

Komande za kontrolu toka izvršavanja programa

Java komande se izvršavaju u redosledu u kojem su napisane - “odozgo na dole”. Međutim, u nekim situacijama je poželjno izvršiti neke komande samo ako je neki određeni uslov ispunjen. Druge situacije zahtevaju da se određena komanda ponovi veći broj puta. U ovim situacijama je potrebno kontrolisati tok izvršavanja programa. Java naredbe koje to omogućavaju se dele u dve grupe:

- Naredbe za uslovno grananje (IF i SWITCH naredbe)
- Naredbe za ciklično ponavljanje (FOR, WHILE i DO-WHILE naredbe).

IF naredba

Programeri se često susreću sa situacijama u kojima je potrebno proveriti neki uslov, pa u zavisnosti od njegove ispunjenosti, uraditi jednu stvar ili drugu. Na primer, potrebno je proveriti da li je uneta visina u okviru granica 80 - 240 cm, pa ako je visina van granica ispisati poruku o grešci. Ovo se može postići korišćenjem IF naredbe. Deklaracija IF naredbe se vrši na sledeći način:

```
if ( ...uslov... ) komanda_1;
```

Deklaracija počinje rezervisanom reči “if” (mala slova). Posle toga, piše se logički uslov koji se proverava. Posle logičkog uslova se navodi **komanda koja se izvršava ako je uslov ispunjen (komanda_1)**.

Uslov koji se piše između zagrada u okviru IF naredbe predstavlja neki logički izraz koji se može proveriti i svesti na tačan (true) ili netačan (false). U najjednostavnijem slučaju, taj izraz može da predstavlja proveru da li je neki broj veći, manji ili jednak nekom drugom broju. Operatori za poređenje vrednosti su dati u sledećoj tabeli.

<i>Operatori za poređenje vrednosti</i>		
<i>Operator</i>	<i>Opis</i>	<i>Primeri</i>
>	Veće od	a > b x > 2
<	Manje od	a < b x < 2
>=	Veće ili jednako	a >= b x >= 135.5
<=	Manje ili jednako	a <= b x <= 13
==	Jednako (piše se kao dva znaka jednakosti jedan pored drugog)	a == b x == 45.22
!=	Različito	a != b x != 1

Operatori za poređenje vrednosti se **moгу koristiti isključivo sa prostim tipovima podataka**, jer

ne funkcionišu u onim situacijama kada je potrebno uporediti dva objekta. String, Calendar i svi drugi objekti se porede na drugi način, a ne preko operatora za poređenje vrednosti. Ako bi se pokušalo poređenje dva objekta preko operatora "=", **poredile bi se adrese objekata u memoriji a ne sadržaj njihovih atributa.**

Potrebno je napomenuti i **da operatori "=" i "==" nisu isti**. U prvom slučaju, u pitanju je operator dodeljivanja vrednosti (levoj strani se dodeljuje vrednost sa desne strane), dok je u drugom slučaju u pitanju operator poređenja jednakosti (poređi se da li su leva i desna strana jednake). Ako se umesto jednog napiše drugi, Java javlja grešku. Novi programeri često greše i mešaju ova dva operatora, pa je potrebno obratiti pažnju pri pisanju logičkih uslova.

Primer 1

Napraviti klasu **ProveraCelihBrojeva** koja ima:

- Metodu **proveriZnak** koja proverava da li je broj, koji se prosleđuje kao ulazni parametar pozitivan, negativan ili nula i ispisuje poruku o tome na ekranu u formi "Broj ## je ##".
- Metodu **proveriVeceManjeJednako** koja prima dva broja A i B kao ulazne parametre i proverava da li je $A > B$, $A < B$ ili $A = B$ i ispisuje odgovarajucu poruku o tome na ekranu.

```
class ProveraCelihBrojeva {

    void proveriZnak (int a){

        if (a == 0) System.out.println("Broj "+a+" je jednak nuli");

        if (a > 0) System.out.println("Broj "+a+" je veci od nule");

        if (a < 0) System.out.println("Broj "+a+" je manji od nule");

    }

    void proveriVeceManjeJednako (int a, int b){

        if (a > b)
            System.out.println("Broj "+a+" je veci od broja "+b);

        if (a == b)
            System.out.println("Broj "+a+" je jednak broju "+b);

        if (a < b)
            System.out.println("Broj "+a+" je manji od broja "+b);

    }

}
```

Metoda ProveraCelihBrojeva kao ulazni parametar dobija broj, proverava da li je jednak nuli, manji od nule ili veći od nule i ispisuje poruku o tome na ekranu. Provera se vrši uz pomoć tri nezavisne IF naredbe. Uslov koji se proverava u okviru prve naredbe je da li je broj jednak nuli. Ako jeste, izvršiće se komanda koja se nalazi u produžetku tj. ispisaće se na ekranu poruka da je broj jednak nuli. Ako uslov nije ispunjen, tj. ako broj nije jednak nuli, neće se izvršiti ispis na ekranu. Slično važi i za naredne dve IF naredbe, samo što one proveravaju da li je broj veći od nule

ili manji od nule.

