

Java

primeri

Studenti koji uvide greške u ovim materijalima, imaju neke primedbe, predloge, pohvale ili na drugi način žele da pomognu u pripremi materijala za ovaj kurs, mogu se javiti na e-mail:

drazen.draskovic@etf.bg.ac.rs

/verzija 16.12.2012./

Primer 1 - Hello World

primer1\HelloWorld.java

Ime javne klase mora biti jednako imenu datoteke!!!

```
package primer1; //oznaka za paket u kome se nalazi klasa

public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello world!");

    } //kraj main metode
} //kraj klase
```

Napomene:

//	je oznaka za komentar u jednoj liniji koda
/* komentar */	je oznaka za komentar u više linija
package	je oznaka za paket i biće kasnije objašnjena

Primer 2 - Sabiranje celih brojeva

primer2\Zbir.java

```
package primer2;

public class Zbir {

    public static void main(String[] args) {
        int a = 5;
        int b = 3;
        int c = a+b; //pomocna promenljiva za sabiranje

        System.out.println("Zbir unetih brojeva je: " + c);
        //moze i ovako sa ispisivanjem unetih brojeva:

        System.out.println("Zbir unetih brojeva " + a + " i " + b + " je " + c);
        //promenljive moraju da se pisu van navodnika, one nisu string!

        int e = 2;
        int f = -7;
        //moze i bez izracunavanja zbira u pomocnu promenljivu

        System.out.println("Zbir unetih brojeva 2 i -7 je: " + (e+f));
        //e+f mora da se pise u zagradi zbog prioriteta

    } //main
} //class
```

Primer 3 - Korišćenje matematičkih funkcija

primer3\Kvadrat.java

Za unetu realnu vrednost (tip double) stranice kvadrata, izračunati dijagonalu tog kvadrata.

```
package primer3;
import java.lang.Math;

public class Kvadrat {

    public static void main(String[] args) {
        double a = 5.75; //stranica kvadrata

        System.out.println("Stranica kvadrata je: " + a);

        double d = a*Math.sqrt(2);

        //sqrt je ugradjena metoda u klasi Math koja daje koren realnog broja

        System.out.println("Dijagonala kvadrata je: " + d);

    }
}
```

primer3\Krug.java

Za unetu realnu vrednost (tip double) poluprečnika kruga, izračunati obim i površinu tog kruga.

```
package primer3;
import java.lang.Math;

public class Krug {

    public static void main(String[] args) {
        double r = 3.23; //poluprecnik kruga

        System.out.println("Poluprecnik kruga je: " + r);

        System.out.println("Obim kruga je: " + (2*r*Math.PI));

        /*  kod izracunavanja obima i povrsine kruga mora da se koristi
            broj PI iz ugradjene klase Math, koja se nalazi u paketu java.lang

            Paket se ucitava na pocetku programa, ispod navedenog imena paketa
            i koristi se rec import.
        */

        System.out.println("Povrsina kruga je: " + (r*r*Math.PI));
    }
}
```

Primer 4 - Naredba IF-ELSE

Napisati na jeziku Java program za nalaženje najmanjeg i najvećeg elementa od tri cela broja.

primer4\MinMax.java

```
package primer4;

public class MinMax {
    public static void main(String[] args) {
        int a = 37;
        int b = 54;
        int c = 28;

        int min = a, max = a;

        if (b > max) max = b;
            else if (b < min) min = b;

        if (c > max) max = c;
            else if (c < min) min = c;

        System.out.println("Minimum je: " + min);
        System.out.println("Maksimum je: " + max);
    }
}
```

Primer 5 - Niz celih brojeva i FOR petlja

primer5\Niz.java

```
package primer5;

public class Niz {

    public static void main(String[] args) {

        int[] niz = {22,5,8,3,13,16,4,2};
        //definisanje niza celih brojeva

        //u for petlji moze da se definise pomocna promenljiva i
        //svaki niz ima svoj argument length koji oznacava broj elemenata niza

        for(int i=0; i<niz.length-1; i++)
            for(int j=i+1; j<niz.length; j++) //for petlja u spoljasnjoj for petlji
                if(niz[i]>niz[j]) { //if blok u okviru unutrasnje for petlje
                    int t = niz[i]; //pomocna promenljiva t
                    niz[i] = niz[j];
                    niz[j] = t;
                }

        //ispisivanje
        for(int i=0; i<niz.length; i++)
            System.out.println(niz[i]);
    }
}
```

primer5\Niz2.java

```
package primer5;

public class Niz2 {

    public static final int velicina = 10;

    public static void main(String[] args) {

        int[] niz = new int[velicina];
        //deklaracija niza celih brojeva (inicijalizuje se nulama!)

        for(int i=1; i<niz.length; i++) {
            niz[i] = i*i;
            System.out.println(niz[i]);
        }
        //Niz se popunjava kvadratima brojeva od 1 do 9
    }
}
```

