

## SWITCH naredba

IF naredba omogućava uslovno grananje tj. izvršavanje jedne grupe komandi ako je neki logički uslov ispunjen, a druge grupe komandi ako nije. Ovo je **“jednostruko” grananje** jer su ponuđene maksimalno dve alternativne putanje (“grane”) od kojih se bira i izvršava samo jedna. Ako je potrebno izvršiti proveru više uslova, moguće je napisati više IF naredbi od kojih svaka proverava po jedan uslov. Međutim, isti efekat se može postići korišćenjem SWITCH naredbe.

SWITCH naredba omogućava proveravanje više uslova odjednom i predstavlja način za izražavanje **“višestrukog grananja”** u Javi. Višestruko grananje je takvo grananje u kojem je ponuđeno više alternativnih putanja tj. grana od kojih se, u zavisnosti od ispunjenosti uslova, može izvršiti jedna ali i više. Deklaracija SWITCH naredbe se vrši na sledeći način:

```
switch ( ...selektor... ){  
  
    case vrednost_1:      komanda_1;  
                          break;  
  
    case vrednost_2:      komanda_2;  
                          break;  
  
    ...  
  
    default:              komanda_d;  
}
```

Deklaracija SWITCH naredbe počinje rezervisanom reči “switch” posle koje sledi **selektor** koji se piše u zagradi. Selektor može biti **bilo koja promenljiva koja je celobrojnog ili char tipa**. Promenljive koje su bilo kog drugog tipa (String, boolean...) ne mogu biti selektori. Selektor može da bude i **neki izraz čiji rezultat je celobrojnog ili char tipa** npr. “A%2”. SWITCH naredba ima i telo u kojem su definisane sve moguće grane i komande koje bi trebalo izvršiti u okviru grane. Grananje se definiše ključnom reči “case” iza koje sledi neka vrednost. Pri pokretanju SWITCH naredbe, **poredi se trenutna vrednost selektora sa vrednostima koje su napisane iza reči “case”**. Kada se nađe grananje koje ima vrednost koja je jednaka vrednosti selektora, izvršavaju se komande napisane u produžetku (npr. komanda\_1). **Ako se ne nađe nijedno odgovarajuće grananje, izvršava se ona grana koja je označena sa “default”(komanda\_d)**. SWITCH naredba **ne mora da ima “default” granu**.

Ovako kako je napisana, ova SWITCH naredba predstavlja višestruko grananje pri kojem se bira i izvršava samo jedna grana od više ponuđenih. Upravo to je smisao **“break” naredbe koja je napisana kao poslednja komanda svakog grananja**. Kada se izvrše sve komande nekog grananja, nailazi se na “break” komandu čime se prekida SWITCH naredba. Da je “break” naredba izostavljena, **izvršila bi se odabrana grana, ali i sve ostale koje su napisane “ispod” nje (uključujući i “default” granu)**. Tako napisana SWITCH naredba bi predstavljala višestruko grananje pri kojem se bira i izvršava više grana.

Takođe, potrebno je primetiti da komande koje su napisane u okviru jedne grane nisu uokvirene vitičastim zagradama pa je i sledeći kod potpuno sintaksno ispravan:

```
switch ( ...selektor... ){
```

```

    case vrednost_1:      komanda_1_1;
                          komanda_1_2;
                          komanda_1_3;
                          break;

    case vrednost_2:      komanda_2_1;
                          komanda_2_2;
                          komanda_2_3;
                          break;

    ...

    default:              komanda_d_1;
                          komanda_d_2;
                          komanda_d_3;

}

```

### Primer 1

Napraviti klasu **DesifrovanjeDNK** koja ima:

- Statičku metodu **desifrujDNK** koja kao ulazni parametar dobija neki znak koji predstavlja početno slovo neke od nukleotida od kojih se sastoje karike DNK lanca. Ako znak ima vrednost 'A','C','G' ili 'T', potrebno je na ekranu ispisati naziv odgovarajuće nukleotide koja počinje tim slovom: Adenin, Citozin, Guanin i Timin. Ako uneti znak nema nijednu od tih vrednosti ispisati na ekranu poruku o grešci.