*Metoda **proveriVeceManjeJednako** isto ima tri IF naredbe, ali se u okviru njih proverava da li je ulazni parametar A veći, jednak ili manji od ulaznog parametra B.*

U prethodnom primeru je prikazan i princip da se Java komande mogu pisati u više redova. Drugim rečima, IF komanda se može napisati i ovako:

```
if ( ...uslov... )  
    komanda_1;
```

Ovo je korisno u situacijama kada je uslov ili komanda koja ga sledi prilično dugačka pa ne može da stane u jedan red na ekranu, ili ne može da se vidi u celosti bez pomeranja teksta.

IF naredba može da ima i “else” deo (ali ne mora). Posle rezervisane reči “else” se piše **komanda koja se izvršava samo ako uslov nije ispunjen (komanda_2)**. Deklaracija IF naredbe koja ima i “else” glasi:

```
if ( ...uslov... )    komanda_1;  
    else              komanda_2;
```

Primer 2

Dopuniti klasu **ProveraCelihBrojeva** tako da ima i:

- Metodu **razlicito** koja prima dva broja kao ulazne parametre i vraća TRUE ako su brojevi različiti a inače FALSE.
- Metodu **proveriParnost** koja prima ceo broj kao ulazni parametar i proverava da li je paran ili neparan. Metoda vraća TRUE ako je paran, a FALSE ako je neparan.

Napraviti klasu **TestProveraCelihBrojeva** koja kreira jedan objekat klase **ProveraCelihBrojeva** i koristi njene metode.

Kod ove dve metode je sledeći:

```
boolean razlicito (int a, int b){  
    if (a != b) return true;  
    else return false;  
}  
  
boolean proveriparnost (int a){  
    if ((a%2) == 0) return true;  
    else return false;  
}
```

Kod klase **TestProveraCelihBrojeva** je sledeći:

```
class TestProveraCelihBrojeva {  
  
    public static void main(String[] args) {  
  
        ProveraCelihBrojeva p = new ProveraCelihBrojeva();
```

```

p.proveriZnak(-12);

p.proveriVeceManjeJednako(123, 456);

boolean razliciti = p.razlicito(12, 13);
if (razliciti)
    System.out.println("Brojevi su razliciti");
else
    System.out.println("Brojevi su jednaki");

boolean paran = p.proveriParnost(33);
if (paran)
    System.out.println("Broj 33 je paran");
else
    System.out.println("Broj 33 je neparan");
}
}

```

U metodi proveraParnosti je prikazano kako se proverava da li je broj paran ili neparan. Ako je ostatak pri deljenju brojem dva jednak nuli, broj je paran, a inače je neparan.

Dosadašnji primeri obuhvataju one slučajeve u kojima je posle provere uslova IF naredbe potrebno izvršiti samo jednu komandu. U slučaju kada je potrebno izvršiti više naredbi posle provere uslova, naredbe se moraju uokviriti u **blok naredbi** vitičastim zagradama:

```

if ( ...uslov... ) {
    komanda_1;
    komanda_2;
    ...
    komanda_n;
}
else {
    komanda_1_1;
    komanda_1_2;
    ...
    komanda_1_m;
}

```

Primer 3

Napraviti klasu **ProveraRealnihBrojeva** tako da ima:

- Metodu **manjiOdPi** koja prima realan broj kao ulazni parametar i vraća TRUE ako je broj manji od 3.141592 a inače FALSE. U oba slučaja, trebalo bi ispisati poruku na ekranu o tome da li je broj manji od pi ili nije.

```

class ProveraRealnihBrojeva {

    boolean manjiOdPi(double x) {
        if (x < 3.141592) {
            System.out.println("Broj "+x+" je manji od pi");
            return true;
        }
    }
}

```

```

    }
    else {
        System.out.println("Broj "+x+" je veci ili jednak pi");
        return false;
    }
}
}

```

Metoda *manjiOdPi* sadrži IF naredbu koja pokreće dve naredbe ako je uslov ispunjen i druge dve ako nije. Ove naredbe su morale da budu uokvirene u blok naredbi korišćenjem vitičastih zagrada.

