
if (x >= 0)
    if (x > 3)
        System.out.print("gore");
    else
        System.out.print("dole");
```

```
if (x >= 0) {
    if (x > 3)
        System.out.print("gore");
}
else
    System.out.print("dole");
```

```
if (x > 3)
    System.out.print("gore");
else
    if (x < 0)
        System.out.print("dole");
```

- else** se uvek vezuje za poslednji **if**

Ugnježdavanje if selekcija ...

- Može se napraviti ugnježdavanje if selekcija:

Ugnježdene if selekcije

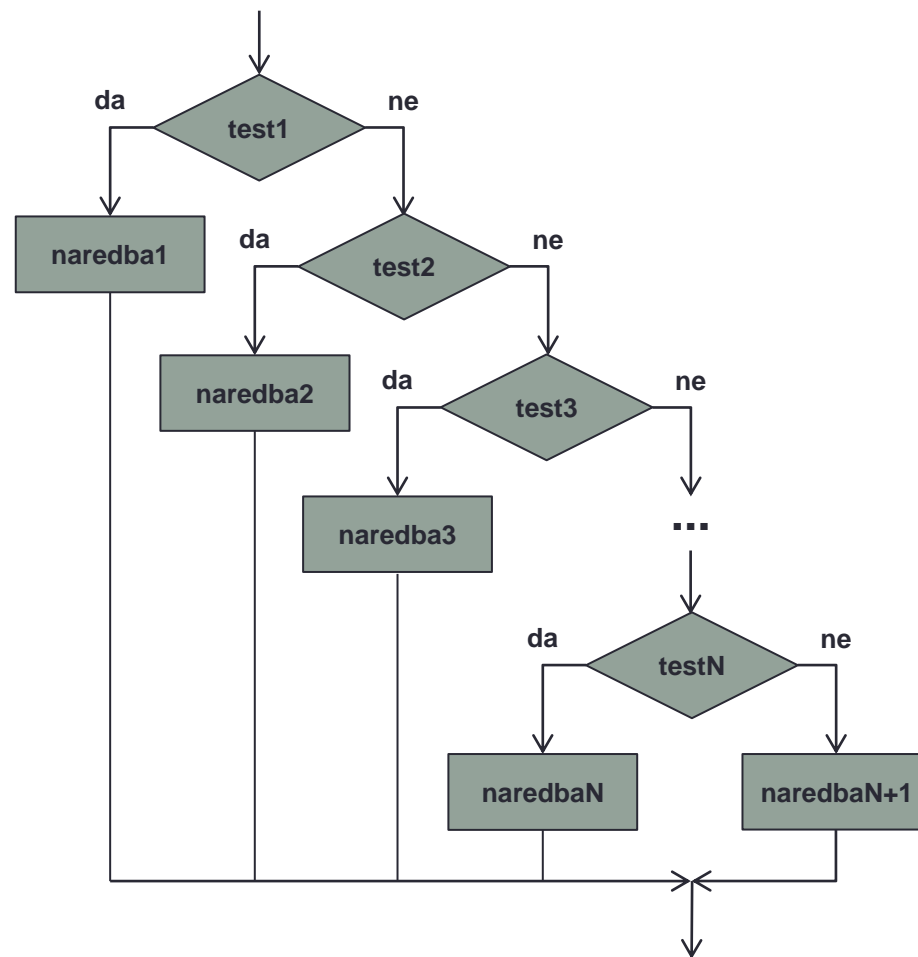
```
if (boolean_izraz1)
    naredba1
else
    if (boolean_izraz2)
        naredba2
    else
        naredba3
```

Ugnježdene if selekcije (kompaktan zapis)

```
if (boolean_izraz1)
    naredba1
else if (boolean_izraz2)
    naredba2
else
    naredba3
```

... Ugnježdavanje if selekcija

```
if (test1)
    naredba1
else if (test2)
    naredba2
else if (test3)
    naredba3
    ...
else if (testN)
    naredbaN
else
    naredbaN-1
```



if selekcija – primer

