

FOR naredba

Kontrolisanje toka izvršavanja programa podrazumeva i rešavanje onih situacija u kojima je potrebno jednu ili više komandi ponoviti veći broj puta. U ovim slučajevima, koriste se **naredbe za ciklično ponavljanje**. Ove naredbe se nazivaju i **ciklusi (petlje)**. Jedna vrsta petlje je i FOR petlja. Deklaracija FOR petlje se vrši na sledeći način:

```
for (komanda_z1; uslov_z; komanda_z2) komanda_p;
```

Deklaracija počinje rezervisanom reči “for”, posle koje sledi zagrada sa dve komande i uslovom. Prva komanda u zagradi (komanda_z1) se **izvršava samo jedanput i to na početku**. Uslov (uslov_z) predstavlja **logički izraz koji se proverava pre izvršavanja svake iteracije (kruga)**. Ako je uslov zadovoljen, izvršiće se još jedna iteracija. Ako nije, petlja se prekida. Poslednja komanda u zagradi (komanda_z2) se **izvršava na kraju svake iteracije**. Posle zagrade se nalazi komanda koju je potrebno izvršiti više puta (komanda_p). Kada se petlja pokrene, redosled izvršavanja komandi i provere uslova je sledeći:

POČETAK PETLJE

- Izvršava se komanda_z1

1. ITERACIJA

- Proverava se uslov_z (uslov važi)
- izvršava se komanda_p
- izvršava se komanda_z2

2. ITERACIJA

- Proverava se uslov_z (uslov važi)
- izvršava se komanda_p
- izvršava se komanda_z2

3. ITERACIJA

- Proverava se uslov_z (uslov važi)
- izvršava se komanda_p
- izvršava se komanda_z2

...

POSLEDNJA ITERACIJA

- Proverava se uslov_z (uslov NE važi - petlja se prekida)

KRAJ PETLJE

Ako je potrebno ciklično ponavljanje više naredbi, potrebno ih je uokviriti u blok naredbi uz pomoć vitičastih zagrada:

```
for (komanda_z1; uslov_z; komanda_z2) {  
    komanda_p_1;  
    komanda_p_2;  
    ...  
    komanda_p_n;  
}
```

Primer 1

Napraviti klasu **Ispisivac** koja ima:

- Statičku metodu **ispisiPorukuPetPut** koja na ekranu pet puta ispisuje poruku "Dobar dan".

```
class Ispisivac {  
  
    static void ispisiPorukuPetPut() {  
        for (int i=1; i<=5; i++)  
            System.out.println("Dobar dan");  
    }  
  
}
```

Metoda `ispisiPorukuPetPut` je mogla da se realizuje i tako što bi se napisala "`System.out.println`" komanda pet puta zaredom, ali to ne bi moglo da se uradi u situaciji da je bilo potrebno da se poruka ispiše sto ili hiljadu puta. Zbog toga je korišćena FOR petlja. U ovom slučaju, napravljena je promenljiva "i" i dodata joj je početna vrednost 1. Pre svake iteracije se proverava da li je "i" manje ili jednako 5. Ako jeste, izvršava se iteracija - ispisuje se "Dobar dan" na ekranu i "i" se uvećava za 1 (i++). Ovo se radi sve dok "i" ne dobije vrednost 6 u kom slučaju se prekida petlja. Sledi kratak opis toga šta se dešava kada se pokrene petlja:

POČETAK PETLJE

- `int i = 1;` (deklariše se promenljiva "i" i dobija vrednost 1)

1. ITERACIJA (i = 1)

- `i <= 5 ?` (`1 <= 5` - uslov je ispunjen)
- `System.out.println("Dobar Dan");`
- `i++;` (`i = 2` - promenljiva je uvećana za 1)

2. ITERACIJA (i = 2)

- `i <= 5 ?` (`2 <= 5` - uslov je ispunjen)
- `System.out.println("Dobar Dan");`
- `i++;` (`i = 3` - promenljiva je uvećana za 1)

3. ITERACIJA (i = 3)

- `i <= 5 ?` (`3 <= 5` - uslov je ispunjen)
- `System.out.println("Dobar Dan");`
- `i++;` (`i = 4` - promenljiva je uvećana za 1)

4. ITERACIJA (i = 4)

- `i <= 5 ?` (`4 <= 5` - uslov je ispunjen)
- `System.out.println("Dobar Dan");`
- `i++;` (`i = 5` - promenljiva je uvećana za 1)