Prethodni primer ukazuje i na još jednu pojavu. Pored toga što se return naredbom vraća neka vrednost, vrši se i momentalno prekidanje daljeg izvršavanja metode. Drugim rečima, **svaki put kada se return komanda izvrši, metoda se prekida i sve komande koje su napisane posle (“ispod”) ove komande se ne izvršavaju.** Zbog toga, trebalo bi obratiti pažnju na to da je return komanda poslednja u bloku naredbi.

Uslov koji se proverava u okviru IF komande može da bude i složen tj. da se sastoji iz više prostijih uslova. Formiranje složenijih uslova se vrši korišćenjem **logičkih operatora** I (AND), ILI (OR) i NE (NOT) čija specifikacija je data u sledećoj tabeli.

Operatori za poređenje vrednosti		
Operator	Opis	Primeri
&&	I (AND)	((a > 2) && (a < 5)) a mora da bude veće od 2 i manje od 5 da bi ceo uslov bio tačan (true)
	ILI (OR)	((x < 0) (x > 33.3)) x mora da bude manje od 0 ili veće od 33.3 da bi ceo uslov bio tačan
!	NE (NOT)	(! (a > b)) a ne sme da bude veće od b da bi ceo uslov bio tačan

Primer 4

Dopuniti klasu **ProveraRealnihBrojeva** tako da ima i:

- Metodu koja proverava da li je ulazni parametar *A* (realan broj) u rasponu od 100 do 200 uključujući i te vrednosti. Ako jeste, metoda vraća TRUE a inače FALSE.
- Metodu koja proverava da li je ulazni parametar *A* (realan broj) manji od nule ili veći od 33.3. Ako važi bilo koji od ova dva uslova, metoda vraća TRUE a inače FALSE. U oba slučaja je potrebno ispisati i odgovarajuće obaveštenje na ekranu.

Kod traženih metoda je:

```

boolean proveraRasponal (double a) {
    if ((a >=100) && (a <=200)) return true;
    else return false;
}

```

```

}

boolean proveraRaspona2 (double a){
    if ((a < 0) || (a > 33.3)) {
        System.out.println("Broj "+a+
            " manji od nule ili veci od 33.3");
        return true;
    }
    else {
        System.out.println("Broj "+a+
            " je u rasponu 0 - 33.3");
        return false;
    }
}

```

U dosadašnjim primerima su prikazane neke metode koje sadrže više IF naredbi koje su međusobno nezavisne - napisane su kao odvojene naredbe. Međutim, u nekim situacijama se traži da se posle provere jednog uslova provere i još neki uslovi ali samo ako je prvi uslov zadovoljen. Rešenje ovakvih problema podrazumeva pisanje tzv. **ugnježdenih naredbi**, konkretno - ugnježdenih IF naredbi:

```

if ( ...uslov 1... )      if ( ...uslov 2... )      komanda_1_1;
                           else                    komanda_2_1;

                           if ( ...uslov 3... )      komanda_1_2;
                           else                    komanda_2_2;
else

```

Iz deklaracije se može videti da se umesto običnih komandi koje slede uslov i “else” deo IF naredbe, koriste nove IF naredbe. Tako će da se proveri “uslov 2” tek ako je zadovoljen “uslov 1”. Ako i “uslov 1” i “uslov 2” važe, izvršiće se “komanda_1_1”, a ako važi “uslov 1” a ne važi “uslov 2” izvršiće se “komanda_2_1”. Slično tome, ako ne važi “uslov 1”, proveriće se “uslov 3” pa, ako on bude važio, izvršiće se “komanda_1_2”, a u suprotnom “komanda_2_2”. IF naredbe se mogu ugnježđavati i stavljati u blok naredbi zajedno sa drugim naredbama tako da je i sledeća situacija česta:

```

if ( ...uslov... ) {
    komanda_1;
    komanda_2;
    if ( ...uslov 2... )      komanda_1_1;
                           else                    komanda_2_1;
    ...
    komanda_n;
}
else {
    if ( ...uslov 3... )      komanda_1_2;
                           else                    komanda_2_2;
    komanda_3_1;
    komanda_3_2;
    ...
    komanda_3_m;
}

```

Primer 5

Napraviti klasu **VisinskeIStarosneGrupe** tako da ima:

- Metodu *proveriVisinu* koja, kao ulazni parametar prima visinu neke osobe u santimetrima (realan broj) i ispisuje na ekranu da li ta osoba pripada niskim osobama (manje od ili jednako 158cm), srednje visokim osobama (više od 158 cm, manje ili jednako 179cm) ili visokim osobama (više od 179 cm). Ako je uneta visina van granica (120-240cm), na ekranu je potrebno ispisati samo poruku o grešci.
- Metodu *proveriStarosnoDoba* koja kao ulazni parametar dobija starost osobe izraženu u godinama. Metoda prvo proverava da li je uneta starost u granicama 0 - 120 godina. Ako nije, ispisuje se poruka o grešci na ekranu. Ako jeste, potrebno je proveriti da li je osoba mlada (0-30 godina), srednjeg doba (31-55 godina) ili stara (56 godina i više) i ispisati poruku o tome na ekranu.