Primer 6 - Prosti brojevi

Napisati program na jeziku Java koji ispisuje sve proste brojeve iz skupa od prvih 100 celih brojeva.

primer6\ProstBroj.java

```
package primer6;

public class ProstBroj {

    public static void main(String[] args) {

        System.out.println("Prosti brojevi izmedju 1 i 100 su: ");

        for(int i=1; i<100; i++) {
            boolean prost=true;
            for(int j=i-1; j>1; j--){
                if(i%j == 0) prost=false;
            }
            if(prost) System.out.println(i);
        }
    }
}
```

Primer 7 - Rad sa Stringom

primer7\Imena.java

```
package primer7;

public class Imena {

    public static void main(String[] args) {

        String s1 = "Ana";
        String s2 = "Marina";

        //Ispitivanje da li su 2 objekta tipa String jednaka vrši se metodom equals
        //string1.equals(string2)
        if(s1.equals(s2))
            System.out.println("Stringovi su isti!");
        else
            System.out.println("Stringovi nisu isti.");
        //Rezultat ovog primera je da ova 2 stringa nisu jednaka

        //Uporedjivanje 2 objekta tipa String vrši se metodom compareTo
        System.out.println(s1.compareTo("ana"));

        //Ova metoda će vratiti u našem primeru rezultat -32, jer stringovi
        //s1="Ana" i "ana" nisu jednaki, a 32 je razlika u vrednosti između
        //malih i velikih slova u ASCII tabeli

        System.out.println(s1.equalsIgnoreCase("aNa"));

        //Ova metoda upoređuje da li su stringovi jednaki isključujući
        //osobinu case-sensitive
```

```
//A ovo je niz stringova

String[] ljudi = {"Vesna", "Aleksa", "Pavle", "Sanja", "Nikola", "Milos", "Nada"};

//Niz pomocu 2 petlje sortiramo u rastucem abecednom redosledu od A do Z

for(int i=0; i<ljudi.length; i++)
    for(int j=i+1; j<ljudi.length; j++){
        if(ljudi[i].compareTo(ljudi[j])>0){
            String t=ljudi[i];
            ljudi[i]=ljudi[j];
            ljudi[j]=t;
        }
    }

//Ispisivanje sortiranog niza

for(int i=0; i<ljudi.length; i++)
    System.out.println(ljudi[i]);
}
}
```

Primer 8 - Korišćenje Scanner

Napisati program koji sa standardnog ulaza učitava 2 cela broja i izračunava njihov zbir. Koristite klasu `Scanner`.

primer8\Skener.java

```
package primer8;
import java.util.Scanner;

public class Skener {

    public static void main(String[] args) {

        int a, b;
        Scanner sc = new Scanner(System.in);

        System.out.println("Unesite 2 cela broja: ");

        a = sc.nextInt();
        b = sc.nextInt();

        System.out.println("Zbir unetih brojeva je: " + (a+b));
    }
}
```

Primer 9 - Rad sa matricom

primer9\Matrica.java

Napisati program na Javi koji koristeći klasu Scanner učitava broj vrsta i kolona matrice, a zatim sa standardnog ulaza učitava sve elemente matrice. Potrebno je izračunati i ispisati na standardnom izlazu:

a) zbir svih elemenata matrice;

Program treba da se izvršava sve dok se za broj redova ili kolona ne unese negativan broj ili nula.

```
package primer9;
import java.util.Scanner;

public class Matrica {

    public static void main(String[] args) {
        int[][] matrica;
        //ovo je samo deklaracija matrice

        while(true){
            Scanner s = new Scanner(System.in);

            System.out.println("Unesite broj redova?");
            int m = s.nextInt();

            System.out.println("Unesite broj kolona?");
            int n = s.nextInt();

            //ako se unese negativan broj ili nula za broj redova ili kolona, program se prekida
            if(m<=0 || n<=0) break;

            matrica = new int[m][n];
            //ovde matrici zadajemo dimenzije koje smo uneli i rezervisemo prostor u memoriji

            System.out.println("Unesite elemente matrice:");
            for(int i=0; i<m; i++){
                System.out.println((i+1) + ".vrsta:");
                for(int j=0; j<n; j++)
                    matrica[i][j]=s.nextInt();
            }
        }
    }
}
```

```

    }

    int zbir=0;

    for(int i=0; i<m; i++)
        for (int j=0; j<n; j++)
            zbir+=matrica[i][j];

    System.out.println("Zbir svih elemenata: " + zbir);
}
}
}

