



# 3

## Naredbe grananja

- `if, if/else` naredba
- `switch/case` naredba

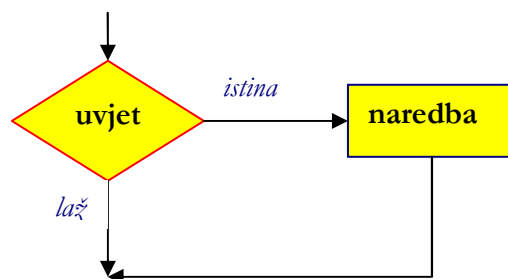
## if, if/else naredba

Metode koje smo do sada pisali su dobivale parametre izvana, uvijek su obavile iste operacije nad parametrima i vraćale rezultate. U praksi su takvi problemi izuzetno rijetki. Prvi sljedeći korak nakon ovakvih problema su problemi kod kojih se ovisno o nekom uvjetu izvršava određeni dio programa. U takvim slučajevima ćemo koristiti neku od naredbi grananja. U ovom ćemo se poglavlju detaljnije upoznati s naredbom **if** i svim njenim oblicima.

Najjednostavniji oblik **if** naredbe je:

```
if (uvjet)
    naredba;
```

Shematski bi se ova naredba mogla prikazati kao:



Slika 3 – 1 Shematski prikaz naredbe grananja

Ukoliko je *uvjet* istinit izvršava se **naredba**, ukoliko je *uvjet* lažan ne događa se ništa, tj. prelazi se na prvu sljedeću naredbu iza **if**.

Ukoliko je **naredba** složena naredba složena, tj. sastoji se od više jednostavnih naredbi, ona se navodi unutar vitičastih zagrada {}.

### Primjer 3 – 1:

Ukoliko je vrijednost varijable *n* jednaka 65 varijabli, vrijednost varijable *c* će postati 'A'.

```
if (n == 65)
    c = 'A';
```

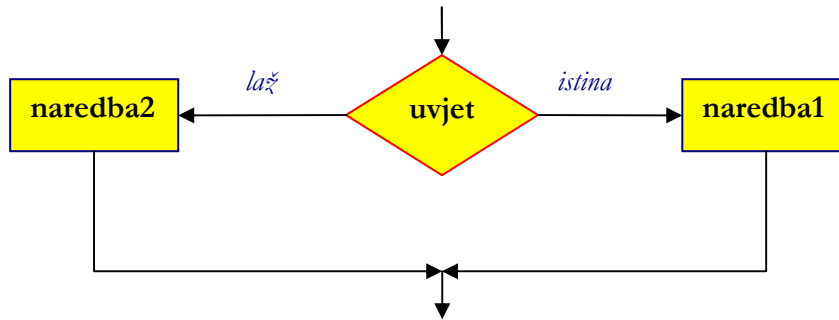
### Napomena:

Ukoliko uvjet ne stavimo unutar zagrada, doći će do greške prilikom kompajliranja.

Osim gore opisanog oblika **if** naredbe, često puta se koristi i tzv. **if/else** oblik. Sintaksa takvog oblika if naredbe je:

```
if (uvjet)
    naredba1;
else
    naredba2;
```

Shematski bi takvu naredbu prikazali na sljedeći način:



Slika 3 – 2 Shematski prikaz **if/else** naredbe

Ukoliko je uvjet istinit izvršava se **naredba1**, inače se izvršava **naredba2**.

### Primjer 3 – 2:

Napišimo metodu kojoj ćemo prosljeđivati koeficijente kvadratne jednadžbe a metoda će vraćati **true** ukoliko jednadžba ima realnih rješenja, inače će vraćati **false**.

#### Rješenje:

Prisjetimo se da će jednadžba imati realnih rješenja ako joj je diskriminanta veća ili jednaka od 0.

Diskriminanta kvadratne jednadžbe  $ax^2 + bx + c = 0$  dana je s  $D = b^2 - 4ac$ .

Dakle rješenje bi bilo:

```

public static boolean RRjesenje (int a, int b, int c)
{
    int D;
    D = b * b - 4 * a * c;
    if (D >= 0)
        return true;
    else
        return false;
}
  
```

### Primjer 3 – 4:

Napišimo metodu koja će unositi iznos bruto plaće u kunama. Procedura treba vraćati iznos poreza na plaću, ako je poznato da je do iznosa od 5000 kn porez 15%, za iznos od 5000 do 10000 je porez 25%, te je za iznos iznad 10000 porez 35%.