Napraviti klasu **TestVisinskeIStarosneGrupe** koja kreira jedan objekat klase **VisinskeIStarosneGrupe** i poziva njegove metode.

```
class VisinskeIStarosneGrupe {  
  
    void proveriVisinu(double visina){  
        if ((visina < 120) || (visina > 240))  
            System.out.println("Visina je van granica");  
        else{  
            if (visina <= 158)  
                System.out.println("Osoba je niska");  
            if ( (visina > 158) && (visina <= 179))  
                System.out.println("Osoba je srednje visine");  
            if (visina > 179)  
                System.out.println("Osoba je visoka");  
        }  
    }  
  
    void proveriStarosnoDoba (int starost){  
        if ((starost < 0) || (starost > 120))  
            System.out.println("Uneta starost je van granica");  
        else{  
            if (starost <= 30)  
                System.out.println("Osoba je mlada");  
            if ( (starost > 30) && (starost <= 55))  
                System.out.println("Osoba je srednjeg doba");  
            if (starost > 55)  
                System.out.println("Osoba je stara");  
        }  
    }  
}  
  
class TestVisinskeIStarosneGrupe {  
  
    public static void main(String[] args) {  
  
        VisinskeIStarosneGrupe vsg = new VisinskeIStarosneGrupe();  
  
        vsg.proveriStarosnoDoba(5);  
        vsg.proveriStarosnoDoba(35);  
        vsg.proveriStarosnoDoba(56);  
    }  
}
```

```

        vsg.proveriVisinu(145);
        vsg.proveriVisinu(185);
        vsg.proveriVisinu(175);
    }
}

```

IF naredba se često koristi kada je potrebno proveriti da li je neka ulazna vrednost u okviru zadatih granica za neki atribut ili promenljivu (npr. da li je visina u okviru granica 120-240cm). Ovaj postupak provere ulaznih vrednosti se često zove **logička kontrola ulaznih podataka**.

Zadaci

Zadatak 1

Napraviti klasu **Televizor2**. Ova klasa bi trebalo da ima:

- Atribut **jacinaTona** koji je ceo broj i označava trenutnu jačinu tona na televizoru. Početna vrednost ovog atributa je 0 (ton je utišan do kraja).
- Atribut **trenutniProgram** koji označava broj programa koji je trenutno na televizoru (npr. uključen je program 5). Početna vrednost ovog atributa je 1.
- Atribut **uključen** koji označava da li je televizor uključen ili nije (ako je uključen ima vrednost TRUE, a inače FALSE). Smatra se da je na početku televizor isključen.
- Metodu **uključi** koja uključuje televizor ako je isključen (postavlja vrednost atributa uključen na TRUE). Ako je televizor već uključen, onda ispisuje poruku "Televizor je vec uključen".
- Metodu **isključi** koja isključuje televizor ako je isključen (postavlja vrednost atributa uključen na FALSE). Ako je televizor već isključen, ispisuje poruku "Televizor je vec isključen".
- Metodu **pojacajTon** koja povećava vrednost atributa jacinaTona za jedan pod uslovom da maksimalna vrednost tona nije postignuta (ona iznosi 10). Ako jeste, ispisuje se poruka "Ton je vec na maksimumu".
- Metodu **smanjiTon** koja smanjuje vrednost atributa jacinaTona za jedan pod uslovom da ton nije već na nuli. Ako jeste, potrebno je ispisati poruku "Ton je vec na minimumu".
- Metodu **promeniProgramNavise** koja povećava vrednost atributa trenutniProgram za jedan pod uslovom da trenutno nije uključen 99-ti program. Ako jeste, onda se program podesava na prvi (televizor ima ukupno 99 programa pa se sa 99-tog programa direktno prelazi na 1.).
- Metodu **promeniProgramNanize** koja smanjuje vrednost atributa trenutniProgram za jedan pod uslovom da trenutno nije uključen 1. program. Ako jeste, onda se program podesava na 99-ti.
- Metodu **ispisiParametre** koja, ako je televizor uključen, ispisuje podatke o trenutnom programu i jacini tona, a ako je televizor isključen, samo ispisuje poruku "Televizor je isključen".