```
/* *****  
 * Primer programa za ispis uspeha na osnovu unete ocene (verzija sa visestrukom if selekcijom).  
 ***** */  
public class IfSelekcijaPrimer {  
    public static void main(String[] args) {  
        /* Definicija promenljive */  
        int ocena;  
  
        /* Unos podatka */  
        System.out.println("Program za ispis uspeha na osnovu unete ocene!");  
        System.out.print("Unesite ocenu: ");  
        ocena = TextIO.getLnInt();  
  
        /* Odredjivanje uspeha i ispis */  
        if (ocena == 1)  
            System.out.println("Nedovoljan!");  
        else if (ocena == 2)  
            System.out.println("Dovoljan!");  
        else if (ocena == 3)  
            System.out.println("Dobar!");  
        else if (ocena == 4)  
            System.out.println("Vrlo dobar!");  
        else  
            System.out.println("Odlican!");  
    }  
}
```

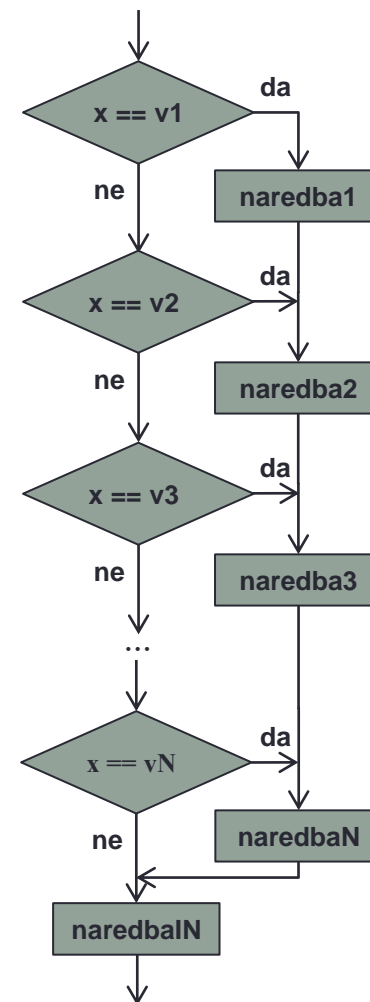
if selekcija – zadaci

- Šta ako uneti broj nije iz opsega 1 - 5?
 1. Dopuniti prethodni zadatak sa proverom da li je uneti broj unutar odgovarajućeg opsega
 2. Dodati unutar if selekcije kod koji reaguje na broj van opsega
- Implementirati program koji ispisuje koji je od dva broja veći (odrediti tipove podataka, potrebnu proveru, ...)

switch selekcija

- Koju granu (od više njih) izvršiti?
- Izbor se može vršiti samo na osnovu: short, int, byte, char, boolean, i String tipova

```
switch (a)
{
    case v1: naredba1;
    case v2: naredba2;
    case v3: naredba3;
    ...
    case vN: naredbaN;
    default: naredbaIN;
};
```



switch selekcija – primer 1

```
/* *****  
 * Primer programa za ispis uspeha na osnovu unete ocene (switch verzija).  
 ***** */  
public class IfSelekcijaPrimer {  
    public static void main(String[] args) {  
        /* Definicija promenljive */  
        int ocena;  
  
        /* Unos podatka */  
        System.out.println("Program za ispis uspeha na osnovu unete ocene!");  
        System.out.print("Unesite ocenu: ");  
        ocena = TextIO.getLnInt();  
  