```

Modifikacije prethodnog zadatka za vežbu:

- b) zbir svih elemenata na glavnoj dijagonali (samo u slučaju da je matrica kvadratna);**
- c) zbir svih elemenata na sporednoj dijagonali (samo u slučaju da je matrica kvadratna);**
- d) najmanji i najveći element matrice.**

Primer 10 - Korišćenje BufferedReader *

primer10\ImeGodine.java

```
package primer10;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class ImeGodine {
    public static void main(String[] args) {
        InputStreamReader ulaz = new InputStreamReader(System.in);
        BufferedReader bafer = new BufferedReader(ulaz);
        try{
            while(true){
                System.out.println("Unesite ime?");
                String ime = bafer.readLine();

                System.out.println("Koliko imate godina?");
                String temp = bafer.readLine();
                int godine = Integer.parseInt(temp);

                System.out.println("Zovete se: " + ime + " i imate " + godine + " god.");
            }
        }catch(IOException e){}
    }
}
```

**** opširnije u odeljku I/O***

Primer 11 - Pojam klase i objekta (kvadrat)

primer11\Kvadrat.java

```
package primer11;

public class Kvadrat {
    private int a; // stranica kvadrata

    //konstruktori
    public Kvadrat() { a = 1; }
    public Kvadrat (int str) { a = str; }

    public int dohvati() { //metoda koja vraca stranicu
        return a;
    }

    public int površina(){ //metoda koja izracunava površinu kvadrata
        return a*a;
    }

    public int obim(){ //metoda koja izracunava obim kvadrata
        return 4*a;
    }
}
```

primer11\Main.java

```
package primer11;

public class Main {
    public static void main(String[] args) {

        Kvadrat k1 = new Kvadrat();
        Kvadrat k3 = new Kvadrat(3);
        Kvadrat k6 = new Kvadrat(6);

        double pov = k1.povrsina();

        System.out.println("Povrsina kvadrata k1 je " + pov);
        System.out.println("Obim kvadrata k3 je " + k3.obim());
        System.out.println("Povrsina kvadrata stranice " + k6.dohvati() + " je " + k6.povrsina());
    }
}
```

Primer 12 - Pojam klase i objekta (osoba)

primer12\Osoba.java

```
package primer12;

public class Osoba {
    public static int punoletna = 18;
    public String ime = "neki string";
    public String prezime;
    public int godine;

    /** Creates a new instance of Osoba */
    public Osoba(String i, String p, int g) {
        ime = i;
        prezime = p;
        godine = g;
    }

    public int brojGodina() {
        return godine;
    }
}
```

primer12\RadOsoba.java

```
package primer12;

public class RadOsoba {
    /** Creates a new instance of RadOsoba */
    public RadOsoba() { }

    public static void main(String[] args) {
        Osoba o1 = new Osoba("Zoran", "Cirovic", 29);
        Osoba o2 = new Osoba("Jelena", "Jovanovic", 22);

        int result = o1.brojGodina();
        System.out.println("Zoran ima " + result + "god.");
    }
}
```

Izvedene klase

Primer 13 - Izvedene klase (valjak->kanta)

primer13\Valjak.java

```
package primer13;
import java.lang.Math;

public class Valjak {
    private double r, h;

    public Valjak(double rr, double hh) { r=rr; h=hh; }
    public Valjak() { r=h=1; }

    public double r(){
        return r;
    }

    public double h(){
        return h;
    }

    public double V(){
        return r*r*Math.PI*h;
    }

    public String toString(){
        return "[r=" + r + ", h=" + h + "]\n";
    }
}
```