Napraviti klasu **TestTelevizor2** koja kreira jedan objekat klase Televizor2 i poziva neke od njegovih metoda. Posle svakog poziva metode, pozvati metodu ispisiParametre i uočiti promene u vrednostima atributa.

Rešenje:

```

class Televizor2 {

    int jacinaTona = 0;

    int trenutniProgram = 1;

    boolean ukljucen = false;

    void ukljuci() {
        if (ukljucen == false)
            ukljucen = true;
        else
            System.out.println("Televizor je vec ukljucen");
    }
}

```



```

void iskljuci() {
    if (ukljucen == true)
        ukljucen = false;
    else
        System.out.println("Televizor je vec iskljucen");
}

void pojacajTon() {
    if (jacinaTona < 10)
        jacinaTona = jacinaTona + 1;
    else
        System.out.println("Ton je vec na maksimumu");
}

void smanjiTon() {
    if (jacinaTona > 0)
        jacinaTona = jacinaTona - 1;
    else
        System.out.println("Ton je vec na minimumu");
}

void promeniProgramNavise() {
    if (trenutniProgram < 99)
        trenutniProgram = trenutniProgram + 1;
    else
        trenutniProgram = 1;
}

void promeniProgramNanize() {
    if (trenutniProgram > 1)
        trenutniProgram = trenutniProgram - 1;
    else
        trenutniProgram = 99;
}

void ispisiParametre() {
    if (ukljucen) {
        System.out.println("Jacina tona je " + jacinaTona);
        System.out.println("Trenutni program je " + trenutniProgram);
        System.out.println("Televizor je ukljucen");
    } else
        System.out.println("Televizor je iskljucen");
}

}

class TestTelevizor2 {

    public static void main(String[] args){

        Televizor2 t = new Televizor2();

        t.ispisiParametre();

        t.ukljuci();

        t.ispisiParametre();

        t.pojacajTon();
    }
}

```

```

        t.ispisiParametre();

        t.promeniProgramNanize();

        t.ispisiParametre();
    }

}

```

Zadatak 2

Napraviti klasu **Radio2**. Ova klasa bi trebalo da ima:

- Atribut **fmFrekvencija** koji označava trenutnu FM radio frekvenciju koju radio pušta (npr. 102.2 ili 87.5). Početna vrednost ovog polja je 87.5.
- Atribut **amFrekvencija** koji označava trenutnu AM radio frekvenciju koju radio pušta (npr. 567 ili 1500). Početna vrednost ovog parametra je 567.
- Atribut **band** koji ima vrednost 'A' (slovo A) ako radio pušta AM frekvenciju ili 'F' (slovo F) ako radio pušta FM frekvenciju. Početna vrednost ovog parametra je 'F'.
- Metodu **podesiFMFrekvenciju** koja prima kao ulazni parametar novu FM frekvenciju i podešava vrednost atributa fmFrekvencija na nju pod uslovom da je frekvencija u opsegu između 87.5 i 108.0. Ako je frekvencija manja od 87.5 ili veća od 108.0 ispisuje se poruka o grešci.
- Metodu **podesiAMFrekvenciju** koja prima kao ulazni parametar novu AM frekvenciju i podešava vrednost atributa amFrekvencija na nju pod uslovom da je da je frekvencija u opsegu između 567 i 1576. Ako je frekvencija manja od 567 ili veća od 1576 ispisuje se poruka o grešci.
- Metodu **promeniBand** koja kao ulazni parametar prima slovo i menja vrednost atributa band na to slovo. Potrebno je da ulazni parametar ima vrednost 'A' ili 'F'. Ako nema, ne vrši se promena vrednosti atributa band već se ispisuje poruka o grešci.
- Metodu **ispisiParametre** koja, ako radio pušta AM frekvenciju, ispisuje poruku "Radio pušta AM frekvenciju." i trenutnu AM frekvenciju a ako radio pušta FM frekvenciju ispisuje "Radio pušta FM frekvenciju." i trenutnu FM frekvenciju.

Napraviti klasu **TestRadio2** koja kreira dva objekta klase Radio2. Podesiti prvi radio da radi na AM frekvenciji 570. Podesiti drugi radio da radi na FM frekvenciji 107.9. Ispisati parametre oba radija.