        /* Odredjivanje uspeha i ispis */  
        switch (ocena){  
            case 1: System.out.println("Nedovoljan!");  
            case 2: System.out.println("Dovoljan!");  
            case 3: System.out.println("Dobar!");  
            case 4: System.out.println("Vrlo dobar!");  
            case 5: System.out.println("Odlican!");  
        }  
    }  
}
```

- Kako naterati da se izvršava samo jedan case?

break

- Kada se naiđe na **break**, blok naredbi se napušta bez obzira na trenutnu vrednost
- Program nastavlja sa izvršavanjem prve naredbe iza bloka
- Ukoliko se pozove u ugnježdenim blokovima, **break** izlazi samo iz unutrašnjeg bloka
- Koristiti vrlo obazrivo i samo u krajnjoj nuždi

switch selekcija – primer 3

```
/* *****  
 * Primer programa za ispis uspeha na osnovu unete ocene (switch verzija u kombinaciji sa break i default).  
***** */  
public class SwitchSelekcijaPrimer {  
    public static void main(String[] args) {  
        /* Definicija promenljive */  
        int ocena;  
  
        /* Unos podatka */  
        System.out.println("Program za ispis uspeha na osnovu unete ocene!");  
        System.out.print("Unesite ocenu: ");  
        ocena = TextIO.getlnInt();  
  
        /* Odredjivanje uspeha i ispis */  
        switch (ocena){  
            case 1: System.out.println("Nedovoljan!");  
                    break;  
            case 2: System.out.println("Dovoljan!");  
                    break;  
            case 3: System.out.println("Dobar!");  
                    break;  
            case 4: System.out.println("Vrlo dobar!");  
                    break;  
            case 5: System.out.println("Odlican!");  
                    break;  
            default: System.out.println("Ocena mora biti izmedju 1 i 5.");  
        }  
    }  
}
```

switch selekcija i inicijalizacija ...

- Ako se unutar switch-a inicijalizuje promenljiva (šta znači inicijalizacija), može doći do greške u kodu ako se upotrebi ovako kod (isti problem može se javiti i prilikom ugnježdavanja if selekcije)

```
int status;  
String potez;  
status = TextIO.getlnInt();  
switch (status){  
    case 0:  
        potez = "pomerio gore";  
        break;  
    case 1:  
        potez = "pomerio dole";  
        break;  
    case 2:  
        potez = "pomerio levo";  
        break;  
    case 3:  
        potez = "pomerio desno";  
}  
System.out.println("Igrac se "+potez); // Greska!
```

... switch selekcija i inicijalizacija

- Do greške je došlo jer promenljiva nije inicijalizovana
- Ili definisati promenljivu ili koristiti default (što garantuje da će za sve vrednosti promenljive **status** promenljiva **potez** dobiti vrednost)

```
int status;
String potez = "nije pomerio";
status = TextIO.getLnInt();
switch (status){
    case 0:
        potez = "pomerio gore";
        break;
    case 1:
        potez = "pomerio dole";
        break;
    case 2:
        potez = "pomerio levo";
        break;
    case 3:
        potez = "pomerio desno";
}
System.out.println("Igrac se "+potez);
```

```
int status;
String potez;
status = TextIO.getLnInt();
switch (status){
    case 0:
        potez = "pomerio gore";
        break;
    case 1:
        potez = "pomerio dole";
        break;
    case 2:
        potez = "pomerio levo";
        break;
    case 3:
        potez = "pomerio desno";
        break;
    default:
        potez = "nije pomerio";
}
System.out.println("Igrac se "+potez);
```

switch selekcija i String

- Isto kao i za int

```
String units;  
double measurement;  
double inches;  
...  
  
switch ( units ) {  
    case "inches":  
        inches = measurement;  
        break;  
    case "feet":  
        inches = measurement * 12;  
        break;  
    case "yards":  
        inches = measurement * 36;  
        break;  
    case "miles":  
        inches = measurement * 12 * 5280;  
        break;  
    default:  
        System.out.println("Wait a minute!  Illegal unit of measure!  I quit!");  
        System.exit(1);  
}
```

switch selekcija i enum

- Slično kao i za ostale tipove jedino što ne ide enum.vrednost, već samo vrednost

```
enum Season { SPRING, SUMMER, FALL, WINTER }  
  
...  
  