primer13\Kanta.java

```
package primer13;

public class Kanta extends Valjak {
    private double pp; //popunjenost

    public Kanta(double rr, double hh, double pp) { //npr. k2=new Kanta(2,3,4);
        super(rr,hh);
        this.pp = pp<=V()?pp:V();
    }
    public Kanta(double rr, double hh){ //npr. k3 = new Kanta(3,5);
        this(rr,hh,0);
    }
    //metode
    public double ima(){ //da li i koliko je popunjena kanta
        return pp;
    }
    public boolean puna(){ return pp==V(); } //da li je kanta puna do vrha
    public boolean prazna() { return pp==0; } //da li je kanta skroz prazna

    public Kanta presipaj(Kanta k) throws GreskaP, GreskaIK {
        if(this!=k){
            double prazno=V()-pp;
            if(prazno>=k.pp){
                pp+=k.pp;
                k.pp=0;
            }
            else throw new GreskaP();
        }
        else throw new GreskaIK();
        return this;
    }

    public String toString(){
        return "[" + super.toString() + " ," + pp + "]";
    }
}
```

primer13\GreskaP.java

```
package primer13;
public class GreskaP extends Exception {
    public String toString(){ return "Greska! Nemate dovoljno praznog prostora u vasoj kanti"; }
}
```

primer13\GreskaIK.java

```
package primer13;
public class GreskaIK extends Exception {
    public String toString(){ return "Greska! Pokusavate da presipate iz supljeg u prazno"; }
}
```

primer13\Main.java

```
package primer13;
import java.io.*;

public class Main {
    public static void main(String[] args) {

        try{
            Valjak v1 = new Valjak(1,5);
            System.out.println(v1);
            Kanta k2 = new Kanta(2,3,20);
            System.out.println(k2);
            Kanta k3 = new Kanta(3,4,80);
            System.out.println(k3);
            System.out.println("k3<-k2: " + k3.presipaj(k2));
        } catch(GreskaP gp) { System.out.println(gp); }
        catch(GreskaIK gi) { System.out.println(gi); }
    }
}
```

Primer 14 - Izvedene klase (geometrijska tela)

primer14\GeometrijskoTelo.java

```
package primer14;

public abstract class GeometrijskoTelo {
    private double stezina;

    public GeometrijskoTelo(double t) { stezina = t; }
    public GeometrijskoTelo() { stezina = 1; }

    public abstract char vrsta();

    public abstract double zapremina();

    public String toString() {
        return vrsta() + "[" + Double.toString(stezina) + "|";
    }
}
```

primer14\Lopta.java

```
package primer14;

public class Lopta extends GeometrijskoTelo {
    private double r;

    public Lopta(double st, double rr) { super(st); r=rr; }
    public Lopta() { this(1,1); }

    public char vrsta() { return 'L'; }

    public double zapremina() {
        return 4./3*r*r*r*Math.PI;
    }

    public String toString() {
        return super.toString() + r + "]";
    }
}
```

primer14\Kocka.java

```
package primer14;

public class Kocka extends GeometrijskoTelo{
    private double a;

    public Kocka(double st, double aa) { super(st); a=aa; }
    public Kocka() { this(1,1); }

    public char vrsta() { return 'K'; }

    public double zapremina() { return a*a*a; }

    public String toString(){
        return super.toString() + a + " ";
    }
}
```