Rešenje:

```

class Radio2 {

    double fmFrekvencija = 87.5;
    int amFrekvencija = 567;
    char band = 'F';

    void podesiFMFrekvenciju (double fmFrekvencija) {
        if ((fmFrekvencija >= 87.5) && (fmFrekvencija <= 108.0))
            this.fmFrekvencija = fmFrekvencija;
        else System.out.println(
            "Greska! FM frekvencija mora da bude u opsegu 87.5-108 MHz");
    }

    void podesiAMFrekvenciju (int amFrekvencija) {
        if ((amFrekvencija >= 567) && (amFrekvencija <= 1576))
            this.amFrekvencija = amFrekvencija;
        else System.out.println(
            "Greska! AM frekvencija mora da bude u opsegu 567-1576 KHz");
    }

    void promeniBand (char b) {
        if ((b == 'A') || (b == 'F')) band = b;
        else System.out.println("Greska! Band moze samo da ima vrednost A ili F");
    }
}

```

```

    }

    void ispisiParametre () {
        if (band == 'A') {
            System.out.println ("Radio pusta AM frekvenciju");
            System.out.println ("Trenutna AM frekvencija je "+
                                amFrekvencija);
        }
        else {
            System.out.println ("Radio pusta FM frekvenciju");
            System.out.println ("Trenutna FM frekvencija je "+
                                fmFrekvencija);
        }
    }
}

class TestRadio2 {

    public static void main(String[] args) {

        Radio2 r1 = new Radio2();
        Radio2 r2 = new Radio2();

        r1.promeniBand('A');
        r1.podesiAMFrekvenciju(570);

        r2.promeniBand('F');
        r2.podesiFMFrekvenciju(107.9);

        r1.ispisiParametre();
        r2.ispisiParametre();
    }
}

```

Zadatak 3

Napraviti klasu **BMIKalkulator** koja ima sledeće metode:

- Metodu **izracunajBMI** koja izračunava i vraća BMI tj. body mass index na osnovu zadate visine u metrima (realan broj) i težine u kilogramima (realan broj) i to po formuli $BMI = \text{težina} / (\text{visina})^2$ (težina kroz visina na kvarat). Pre izračunavanja je potrebno proveriti da li su visina i težina u okviru granica (1.20-2.40m i 30-200kg). Ako nisu, metoda ispisuje na ekranu poruku o grešci i vraća nulu.
- Metodu **ispisiBMIKategoriju** koja dobija kao ulazni argument neku vrednost BMI (npr. 20,02) i ispisuje na ekranu kategoriju u koju spada nosilac BMI u formi “Vaš BMI je: ##, vi pripadate grupi: ###”.

Primer računanja BMI: Visina = 1.87 m,
 Težina = 100 kg,
 $BMI = 100 / (1.87)^2 = 28,59$

Kategorija	BMI
Anoreksija	$BMI \leq 15$
Mršavost	$15 < BMI \leq 18.5$
Normalno	$18.5 < BMI \leq 25$
Popunjen	$25 < BMI \leq 30$
Debeo	$30 < BMI \leq 40$
Gojazan	$BMI > 40$

Napraviti klasu **ProveraBMIKalkulatora** koja proverava rad klase BMIKalkulator.

Rešenje:

```
class BMIKalkulator {  
  
    double izracunajBMI (double visina, double tezina){  
  
        if ((visina < 1.2) || (visina > 2.4)) {  
            System.out.println (  
                "Greska: visina mora da bude izmedju 1.2 i 2.4 metra");  
                return 0;  
            }  
        if ((tezina < 30) || (tezina > 200)) {  
            System.out.println ("Greska: tezina mora da bude izmedju 30 i 200 kg");  
                return 0;  
            }  
  
        double rezultat = (tezina / (visina*visina));  
        return rezultat;  
  
    }  
  
    void ispisiBMIKategoriju (double bmi){  
  
        if (bmi <= 15)  
            System.out.println("Vas BMI je: "+bmi+" vi pripadate grupi: anoreksicni");  
  
        if ((bmi > 15) && (bmi <= 18.5))  
            System.out.println("Vas BMI je: "+bmi+" vi pripadate grupi: mrsavi");  
  
        if ((bmi > 18.5) && (bmi <= 25))  
            System.out.println("Vas BMI je: "+bmi+" vi pripadate grupi: normalni");  
  
        if ((bmi > 25) && (bmi <= 30))  
            System.out.println("Vas BMI je: "+bmi+" vi pripadate grupi: popunjeni");  
  
        if ((bmi > 30) && (bmi <= 40))  
            System.out.println("Vas BMI je: "+bmi+" vi pripadate grupi: debeli");  
  
        if (bmi > 40)  
            System.out.println("Vas BMI je: "+bmi+" vi pripadate grupi: gojazni");  
    }  
  
}
```