Season currentSeason;  
  
...  
  
switch ( currentSeason ) {  
    case WINTER:      // ( Obratiti paznju da nije Season.WINTER ! )  
        System.out.println("December, January, February");  
        break;  
    case SPRING:  
        System.out.println("March, April, May");  
        break;  
    case SUMMER:  
        System.out.println("June, July, August");  
        break;  
    case FALL:  
        System.out.println("September, October, November");  
        break;  
}
```

switch selekcija i tekstualni meni

```
int opcija;

System.out.println("Izaberite jednu od ponudjenih opcija?");
System.out.println();
System.out.println("          1.  Unos studenta");
System.out.println("          2.  Izmena studenta");
System.out.println("          3.  Brisanje studenta");
System.out.println("          4.  Prikaz studenta");
System.out.println();

System.out.println("Vas izbor je: ");
opcija = TextIO.getlnInt();

switch ( opcija ) {
    case 1: System.out.println("Izabrali ste prvu opciju.");
            break;
    case 2: System.out.println("Izabrali ste drugu opciju.");
            break;
    case 3: System.out.println("Izabrali ste trecu opciju.");
            break;
    case 4: System.out.println("Izabrali ste cetvrtu opciju.");
            break;
    default: System.out.println("Niste odabrali ni jednu od ponudjenih opcija");
            System.exit(1);
}
```

Zadatak iz if i switch selekcija

- Implementirati program za računanje kvadratne jednačine:

$$a \cdot x^2 + b \cdot x + c = 0$$

- http://sh.wikipedia.org/wiki/Kvadratna_jednacina
- Promenljive:
a, b, c, d, x1, x2, y1, y2
- Šta ako je **a == 0** ili **b == 0** ?
- Šta ako je **d == 0** ili **d < 0** ili **d > 0** ?
- U zavisnosti od vrste korena prikazati rezultat i odgovarajući tekst (**switch?**)

Iskazi iteracije ...

- Nazivaju se još i **programskim petljama** jer vrte naredbe u krug (petlju)
- **Omogućavaju ponavljanje određenih naredbi ili blokova više puta (telo petlje)**
- Programski jezik Java omogućava **fleksibilne načine** za određivanje broja ponavljanja petlji ili donošenje odluke o izlasku iz petlje
- Koliko puta će se iteracija izvršavati određuje se na osnovu toga da li je određen iskaz (**uslov izvršavanja iteracije**) tačan ili netačan

... Iskazi iteracije

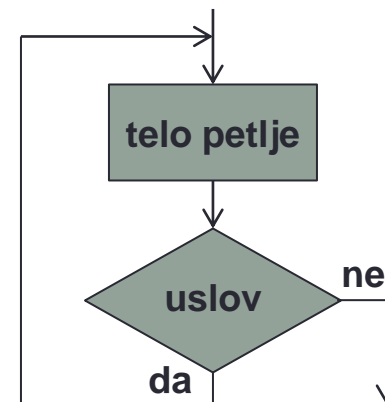
- U zavisnosti od toga da li se uslov izvršavanja iteracije proverava na početku ili na kraju, razlikuje iteracije sa izlaskom na vrhu, na dnu, ili u sredini
- Iskazi iteracije u programskom jeziku Java:
 - **do-while** iskaz (izlazak na dnu)
 - **while** iskaz (izlazak na vrhu)
 - **for** iskaz (izlazak na vrhu)

do while iteracija

- Iteracija sa izlaskom na dnu, uslov iteracije se izračunava nakon tela petlje
- Telo petlje se izvršava **NAJMANJE** jednom, čak i kada je uslov izvršavanja iteracije odmah netačan (jednak nuli)

```
do  
  naredba  
while (uslov);
```

```
do  
{  
  naredba1  
  ...  
  naredbaN  
} while (uslov);
```



- Proširite primer za ispis uspeha na osnovu unete ocene tako da se korisniku ne dozvoli unos ni jednog drugog broja osim onog iz opsega 1 - 5

do while iteracija – primer