primer14\Main.java

```
package primer14;

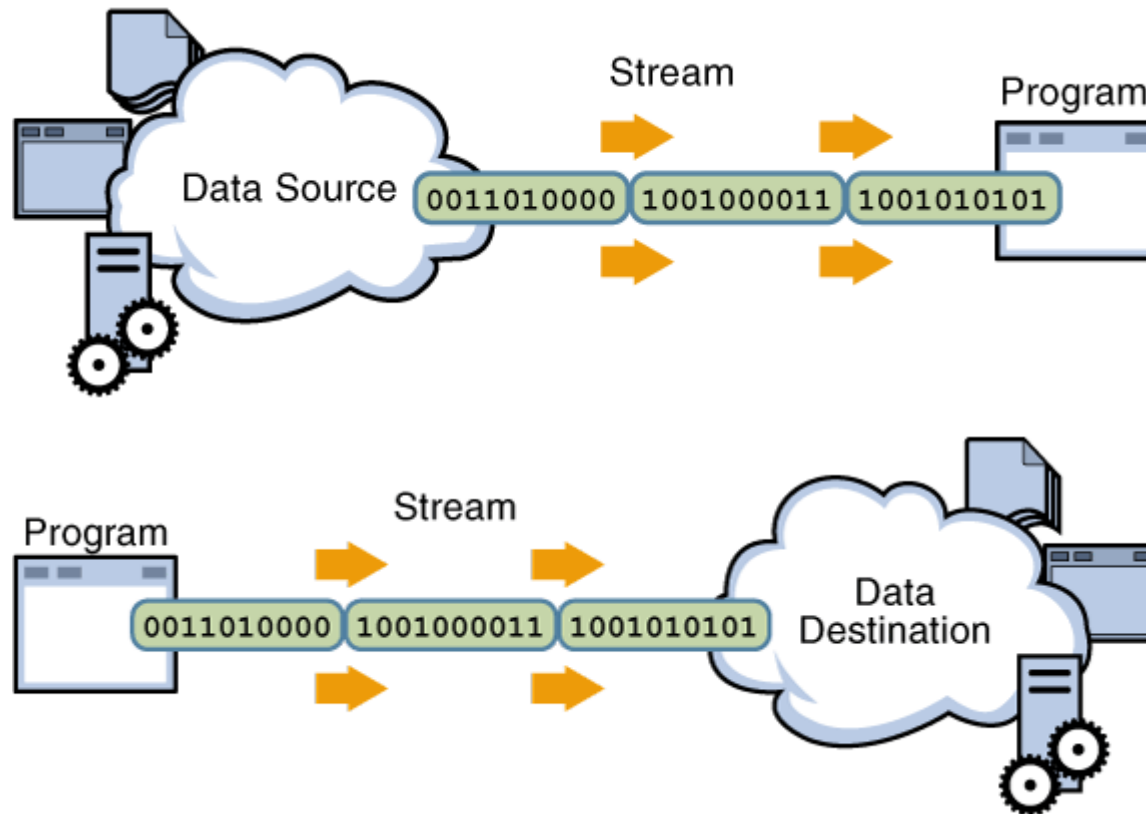
public class Main {
    public static void main(String[] varg){

        // ne moze da se napravi objekat tipa geometrijsko telo:
        // GeometrijskoTelo p = new GeometrijskoTelo(5);

        // Niz predmeta moze da se napravi
        GeometrijskoTelo[] g = new GeometrijskoTelo[100];
        g[0] = new Lopta();
        g[1] = new Kocka(3.3,5.0);
        //ali ovo ne moze: g[2] = new GeometrijskoTelo(2);
        g[3] = new Kocka();
        Lopta lop = new Lopta(2,2);

        System.out.println(g[0]);
        System.out.println(g[1]);
        System.out.println(g[3]);
        System.out.println(lop);
    }
}
```

Input Output



Primer 15 - Kopiranje ulaza na izlaz (bajt po bajt)

primer15\CopyBytes.java

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CopyBytes {
    public static void main(String[] args) throws IOException {
        FileInputStream ulaz = null;
        FileOutputStream izlaz = null;
        try {
            ulaz = new FileInputStream("unos.txt");
            izlaz = new FileOutputStream("ispis.txt");
            int c;

            while ((c = ulaz.read()) != -1) {
                izlaz.write(c);
            }

        } finally {
            if (ulaz != null) {
                ulaz.close();
            }
            if (izlaz != null) {
                izlaz.close();
            }
        }
    }
}
```

Primer 16 - Kopiranje ulaza na izlaz (karakter po karakter)

primer16\CopyCharacters.java

Za razliku od kopiranja bajtova, postoje klase specijalizovane za ulaz-izlaz karaktera `FileReader` i `FileWriter`. Sledeća klasa pokazuje takav primer:

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class CopyCharacters {
    public static void main(String[] args) throws IOException {
        FileReader inputStream = null;
        FileWriter outputStream = null;

        try {
            inputStream = new FileReader("unos.txt");
            outputStream = new FileWriter("ispis.txt");

            int c;
            while ((c = inputStream.read()) != -1) {
                outputStream.write(c);
            }
        } finally {
            if (inputStream != null) {
                inputStream.close();
            }
            if (outputStream != null) {
                outputStream.close();
            }
        }
    }
}
```

`CopyCharacters` je vrlo slična klasi `CopyBytes`. Najvažnija razlika je u tome što `CopyCharacters` koristi `FileReader` i `FileWriter` za ulaz i izlaz. U našim primerima i `CopyBytes` i `CopyCharacters` koriste varijablu `int` kao pomoćnu promenljivu za isčitavanje i upisivanje.