#### Rješenje:

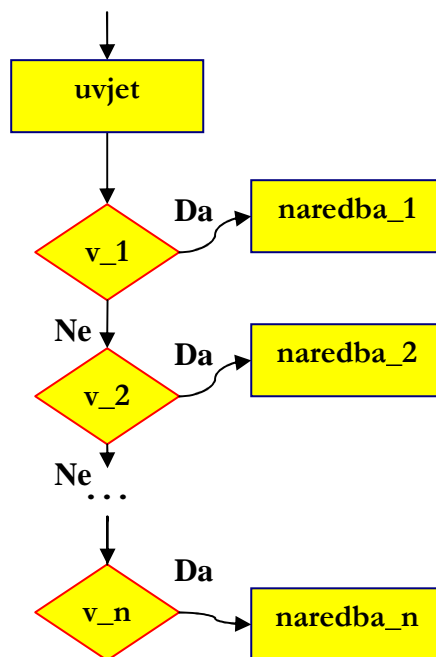
## switch/case naredba

Često puta kod grananja nailazimo na više disjunktih uvjeta. Korištenje **if** naredbe u takvim slučajevima može biti nespretno. Osim **if** naredbe, Java nam, kao i većina ostalih programskih jezika, nudi i naredbu **switch**.

Opći oblik ove naredbe je:

```
switch (izraz)
{
    case v_1: naredba_1;
              break;
    case v_2: naredba_2;
              break;
    ...
    case v_n: naredba_n;
              break;
}
```

Shematski prikaz **switch** naredbe je:



**Slika 3 – 3** Shematski prikaz **switch** naredbe

Pri čemu su:

- **izraz** – bilo koji izraz čija je vrijednost ordinalnog tipa (najčešće cjelobrojni ili znakovni tip)
- **v\_1, v\_2, ... v\_n** – vrijednosti odgovarajućeg tipa (isti kao izraz)
- **naredba\_1, naredba\_2, ... naredba\_n** – naredba koja se izvršava ukoliko vrijednost izraza odgovara odgovarajućoj vrijednosti.

Odnosno, ako je vrijednost izraza **izraz** jednaka **v\_1** izvršava se **naredba\_1**, ukoliko je vrijednost **izraz** jednaka **v\_2** izvršava se **naredba\_2**,...

Naredba **break** općenito služi za prekidanje nekog niza naredbi (najčešće izlazak iz petlje). Možda nije sasvim jasno koja je ovdje uloga naredbe **break**.

Radi se o sljedećem. Ukoliko ne bismo stavili naredbu **break** nakon **naredba\_1**, te ako bi vrijednost izraza **izraz** bila jednaka **v\_1**, izvršila bi se **naredba\_1**, ali i sve naredbe iza nje, **naredba\_2** itd. Stoga koristimo naredbu **break** da bismo, nakon što se odgovarajuća naredba izvrši, izašli iz **switch** naredbe.

Općenito ćemo **switch** naredbu koristiti kada imamo više od 2 uvjeta kod grananja. Korištenje **switch** naredbe uglavnom se može zamijeniti **if** naredbom

#### Napomena:

Primijetimo izraz kod **switch** naredbe uvijek dolazi unutar zagrada – ().  
"Tijelo" **switch** naredbe pišemo unutar vitičastih zagrada – {}.

Prije nego započnemo rješavati sljedeći primjer reći ćemo par riječi o još jednom tipu podataka: **String**. O tipu podataka **String** detaljnije ćemo govoriti u nastavku. Na **String** ćemo u ovom trenutku gledati kao na složeni tip podataka, čije su vrijednosti riječi – niz od jednog ili više znakova. Vrijednosti tipa **String** se pišu unutar dvostrukih navodnika (").

#### Napomena:

Primijetimo da se vrijednosti tipa **String** pišu unutar dvostrukih navodnika ("), za razliku od tipa **char**, gdje se vrijednosti pišu unutar jednostrukih navodnika (').

Deklaracija varijable tipa **String** identična je deklaracije bilo koje druge varijable nekog jednostavnog tipa (**char**, **int**,...).

#### **Primjer 3 – 5:**

Napišimo metodu čiji će ulazni parametar biti jednoznaменkasti prirodan broj n. Metoda treba vraćati broj n ispisan riječima.

#### **Rješenje:**

Dani problem ćemo za usporedbu riješiti na dva načina: pomoću **if** i pomoću **switch** naredbe:

```
//rješenje pomoću if naredbe
public static String broj1(int n)
{
    String s = "";
    if (n == 0)
        s = "Nula";
    else if (n == 1)
        s = "Jedan";
    else if (n == 2)
        s = "Dva";
    else if (n == 3)
        s = "Tri";
    else if (n == 4)
```

```
        s = "Četiri";
    else if (n == 5)
        s = "Pet";
    else if (n == 6)
        s = "Šest";
    else if (n == 7)
        s = "Sedam";
    else if (n == 8)
        s = "Osam";
    else if (n == 9)
        s = "Devet";
    return s;
}

//rješenje pomoću switch naredbe
public static String broj2 (int n)
{
    String s = "";
    switch (n)
    {
        case 0: s = "Nula";
                break;
        case 1: s = "Jedan";
                break;
        case 2: s = "Dva";
                break;
        case 3: s = "Tri";
                break;
        case 4: s = "Četiri";
                break;
        case 5: s = "Pet";
                break;
        case 6: s = "Šest";
                break;
        case 7: s = "Sedam";
                break;
        case 8: s = "Osam";
                break;
        case 9: s = "Devet";
                break;
    }
    return s;
}
```