```

/*****
* Primer programa za ispis uspeha na osnovu unete ocene (do-while petlja se koristi ya proveru ispravnosti ulazne
* vrednosti).
*****/
public class DoWhilePrimer {
    public static void main(String[] args) {
        int ocena;
        System.out.println("Program za ispis uspeha na osnovu unete ocene!");
        // do-while petlja se koristi da se proverí ulazna vrednost
        do{
            System.out.print("Unesite ocenu: ");
            ocena = TextIO.getInt();
            if ((ocena < 1)|| (ocena > 5))
                System.out.println("Ocena mora biti izmedju 1 i 5.");
        } while ((ocena < 1)|| (ocena > 5));
        switch (ocena){
            case 1: System.out.println("Nedovoljan!");
                    break;
            case 2: System.out.println("Dovoljan!");
                    break;
            case 3: System.out.println("Dobar!");
                    break;
            case 4: System.out.println("Vrlo dobar!");
                    break;
            case 5: System.out.println("Odlican!");
                    break;
        }
    }
}

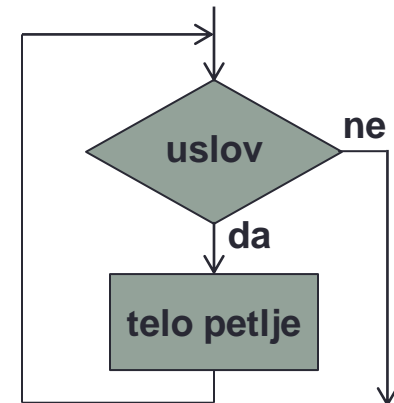
```

while iteracija

- Iteracija sa izlaskom na dnu, uslov se izvršava pre tela petlje
- Telo petlje se izvršava sve dok uslov izvršavanja iteracije ispunjen (jednak jedinici), kada to nije više slučaj izlazi se iz iteracije
- Telo petlje se izvršava nula, jednom ili više puta

```
while (uslov)  
    naredba
```

```
while (uslov)  
{  
    naredba1  
    ...  
    naredbaN  
}
```



- Primer: program za računanje faktoriijela celog broja koji definiše korisnik

while iteracija – primer

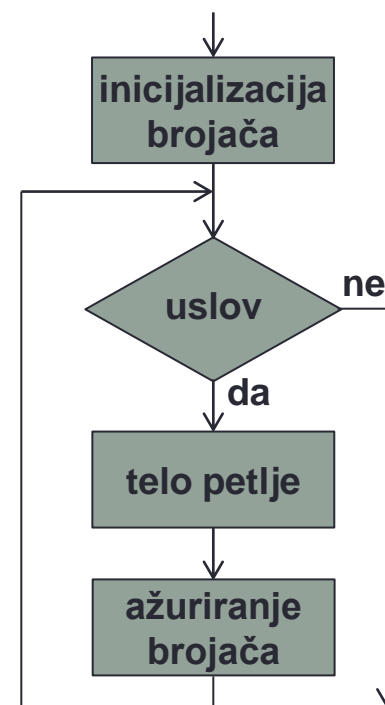
```
/******  
* Primer programa za računanje faktoriijela celog broja koji definiše korisnik.  
*****/  
public class WhilePrimer {  
    public static void main(String[] args) {  
        int broj = 1, i = 1, fakt = 1;  
        System.out.println("Program za racunanje faktoriijala unetog broja.");  
        System.out.println("Unesite broj: ");  
        do{  
            broj= TextIO.getlnInt();  
            if (broj < 0)  
                System.out.println("Broj mora biti veci ili jednak 0. Unesite ponovo broj: ");  
        } while (broj < 0);  
        while (i <= broj){  
            fakt *= i;  
            i++;           //uzrokuje da (i <= broj) konvergira nuli  
        }  
        System.out.println("Faktorijal broja " + broj + " je: " + fakt + ".");  
    }  
}
```

for iteracija ...

- Iteracija sa izlaskom na dnu, uslov se izvršava pre tela petlje
- Telo petlje se izvršava nula, jednom ili više puta
- Slično **while** iteraciji ali se izvršavanje kontroliše brojačkom promenljivom