Napraviti klasu **TestDesifrovanjeDNK** koja poziva metodu klase **DesifrovanjeDNK**.

```

class DesifrovanjeDNK {

    static void desifrujDNK(char karika) {

        switch (karika) {

            case 'A':
                System.out.println("Adenin");
                break;
            case 'C':
                System.out.println("Citozin");
                break;
            case 'G':
                System.out.println("Guanin");
                break;
            case 'T':
                System.out.println("Timin");
                break;
            default:
                System.out.println("Greska!");
        }

    }

}

```

```

}

class TestDesifrovanjeDNK {

    public static void main(String[] args) {

        DesifrovanjeDNK.desifrujDNK('C');

    }

}

```

SWITCH naredba, nažalost, ne omogućava formiranje logičkih uslova kao kod IF naredbe, pa je zato njena upotrebljivost ograničena. Ako je potrebno proveriti neke složenije uslove, ili ako se ne može naći odgovarajući selektor, jedino rešenje koje ostaje je da se napiše više IF naredbi.

## Zadaci

### Zadatak 1

Napisati klasu **Sifrarnik** koja ima:

- Statičku metodu koja kao ulazni parametar dobija ocenu učenika i na ekranu ispisuje da li je u pitanju ocena: odličan, vrlo dobar, dobar, dovoljan ili nedovoljan. Ako je unet broj koji je veći od 5 ili manji od 1, ispisati poruku o grešci.
- Statičku metodu koja kao ulazni parametar dobija ocenu učenika i vraća String u kome piše da li je u pitanju ocena: odličan, vrlo dobar, dobar, dovoljan ili nedovoljan. Ako je unet broj koji je veći od 5 ili manji od 1, ispisati poruku o grešci.
- Statičku metodu koja kao ulazni parametar dobija neki znak i proverava da li je taj znak cifra ili neki znak interpunkcije (! , . ? ; :). Ako je u pitanju cifra ili znak interpunkcije, potrebno je ispisati poruku o tipu znaka na ekranu ("Cifra" ili "Znak interpunkcije"). Ako nije u pitanju ni cifra ni znak interpunkcije, metoda ne bi trebalo ništa da uradi.

Napisati klasu **TestSifrarnik** koja poziva sve metode klase Sifrarnik.

### Rešenje:

```

class Sifrarnik {

    static void ispisiOcenu(int ocena) {

        switch (ocena) {

            case 5:
                System.out.println("Odlican");
                break;
            case 4:
                System.out.println("Vrlo dobar");
                break;
            case 3:
                System.out.println("Dobar");
                break;
            case 2:
                System.out.println("Dovoljan");
                break;
            case 1:
                System.out.println("Nedovoljan");
                break;
            default:
                System.out.println("Greska");
        }
    }
}

```

```

    }

}

static String vratiOcenu(int ocena) {

    switch (ocena) {
        // Nisu potrebne "break" naredbe jer
        // "return" prekida celu metodu.
        case 5:
            return "Odlican";
        case 4:
            return "Vrlo dobar";
        case 3:
            return "Dobar";
        case 2:
            return "Dovoljan";
        case 1:
            return "Nedovoljan";
        default:
            return "Greska";
    }

}

static void cifraIliZnakInterpunkcije(char c) {

    switch (c) {

        //Nema potrebe da se pise ista komanda za
        //ispis u okviru svakog grananja vec
        //je dovoljno izostaviti "break" naredbu
        //i napisati komandu za ispis kod poslednjeg
        //grananja.
        case '!':
        case '.':
        case ',':
        case '?':
        case ';':
        case ':':
            System.out.println("Znak interpunkcije");
            break;
        case '0':
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
            System.out.println("Cifra");
    }

}

}

class TestSifrarnik {

    public static void main(String[] args) {

```

```
Sifrarnik.ispisiOcenu(5);

Sifrarnik.ispisiOcenu(-1);

String ocena = Sifrarnik.vratiOcenu(5);
System.out.println("Ocena je: "+ocena);

Sifrarnik.cifraIliZnakInterpunkcije('.');

}

}
```