Zadatak 4

Napraviti klasu **KlimaUredjaj2**. Ova klasa bi trebalo da ima:

- Atribut **trenutnaTemperatura** koji označava temperaturu u Celzijusima na koju je klima podešena (ceo broj – npr. 23). Početna vrednost ovog atributa je 18 stepeni.
- Atribut **rezimRada** koji označava da li klima greje, hladi ili radi automatski. Ako greje, ovaj atribut ima vrednost 'G', ako hladi 'H' a ako radi automatski onda 'A'. U početku, klima je podešena na automatski režim rada.
- Metodu **smanjiTemperaturu** koja smanjuje trenutnu temperaturu za jedan stepen samo ako je trenutna temperatura viša od 18C. Ako nije, ispisati poruku o grešci.
- Metodu **povecajTemperaturu** koja povećava trenutnu temperaturu za jedan stepen samo ako je trenutna temperatura manja od 30C. Ako nije, ispisati poruku o grešci.
- Metodu **podesiTemperaturu** koja kao ulazni parametar prima konkretnu vrednost temperature i postavlja vrednost atributa trenutnaTemperatura na nju samo pod uslovom da je ova vrednost u opsegu 18-30C. U suprotnom ispisati poruku o grešci.
- Metodu **promeniRezim** koja podešava vrednost atributa rezimRada na ulaznu vrednost pod uslovom da ulazna

- vrednost bude 'A', 'G' ili 'H'. Ako ulazna vrednost nije ni jedno od ova tri, ispisati poruku o grešci.
- Metodu **ispisiParametre** koja ispisuje trenutna podešavanja klima uređaja u formatu “Rezim rada: ##”, “Trenutna temperatura je podesena na ## C”.

Napraviti klasu **ProveraKlimaUredjaja2** koja koristi dva objekta klase **KlimaUredjaj**. Potrebno je prvu klimu podesiti da greje na 26 C a drugu da hladi na 18 C. Ispisati parametre obe klime na ekranu.

Rešenje:

```
class KlimaUredjaj2 {

    int trenutnaTemperatura = 18;
    char rezimRada = 'A';

    void smanjiTemperaturu () {
        if (trenutnaTemperatura > 18) trenutnaTemperatura--;
        else
            System.out.println("Greska: temperatura se ne moze smanjiti ispod 18C");
    }

    void povecajTemperaturu () {
        if (trenutnaTemperatura < 30) trenutnaTemperatura++;
        else
            System.out.println("Greska: temperatura se ne moze povecati preko 30C");
    }

    void podesiTemperaturu (int t) {
        if ((t>=18) && (t<=30)) trenutnaTemperatura = t;
        else
            System.out.println("Greska: temperatura mora da bude u opsegu 18-30C");
    }

    void promeniRezim (char r) {
        if ((r=='A') || (r=='H') || (r=='G')) rezimRada = r;
        else
            System.out.println("Greska: rezim rada mora da bude A, G ili H");
    }

    void ispisiParametre() {

        if (rezimRada == 'A')
            System.out.println ("Rezim rada je automatski");
        if (rezimRada == 'G')
            System.out.println ("Rezim rada je grejanje");
        if (rezimRada == 'H')
            System.out.println ("Rezim rada je hladjenje");

        System.out.println ("Klima je podesena na "+trenutnaTemperatura+" C");
    }
}

class ProveraKlimaUredjaja2 {

    public static void main (String[] args){

        KlimaUredjaj2 k1 = new KlimaUredjaj2();
        KlimaUredjaj2 k2 = new KlimaUredjaj2();

        k1.promeniRezim('G');
```

```

        k1.podesiTemperaturu(26);

        k2.promeniRezim('H');
        k2.podesiTemperaturu(18);

        k1.ispisiParametre();
        k2.ispisiParametre();
    }
}

```

Zadatak 5

Napraviti klasu **Automobil**. Ova klasa bi trebalo da ima:

- Atribut model.
- Atribut marka
- Atribut kubikaža (ceo broj) koji predstavlja kubikažu motora izraženu u kubnim santimetrima (ccm).
- Atribut snaga (ceo broj) koji predstavlja broj konjskih snaga motora (KS).

Napraviti klasu **TroskoviRegistracije** koja ima

- Statičku metodu `obracunajDoprinos` koja kao ulazni parametar dobija objekat klase **Automobil** i vraća iznos doprinosa koji je potrebno platiti pri registraciji. Tarife zavise od kubikaže automobila i date su u sledećoj tabeli. Pre izračunavanja, potrebno je proveriti da li je kubikaža u okviru granica (400-10000 kubika), pa ako nije, ispisati poruku o grešci na ekranu i vratiti vrednost 0.