```
for (inic_br; uslov; ažur_br)  
    naredba
```

```
for (inic.br;uslov;ažur.br){  
    naredba1  
    ...  
    naredbaN  
}
```



... for iteracija

- Kod za prikazivanje prvih 100 prirodnih brojeva:

```
int i;  
for (i = 1; i <= 100; i++){  
    System.out.printf("\n%d\n", i);  
}
```

- Brojač je promenljiva i
- Brojač se inkrementira (ili dekrementira) nakon svake iteracije
- Može i ovako:

```
for (int i = 1; i <= 100; i++){  
    System.out.printf("\n%d\n", i);  
}
```
- Šta su prednosti i mane ovog pristupa?
- Primer: odrediti sumu prvih N prirodnih brojeva, pri čemu se N zadaje na početku programa od strane korisnika

for iteracija – primer

```
/* *****  
 * Primer programa za računanje sume prvih N prirodnih brojeva.  
 ***** */  
public class ForPrimer {  
    public static void main(String[] args) {  
        int N = 1;  
        int i, suma = 0;  
  
        System.out.println("Program za racunanje sume prvih N prirodnih brojeva.");  
        System.out.println("Unesite N: ");  
  
        do {  
            N = TextIO.getLnInt();  
            if (N < 1)  
                System.out.println("N mora biti veci od 0. Unesite ponovo N: ");  
        } while (N < 1);  
  
        for (i = 1; i <= N ; i++)  
            suma += i;  
  
        System.out.println("Suma prvih " + N + " prirodnih brojeva je: " + suma + ".");  
    }  
}
```

Sličnost for i while iteracije

- Postoji ekvivalencija između **for** i **while**
- U Java programskom jeziku **for** se može konvertovati u **while** i obrnuto
- U zavisnosti od prirode problema zavisi i koja je iteracija će se upotrebiti
- Ako nešto treba obaviti određen poznat broj puta koristi se **for**
- Ako se čeka da se neki uslov desi (a to nije unapred određeno brojem ponavljanja) bolje je koristiti **while**
- Primer: odrediti sumu prvih N prirodnih brojeva, pri čemu se N zadaje na početku programa od strane korisnika i koristiti **while** iteraciju

Sličnost for i while iteracije – primer

```
/* *****  
 * Primer programa za računanje sume prvih N prirodnih brojeva primenom while iteracije.  
 * ***** */  
public class ForPrimer {  
    public static void main(String[] args) {  
        int N = 1;  
        int i, suma = 0;  
  
        System.out.println("Program za racunanje sume prvih N prirodnih brojeva.");  
        System.out.println("Unesite N: ");  
  
        do {  
            N = TextIO.getLnInt();  
            if (N < 1)  
                System.out.println("N mora biti veci od 0. Unesite ponovo N: ");  
        } while (N < 1);  
  
        i = 1;           // inicijalizacija brojaca  
        while (i <= N) { // uslov  
            suma += i;   // telo  
            i++;         // azuriranje brojaca  
        }  
  