5. ITERACIJA (i = 5)

- `i <= 5 ?` (`5 <= 5` - uslov je ispunjen)
- `System.out.println("Dobar Dan");`
- `i++;` (`i = 6` - promenljiva je uvećana za 1)

6. ITERACIJA (i = 6)

- $i \leq 5$? ($6 > 5$ - uslov NIJE ispunjen, petlja se prekida)

KRAJ PETLJE

Promenljiva “i” iz prethodnog primera se koristi da bi se obezbedilo da se izvrši tačno 5 iteracija FOR petlje. Ova promenljiva se naziva **brojač ciklusa**. Uobičajena je praksa da se brojač ciklusa i deklariše i da mu se dodeli početna vrednost u okviru prve naredbe u okviru zagrade FOR naredbe. Uslov za izlazak iz petlje je da brojač dostigne neku vrednost, a treća komanda u okviru zagrade menja trenutnu vrednost brojača (u prethodnom primeru, uvećava ga za jedan). Sa obzirom na to da se ova treća komanda izvršava na kraju tekuće iteracije, brojač dobija novu vrednost koja se prenosi u sledeću iteraciju. Kada se brojač ciklusa deklariše u okviru prve komande u zagradi FOR naredbe, on je **“vidljiv” samo u okviru FOR petlje** i ne može se koristiti van nje.

Primer 2

Prepraviti klasu **Ispisivac** tako da ima i:

- Statičku metodu **ispisiPorukuNPut**a koja kao ulazni parametar prima jednu poruku (String) i jedan pozitivan ceo broj N. Ova metoda na ekranu ispisuje N puta unetu poruku.
- Statičku metodu **ispisiOd0Do30** koja na ekranu ispisuje brojeve od 0 do 30.
- Statičku metodu **ispisiOd30Do0** koja na ekranu ispisuje brojeve od 0 do 30 ali u obrnutom redosledu (30, 29, 28, 27, ... , 2, 1, 0).

Napraviti klasu **TestIspisivac** koja poziva sve metode klase **Ispisivac**.

Posle dopuna, kod klase **Ispisivac** izgleda ovako:

```
class Ispisivac {

    static void ispisiPorukuPetPut() {
        for (int i=1; i<=5; i++)
            System.out.println("Dobar dan");
    }

    static void ispisiPorukuNPut(String poruka, int n){
        for (int i=1; i<=n; i++)
            System.out.println(poruka);
    }

    static void ispisiOd0Do30(){
        for (int i=0; i<=30; i++)
            System.out.println(i);
    }

    static void ispisiOd30Do0(){
        for (int i=30; i>=0; i--)
            System.out.println(i);
    }

}

class TestIspisivac {

    public static void main(String[] args) {
```

```

        //Sve metode klase Ispisivac su staticke
        //pa se ne mora kreirati objekat klase
        //Ispisivac    da bi se pozvale
        Ispisivac.ispisiPorukuPetPuti();

        Ispisivac.ispisiPorukuNPuti("Zdravo", 2);

        Ispisivac.ispisiOd0Do30();

        Ispisivac.ispisiOd30Do0();

    }

}

```

U metodi “ispisiPorukuNPuti” se ulazni parametar N koristi u okviru uslova za izlazak iz petlje. Ako se, na primer, unese broj 5 kao vrednost N , petlja će izvršiti 5 iteracija (poruka će se ispisati 5 puta na ekranu) a ako se unese broj 10 izvršiće se 10 iteracija.

Metoda “ispisiOd0Do30” ispisuje na ekranu sve cele brojeve iz ovog raspona. Najjednostavnije rešenje je da se podesi tako da brojač petlje uzima vrednosti od 0 do 30 i da se u svakoj iteraciji ispiše trenutna vrednost brojača za tu iteraciju. Tako, brojač kao početnu vrednost dobija 0, na ekranu se ispisuje trenutna vrednost brojača, brojač se uvećava za 1, ponovo se ispisuje njegova trenutna vrednost itd. U poslednjoj iteraciji, brojač ima vrednost 30, koja se ispisuje na ekranu (jer je $30 \leq 30$ pa uslov važi). Kada se vrednost brojača uveća za 1 i postane 31, uslov prestaje da važi ($31 > 30$) i ciklus se prekida.

U metodi “ispisiOd30Do0” je prikazano da se brojač petlje može i umanjivati u svakom koraku ako je to potrebno. To onda, naravno, zahteva da i se početna vrednost brojača kao i uslov za izlazak iz petlje prilagode situaciji.