Kubikaža (ccm)	Cena (din)
$k < 1000$	145.5
$1000 \leq k < 1400$	200.0
$1400 \leq k < 2500$	250.0
$2500 \leq k$	413.3

- Statičku metodu `obracunajOsiguranje` koja kao ulazni parametar dobija objekat klase **Automobil** i vraća iznos koji je potrebno platiti kao obavezno osiguranje pri registraciji. Tarife osiguranja zavise od snage motora i date su u sledećoj tabeli. Pre izračunavanja, potrebno je proveriti da li je snaga u okviru granica (20-1000 KS), pa ako nije, ispisati poruku o grešci na ekranu i vratiti vrednost 0.

Snaga (KS)	Cena (din)
$s < 55$	3000.0
$55 \leq s < 75$	4500.0
$75 \leq s < 150$	7300.0
$150 \leq s$	9000.0

- Statičku metodu `ispisiSveTroskove` koja kao ulazni parametar dobija objekat klase **Automobil** i na ekranu ispisuje sve troškove koje je potrebno platiti pri registraciji - i doprinose i osiguranje i to prema gore navedenim tarifama.

Napraviti klasu **TestTroskoviRegistracije** koja kreira objekat klase **Automobil** i postavlja vrednosti njegovih atributa na "Mercedes", "A 170", 1690 kubika, 95 konjskih snaga. Ispisati na ekranu vrednosti svih troškova registracije ovog automobila. Posle toga, izračunati koliko iznose ukupni troškovi (sabrati troškove za osiguranje i doprinose) i ispisati rezultat na ekranu.

Rešenje:

```

class Automobil {

    String marka;
    String model;
}

```

```

    int kubikaza;
    int snaga;
}

class TroskoviRegistracije {

    static double obracunajDoprinos(Automobil a){
        if ((a.kubikaza < 400) || (a.kubikaza > 10000)){
            System.out.println("Kubikaza je van granica");
            return 0;
        }
        else{
            if (a.kubikaza < 1000) return 145.5;
            if ((a.kubikaza >= 1000) && (a.kubikaza < 1400)) return 200;
            if ((a.kubikaza >= 1400) && (a.kubikaza < 2500)) return 250;
            if (a.kubikaza > 2500) return 413.3;

            //Ova poslednja return komanda nema nikakvu funkciju
            //ali je morala biti napisana da bi Java protumacila
            //da ce metoda da vrati neku vrednost cak iako se ne
            //izvrsi nijedna IF naredba.
            return 0;
        }
    }

    static double obracunajOsiguranje(Automobil a){
        if ((a.snaga < 20) || (a.snaga > 1000)){
            System.out.println("Snaga je van granica");
            return 0;
        }
        else{
            if (a.snaga < 55) return 3000;
            if ((a.snaga >= 55) && (a.snaga < 75)) return 4500;
            if ((a.snaga >= 75) && (a.snaga < 150)) return 7300;
            if (a.snaga > 150) return 9000;

            //Ova poslednja return komanda nema nikakvu funkciju
            //ali je morala biti napisana da bi Java protumacila
            //da ce metoda da vrati neku vrednost cak iako se ne
            //izvrsi nijedna IF naredba.
            return 0;
        }
    }

    static void ispisiSveTroskove(Automobil a){
        //Nema potrebe da se ponovo pise kod za izracunavanje
        //doprinosa i osiguranja, dovoljno je pozvati metode
        //koje vec postoje u okviru klase i ispisati rezultat
        //koje vrate.
        double doprinos = obracunajDoprinos(a);
        double osiguranje = obracunajOsiguranje(a);

        System.out.println("Doprinosi su: "+doprinos);
        System.out.println("Osiguranje iznosi: "+osiguranje);
    }
}

class TestTroskoviRegistracije {

```

```

public static void main(String[] args) {

    Automobil auto = new Automobil();

    auto.marka = "Mercedes";
    auto.model = "A 170";
    auto.kubikaza = 1690;
    auto.snaga = 95;

    //Ispisivanje troskova na ekranu. Ispisace se troskovi
    //doprinosa i osiguranja posebno. Ova metoda je staticka
    //pa nije potrebno inicijalizovati objekat da bi se pozvala.
    TroskoviRegistracije.ispisiSveTroskove(auto);

    double ukupno = TroskoviRegistracije.obracunajDoprinos(auto) +
        TroskoviRegistracije.obracunajOsiguranje(auto);

    System.out.println("Ukupni troskovi su: "+ukupno);
}
}

```