        System.out.println("Suma prvih " + N + " prirodnih brojeva je: " + suma + ".");  
    }  
}
```

- A kako bi bilo rešiti isti problem primenom do-while iteracije?

Ugnježdavanje for iteracija

- Bilo koja upravljačka struktura može sadržati unutar sebe drugu neku upravljačku strukturu (u prethodnim primerima se unutar **do-while** iteracije nalazila **if** selekcija)
- Tako se, u zavisnosti od problema, unutar **for** iteracija može naći **while**, **do-while**, i sl.
- Java ne nameće nikakvo ograničenje po pitanju broja ugnježđenih upravljačkih kontrola
 - Sa praktične strane, pak, previše ugnježđenih struktura je teško ispratiti, razumeti i testirati

Ugnježdavanje for iteracija

- Ugnježdene **for** upravljačke strukture se često koriste u za rešavanje nekoliko vrsta problema (prolazak kroz elemente matrice, sortiranje, ...)
- Kod ugnježdenih **for** upravljačkih struktura mora se voditi računa da svaka **for** iteracija mora imati svoj brojač (ne može se upotrebiti isti brojač, jer će unutrašnja iteracija pokvariti stanje brojača spoljašnjeg brojača)

```
for (i = 0; i < br_vrst; i++)      - spoljašnja iteracija  
    for (j = 0; j < br_kolona; j++) - unutrašnja iteracija
```

- Zadatak: implementirati program koji preuzima jednu liniju teksta od korisnika i ispisuje koja se sve slova nalaze u tom tekstu i koliko ima ukupno različitih slova

Ugnježdavanje for iteracija – primer...

```
/* *****  
 * Primer programa koji preuzima jednu liniju teksta od korisnika i ispisuje koja se sve slova nalaze u tom tekstu  
 * i koliko ima ukupno različitih slova.  
 * ***** */  
public class IzlistavanjeSlova {  
    public static void main(String[] args) {  
        String str; // linija teksta koju unosi korisnik.  
        int ukSlova; // ukupan broj različitih slova u str tekstu.  
        char slovo; // Slovo iz alfabeta.  
        int i;      // indeks slova u nizu karaktera string str.  
  
        System.out.println("Unesite jedan red teksta:");  
        str = TextIO.getLn();  
        str = str.toUpperCase();  
  
        ukSlova = 0;  
  
        System.out.println("Vas red teksta sadrzi sledeca slova:");  
        System.out.println();  
        System.out.print("    ");
```

...Ugnježdavanje for iteracija – primer

```
for ( slovo = 'A'; slovo <= 'Z'; slovo++ ) {  
    for ( i = 0; i < str.length(); i++ ) {  
        if ( slovo == str.charAt(i) ) {  
            System.out.print(slovo);  
            System.out.print(' ');  
            ukSlova++;  
            break;  
        }  
    }  
}  
  
System.out.println();  
System.out.println();  
System.out.println("Otkriveno je " + ukSlova + " razlicitih slova.");  
}
```

continue

- **continue** naredba nastavlja izvršavanja iteracije sa narednim prolazom uz preskakanje svih naredbi tela tekuće iteracije (while, for i do while)

```
for(i=0;i < N;i++)  
{  
    naredba1; naredba2;  
    if ((X[i] % 2) != 0)  
        continue;  
    naredba3; naredba4;  
}
```

pocetak tela for iteracije

i-ta vrednost nije deljiva sa 2

na sledeci prolaz ako i < N

- Program nastavlja izvršavanje ulaskom u naredni ciklus
- Isti efekat može da se postigne korišćenjem **IF-ELSE** selekcije

Upravljačke strukture – zadaci

- Napisati program koji određuje prosečan broj studenata na godini. Korisnik unosi student po smerovima sve dok ne unese vrednost 0. Broj studenata na smeru je u intervalu 1..160.
- Napisati program koji određuje ukupan broj brojeva deljivih sa 7. Korisnik unosi brojeve sve dok ne unese vrednost -1. Uneti broj mora biti u intervalu od 6..142.
- Napisati program koji određuje koji je broj najmanji u sekvenci od N brojeva. Korisnik unosi N, a zatim brojeve (-10..100).
- Napisati program koji određuje prosečnu ocena studenta u toku jedne godine. Korisnik unosi ocene studenta (5..10) sve dok ne unese vrednost 0.
- Napisati program koji određuje procenat brojeva deljivih sa 3 od N brojeva. Korisnik unosi N, a zatim same brojeve (15-62).