FOR naredba se može kombinovati i sa svim drugim naredbama za kontrolu toka programa. Tako, može da se desi da u nekom bloku naredbi FOR petlje nalazi i neka IF naredba i obrnuto. Ovo ugnježdavanje naredbi može da bude i tako da FOR naredba sadrži i drugu FOR naredbu ili neki drugi ciklus.

Primer 3

Napraviti klasu **SlozeniIspisivac** koja ima:

- Statičku metodu **ispisiParneBrojeveOd1Do25** koja na ekranu ispisuje sve brojeve u rasponu od 1 do 25 koji su parni.
- Statičku metodu **ispisiMatricu** koja na ekranu ispisuje matricu dimenzija 6×4 čiji su svi elementi jednaki 1 i to tako da se svi elementi iz istog reda matrice napišu u jednom redu na ekranu:

```

1      1      1      1
1      1      1      1
1      1      1      1
1      1      1      1
1      1      1      1
1      1      1      1

```

Napisati klasu **TestSlozeniIspisivac** koja poziva metode klase **SlozeniIspisivac**.

```
class SlozeniIspisivac {

    static void ispisiParneBrojeveOd1Do25() {
        //FOR naredba u sebi sadrzi IF naredbu
        //koja u svakoj iteraciji proverava da
        //li je trenutna vrednost brojaca paran
        //broj i ispisuje ga na ekranu ako jeste
        for(int i = 1; i<=25; i++)
            if (i%2 == 0) System.out.println(i);
    }

    static void ispisiMatricu() {
        //FOR naredba u sebi sadrzi drugu FOR
        //naredbu. Unutrasnji FOR ciklus ispisuje
        //sve elemente jednog reda na ekranu dok
        //se spoljasnjim FOR ciklusom obezbedjuje
        //da se to uradi po jednom za svaki red.
        for (int i = 1; i<=6; i++){
            for(int j = 1; j<=4; j++) System.out.print(1+" ");
            //Ova naredba ne pripada unutrasnjem
            //FOR ciklusu
            System.out.println();
        }
    }

}

class TestSlozeniIspisivac {

    public static void main(String[] args) {
        SlozeniIspisivac.ispisiParneBrojeveOd1Do25();
        SlozeniIspisivac.ispisiMatricu();
    }

}
```

Metoda "ispisiParneBrojeveOd1Do25" sadrži FOR petlju koja ima ugnježdenu IF naredbu. U svakoj iteraciji se proverava da li je trenutna vrednost brojača paran broj pa se, ako jeste, ispisuje na ekranu. Sa obzirom na to da brojač uzima vrednosti od 1 do 25, efekat je taj da će se ispisati svi parni brojevi u tom rasponu - što se i tražilo.

Metoda "ispisiMatricu" sadrži FOR petlju koja ima ugnježdenu drugu FOR petlju. Prvo, potrebno je primetiti da svaka petlja ima različit brojač. Spoljašnja petlja ima brojač "i" dok unutrašnja ima brojač "j". Unutrašnja petlja ispisuje elemente iz jednog reda matrice. Sa obzirom na to da je potrebno ispisati ove elemente u jednom redu na ekranu, koristi se "print" a ne "println" naredba. "System.out.println" naredba koja sledi unutrašnju petlju ne pripada ovoj petlji a koristi se da bi se posle ispisivanja elemenata iz jednog reda matrice prešlo u sledeći red na ekranu. Spoljašnja petlja služi tome da se obezbedi ispisivanje elemenata za svih šest redova.

Uobičajen način za izlazak iz FOR petlje je da uslov koji se proverava na početku svake iteracije

prestane da važi. Međutim, moguće je prekinuti FOR petlju i na drugi način, korišćenjem **“break” naredbe**. Znači, ako se u nekoj iteraciji FOR petlje izvrši komanda “break”, **petlja se istog trenutka prekida**. Treći način za prekidanje FOR petlje proizilazi iz karakteristika “return” naredbe. Ako metoda u kojoj se koristi FOR petlja vraća neku vrednost, **petlja se može prekinuti pozivanjem “return” naredbe**. Sporedni efekat je taj da će se na ovaj način prekinuti i izvršavanje cele metode. I **WHILE i DO-WHILE petlje se mogu prekinuti korišćenjem ove dve komande**.