Primer 17 - Kopiranje ulaza na izlaz (linijski orijentisani I/O)

U ovom primeru koristićemo dve klase koje ranije nismo koristili - `BufferedReader` i `PrintWriter`.

primer17\CopyLines.java

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;

public class CopyLines {
    public static void main(String[] args) throws IOException {
        BufferedReader inputStream = null;
        PrintWriter outputStream = null;

        try {
            inputStream =
                new BufferedReader(new FileReader("unos.txt"));
            outputStream =
                new PrintWriter(new FileWriter("ispis.txt"));

            String l;
            while ((l = inputStream.readLine()) != null) {
                outputStream.println(l);
            }
        } finally {
            if (inputStream != null) {
                inputStream.close();
            }
            if (outputStream != null) {
                outputStream.close();
            }
        }
    }
}
```

Rad sa Buffered Streams

Most of the examples we've seen so far use *unbuffered* I/O. This means each read or write request is handled directly by the underlying OS. This can make a program much less efficient, since each such request often triggers disk access, network activity, or some other operation that is relatively expensive.

To reduce this kind of overhead, the Java platform implements *buffered* I/O streams. Buffered input streams read data from a memory area known as a *buffer*; the native input API is called only when the buffer is empty. Similarly, buffered output streams write data to a buffer, and the native output API is called only when the buffer is full.

A program can convert an unbuffered stream into a buffered stream using the wrapping idiom we've used several times now, where the unbuffered stream object is passed to the constructor for a buffered stream class. Here's how you might modify the constructor invocations in the `CopyCharacters` example to use buffered I/O:

```
InputStream =  
    new BufferedReader(new FileReader("xanadu.txt"));  
OutputStream =  
    new BufferedWriter(new FileWriter("characteroutput.txt"));
```

There are four buffered stream classes used to wrap unbuffered streams: [BufferedInputStream](#) and [BufferedOutputStream](#) create buffered byte streams, while [BufferedReader](#) and [BufferedWriter](#) create buffered character streams.

Flushing Buffered Streams

It often makes sense to write out a buffer at critical points, without waiting for it to fill. This is known as *flushing* the buffer.

Some buffered output classes support *autoflush*, specified by an optional constructor argument. When autoflush is enabled, certain key events cause the buffer to be flushed. For example, an autoflush `PrintWriter` object flushes the buffer on every invocation of `println` or `format`. See [Formatting](#) for more on these methods.

To flush a stream manually, invoke its `flush` method. The `flush` method is valid on any output stream, but has no effect unless the stream is buffered.

Primer 18 - Ispitni zadatak (maj, jun 2008, zadatak #2)

- a) Napisati program koji od korisnika prihvata podatke o realizovanim koncertima: korisnik unosi ime grupe, mesto održavanja, broj posetilaca koji je prisustvovao koncertu i datum koncerta, sve dok se ne unese vrednost 0 za ime. Dobijene podatke potrebno je upisati u tekstualni fajl pod imenom "Br_indeksa-koncert", gde je Br_indeksa broj vašeg indeksa.
- b) Prikazati koliko je bilo koncerata sa brojem posetilaca većim od 10000. Podatke treba dobiti čitanjem iz fajla formiranog u prethodnoj tački.

```
package ispit;
import java.io.*;

public class Koncert {

    public static void main(String[] args){
        String[] poruke={"Ime grupe: ", "Mesto odrzavanja: ", "Broj ljudi: ", "Datum
koncerta: " };
        boolean kraj=false;
        try{
            BufferedReader ulaz=new BufferedReader(new InputStreamReader(System.in));
            BufferedWriter izlaz=new BufferedWriter(new FileWriter("etf0404-
koncert.txt"));
            String linija="";

            while(!kraj){
                for(int i=0; i<4; i++) {
                    System.out.println(poruke[i]);
                    linija=ulaz.readLine();
                    if(kraj = (i==0 && linija.equals("0")) ) break;
                    izlaz.write(linija);
                    izlaz.newLine();
                }
            }
            ulaz.close();
```

```

        izlaz.close();
        BufferedReader citaj = new BufferedReader(new FileReader("etf0404-
koncert.txt"));
        int brojac =0;
        int i=0;

        while(linija!=null){
            linija=citaj.readLine();
            if(i==2 && Integer.parseInt(linija)>10000) brojac++;
            i++;
            if(i==4) i=0;
        }
        citaj.close();

        System.out.println("Koncerata sa posetom vecom od 10000 je " +brojac);

    }
    catch(IOException e) {
        System.out.println("GRESKA NA ULAZU/IZLAZU!");
    }
}

```