U Javi postoji i naredba čijim izvršavanjem se obezbeđuje da se preskoči trenutna iteracija i da se nastavi sa sledećom iteracijom petlje. To je **“continue” naredba**. Kada se izvrši, sve naredbe koje bi trebalo da se izvrše u toj iteraciji, a koje su napisane “ispod” ove naredbe se preskaču, i prelazi se u sledeću iteraciju. Jedina komanda koja se regularno izvršava je treća komanda u zagradi FOR petlje kojom se povećava tj. smanjuje vrednost brojača.

Primer 4

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

- Statičku metodu **ispisiMinDeljivSa12i15i9** koja na ekranu ispisuje najmanji broj u rasponu od 10 do 1000 koji je deljiv sa 12, 15 i 9 istovremeno.
- Statičku metodu **vратиMinDeljivSa12i15i9** koja vraća najmanji broj u rasponu od 10 do 1000 koji je deljiv sa 12, 15 i 9 istovremeno.

Kod tražene dve metode je sledeći:

```
static void ispisiMinDeljivSa12i15i9() {
    for(int i = 10; i<=1000;i++)
        if ((i%12 == 0)&&(i%15 == 0)&&(i%9==0)) {
            System.out.println(i);
            break;
        }
}

static int vratiMinDeljivSa12i15i9() {
    for(int i = 10; i<=1000;i++)
        if ((i%12 == 0)&&(i%15 == 0)&&(i%9==0))
            return i;

    return 0;
}
```

U prvoj metodi, prolazi se kroz brojeve od 10 do 1000 korišćenjem FOR petlje. Ova petlja ima ugnježdenu IF naredbu tako da se u svakom koraku proverava da li je trenutna vrednost brojača deljiva sa 12, 15 i 9 istovremeno. Ako nije, prelazi se u sledeću iteraciju. Međutim, kada se nađe broj koji jeste deljiv sa 12, 15 i 9, ispisuje se na ekranu i izvršava se “break” naredba koja prekida FOR petlju. Da nije napisana ova “break” naredba, FOR petlja bi se nastavila i ispisala bi sve brojeve u rasponu od 10 do 1000 koji su deljivi sa 12, 15 i 9 istovremeno a ne samo najmanji.

Druga metoda je skoro ista kao i prva sa tim što se, čim se nađe broj koji je deljiv sa 12, 15 i 9, taj broj vraća korišćenjem “return” naredbe. Ova naredba prekida izvršavanje FOR petlje ali i cele metode.

Zadaci

Zadatak 1

Napraviti klasu **Ispisivac2** koja ima:

- Statičku metodu koja ispisuje brojeve od 1 do 50 na ekranu u rastućem redosledu (1,2,3,4...,49,50).
- Statičku metodu koja ispisuje brojeve od 0 do 30 na ekranu u rastućem redosledu (0,1,2,3...,29,30).
- Statičku metodu koja ispisuje brojeve od 50 do 1 na ekranu u opadajućem redosledu (50,49,48,47,...,3,2,1).
- Statičku metodu koja ispisuje brojeve od 32 do -2 na ekranu u opadajućem redosledu (32,31,30, ..., 3,2,1,0,-1,-2).
- Statičku metodu koja kao ulazni argument dobija ceo broj veci od nule i na ekranu ispisuje sve cele brojeve od nule do tog broja u rastućem redosledu (0,1,2,...,n).
- Statičku metodu koja kao ulazni argument dobija ceo broj manji od nule i na ekranu ispisuje sve cele brojeve od nule do tog broja u opadajućem redosledu (0,-1,-2,-3,...,-n - pri čemu je $n < 0$).
- Statičku metodu koja kao ulazni argument dobija dva cela broja A i B i ispisuje sve brojeve koji su veci od A a manji od B uključujući i A i B (A je uvek manje od B).
- Statičku metodu koja kao ulazni argument dobija dva cela broja A i B i ispisuje sve brojeve koji su veci od A a manji od B ali ne uključujući i A i B (A je uvek manje od B).
- Statičku metodu koja kao ulazni argument dobija dva cela broja A i B i ispisuje sve brojeve koji su veci od A a manji od B (uključujući i A i B) ali u obrnutom redosledu – od vecih ka manjim brojevima (A je uvek manje od B).

Napraviti klasu **TestProveraIspisivaca2** koja poziva sve metode klase Ispisivac2.

Rešenje:

```
class Ispisivac2 {

    static void ispisiOd1Do50() {
        for (int i=1; i<=50; i++) System.out.println(i);
    }

    static void ispisiOd0Do30() {
        for (int i=0; i<=30; i++) System.out.println(i);
    }

    static void ispisiOd50Do1() {
        for (int i=50; i>=1; i--) System.out.println(i);
    }

    static void ispisiOd32DoMinus2() {
        for (int i=32; i>=-2; i--) System.out.println(i);
    }

    static void ispisiOd0DoN (int n) {
        for (int i=0; i<=n; i++) System.out.println(i);
    }

    static void ispisiOd0DoMinusN (int n) {
        for (int i=0; i>=n; i--) System.out.println(i);
    }

    static void ispisiOdADoB (int a, int b) {
        for (int i=a; i<=b; i++) System.out.println(i);
    }

    static void ispisiOdADoBBezAB (int a, int b) {
        for (int i=a+1; i<b; i++) System.out.println(i);
    }

    static void ispisiOdBDoA (int a, int b) {
        for (int i=b; i>=a; i--) System.out.println(i);
    }

}
```

```

class TestIspisivac2 {

    public static void main (String[] args){

        Ispisivac2.ispisiOd1Do50();
        Ispisivac2.ispisiOd0Do30();
        Ispisivac2.ispisiOd50Do1();
        Ispisivac2.ispisiOd32DoMinus2();

        Ispisivac2.ispisiOd0DoN(12);
        Ispisivac2.ispisiOd0DoMinusN(-3);
        Ispisivac2.ispisiOdADoB(10,20);
        Ispisivac2.ispisiOdADoBBezAB(10, 20);
        Ispisivac2.ispisiOdBDoA(2, 8);

    }

}

```

Zadatak 2

Napisati klasu **NaucniKalkulator**. Ova klasa bi trebalo da ima:

- Statičku metodu koja računa i vraća sumu od N (N je ceo broj i daje se u vidu ulaznog argumenta)

$$S(N) = 1+2+3+4+5+...+(N-1)+N$$
- Statičku metodu koja računa i vraća faktoriyel (N je ceo broj i daje se u vidu ulaznog argumenta) $N! = 1*2*3*4*5*...*(N-1)*N$
- Statičku metodu koja računa i vraća 2 na N-ti stepen, pri čemu je N ceo broj i daje se u vidu ulaznog argumenta.
- Statičku metodu koja računa i vraća 3 na N-ti stepen, pri čemu je N ceo broj i daje se u vidu ulaznog argumenta.
- Statičku metodu koja računa i vraća X na N-ti stepen pri čemu je N ceo broj, a X realan broj i oba su data u vidu ulaznih argumenata.
- Statičku metodu koja računa i vraća (A+B) na N-ti stepen pri čemu su A, B i N celi brojevi i ulazni argumenti.
- Statičku metodu koja računa i vraća sumu svih celih brojeva koji se nalaze u intervalu od A do B (uključujući i A i B) pri čemu je $A < B$, i oba broja su pozitivni celi brojevi. A i B su dati u vidu ulaznog argumenta.
- Statičku metodu koja računa i vraća sumu svih celih brojeva koji se nalaze u intervalu od A do B (ne uključujući B) pri čemu je $A < B$, i oba broja su pozitivni celi brojevi. A i B su dati u vidu ulaznog argumenta.
- Statičku metodu koja računa i vraća sumu svih parnih brojeva koji se nalaze u intervalu od A do B uključujući i A i B. A i B su dati u obliku ulaznih argumenata.
- Statičku metodu koja računa i vraća sumu svih brojeva deljivih sa tri koji se nalaze u intervalu od A do B uključujući i A i B. A i B su dati u obliku ulaznih argumenata.
- Statičku metodu koja računa i vraća proizvod svih pozitivnih brojeva koji se nalaze u intervalu od A do B uključujući i A i B. A i B su dati u obliku ulaznih argumenata, a A može biti i manje od nula.

Napraviti klasu **ProveraNaucnogKalkulatora** koja proverava rad klase Kalkulator pozivanjem nekih njenih metoda i ispisivanjem rezultata koji metode vraćaju na ekranu uz odgovarajuću poruku.

Rešenje:

```

class NaucniKalkulator {

    static int suma (int n){
        int s=0;
        for (int i=1; i<=n; i++) s=s+i;
        return s;
    }

    static int faktoriyel (int n){
        int f=1;
        for (int i=1; i<=n; i++) f=f*i;
        return f;
    }

}

```



```

static int dvaNaN (int n){
    int p=1;
    for (int i=1; i<=n; i++) p=p*2;
    return p;
}

static int triNaN (int n){
    int p=1;
    for (int i=1; i<=n; i++) p=p*3;
    return p;
}

static double xNaN (double x, int n){
    double rezultat = 1;
    for (int i=1; i<=n; i++) rezultat=rezultat*x;
    return rezultat;
}

static int aPlusBNaN (int a, int b, int n){
    int rezultat = 1;
    for (int i=1; i<=n; i++) rezultat=rezultat*(a+b);
    return rezultat;
}

static int sumaAB (int a, int b){
    int s = 0;
    for (int i=a; i<=b; i++) s=s+i;
    return s;
}

static int sumaABBezB (int a, int b){
    int s = 0;
    for (int i=a; i<b; i++) s=s+i;
    return s;
}

static int sumaParnihAB (int a, int b){
    int s = 0;
    for (int i=a; i<=b; i++) if ((i%2) == 0) s=s+i;
    return s;
}

static int sumaDeljivihSaTriAB (int a, int b){
    int s = 0;
    for (int i=a; i<=b; i++) if ((i%3) == 0) s=s+i;
    return s;
}

static int sumaPozitivnihAB (int a, int b){
    int s = 0;
    for (int i=a; i<=b; i++) if (i>0) s=s+i;
    return s;
}

}

class ProveraNaucnogKalkulatora {

    public static void main(String[] args){

        // Ispisivanje sume od 5 (5+4+3+2+1=15)
        System.out.println ("Suma broja 5 je "+NaucniKalkulator.suma(5));
    }
}

```

```

        // Ispisivanje faktorijela od 4 (4*3*2*1=24)
        System.out.println ("Faktoriyel broja 4 je "+
            NaucniKalkulator.faktoriyel(4));

        // Ispisivanje 2 na 4-ti stepen (2^4=16)
        System.out.println ("Cetvrti stepen broja 2 je "+
            NaucniKalkulator.dvaNaN(4));

        // Ispisivanje X na n-ti stepen (3.0^3=27.0)
        System.out.println ("Treci stepen broja 3.0 je "+
            NaucniKalkulator.xNaN(3.0,3));

        // Ispisivanje a+b na n-ti stepen ((2+3)^3=5^3=125)
        System.out.println ("Treci stepen (2+3) je "+
            NaucniKalkulator.aPlusBNaN(2, 3, 3));

    }

}

```

Zadatak 3

Napisati klasu **IspisivacMatrica**. Ova klasa bi trebalo da ima:

- Statičku metodu koja kao na ekranu ispisuje matricu dimenzije 4x4 i to tako da se svi elementi iz jednog reda matrice napišu u jednom redu na ekranu. Matrica koj je potrebno ispisati je:

```

1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4

```

- Statičku metodu koja kao ulazni argument prima ceo broj N i na ekranu ispisuje jediničnu matricu dimenzije NxN i to tako da se svi elementi iz jednog reda matrice napišu u jednom redu na ekranu. Ako je uneti broj N manji od nule, trebalo bi na ekranu ispisati poruku o grešci. Jedinična matrica je kvadratna matrica čiji su svi elementi glavne dijagonale jednaki 1 a ostali jednaki 0. Primer jedinične matrice dimenzije 3x3 je:

```

1 0 0
0 1 0
0 0 1

```

Napraviti klasu **TestIspisivacMatrica** koja poziva metode klase IspisivacMatrica.

Rešenje:

```

class IspisivacMatrica {

    static void ispisi4X4() {
        //Svi elementi jednog reda su isti
        //i jednaki su broju tog reda
        //pa je dovoljno ispisati brojac "i"
        //cija vrednost je uvek jednaka
        //broju reda.
        for (int i = 1; i <= 4; i++) {
            for (int j = 1; j <= 4; j++)
                System.out.print(i + " ");

            System.out.println();
        }
    }

    static void ispisiJedinicnuMatricu(int n) {
        //Brojac "i" u stvari broji redove matrice
        //a "j" broji kolone. Kada je vrednost "i"

```

```

//jednaka "j" element koji je na tom mestu
//pripada dijagonali matrice npr. [1,1],
//[2,2], [3,3],...[n,n] pa ce njegova vrednost
//biti 1, a u suprotnom 0.
if (n < 0)
    System.out.println("Greska!");
else for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++)
        if (i==j) System.out.print(1 + " ");
        else System.out.print(0 + " ");

        System.out.println();
    }
}

class TestIspisivacMatrica {

    public static void main(String[] args) {
        IspisivacMatrica.ispisi4X4();
        IspisivacMatrica.ispisiJedinicnuMatricu(5);
    }
}

```