

## Ulazno-izlazni tokovi u Javi

U programskom jeziku Java postoji i podrška za rad sa perifernim uređajima. Povezivanje i rad sa nekim uređajem se vrši korišćenjem tzv. "tokova" ("stream"). Ako je u pitanju uređaj sa koga se preuzimaju podaci (npr. tastatura ili skener), koristi se neka vrsta ulaznog toka, dok se za uređaje na koje se šalju podaci (npr. monitor ili štampač) koristi neka vrsta izlaznog toka. U Javi se monitor računara smatra za standardni izlazni tok ("System.out"), dok se tastatura deklariše kao standardni ulazni tok ("System.in"). Čak se i upis i čitanje iz fajlova vrši preko tokova.

Klase za rad sa tokovima u Javi su već implementirane i nalaze se u paketu "java.io". Izbor odgovarajuće klase zavisi od tipa podatka koji se prenosi.

### Učitavanje sa tastature

Sa stanovišta programskog jezika Java, tastatura predstavlja standardni ulazni tok računara (za razliku od monitora koji je standardni izlazni tok). Referenciranje na tastaturu iz Jave se vrši korišćenjem klase System i njenog statičkog atributa - "**System.in**". Ovaj atribut je u stvari objekat klase **InputStream** koja ima samo osnovne metode za učitavanje bajt po bajt. Rad sa tastaturom na ovako elementarnom nivou uopšte nije jednostavan i dodatne funkcionalnosti su potrebne. Zbog toga se koristi klasa **InputStreamReader**. Ova klasa učitava više bajtova i pretvara ih u slova. Međutim, ovakvo učitavanje može da bude neefikasno bez neke privremene memorije ("buffer") u koju bi se smestio deo niza slova dok se niz ne učitava do kraja. Klasa **BufferedReader** upravo postoji zbog toga da poveća efikasnost ovog učitavanja. Prema tome, otvaranje ulaznog toka za učitavanje sa tastature izgleda ovako.

```
BufferedReader naziv_promenljive =  
    new BufferedReader(new InputStreamReader(System.in));
```

Klasa BufferedReader ima **metodu "readLine"** koja omogućava učitavanje celog niza znakova sa tastature. Ova metoda vraća String u kojem se nalaze svi znakovi koje je korisnik uneo preko tastature sve dok trenutka dok nije pritisnuo taster za kraj reda (znak "enter"). Metoda "readLine" baca proveravani izuzetak klase IOException, pa je svaki poziv ove metode potrebno uokviriti "try-catch" blokom.

#### Primer 1

Napraviti klasu **UcitavanjeSaTastature** koja ima:

- Javnu statičku metodu koja sa tastature učitava String i ispisuje ga na ekranu uz odgovarajuću poruku.

```
import java.io.*;  
  
public class UcitavanjeSaTastature {  
  
    public static void ucitajIIsipisiString() {  
        BufferedReader br = new BufferedReader(  
            new InputStreamReader(System.in));  
  
        try {  
            System.out.print("Unesite poruku: ");
```

```

        String s = br.readLine();

        System.out.println("Uneli ste: "+s);
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}
}

```

Da bi učitavanje sa tastature uopšte moglo da počne, mora se otvoriti odgovarajući tok. Prema tome, kreira se jedan objekat klase *BufferedReader* i njemu se kroz konstruktor prosleđuje objekat klase *InputStreamReader* koji dobija pokazivač na standardni ulaz - "System.in". Kada se to uradi, može se pozvati metoda "readLine" da učitava red slova. Poziv ove metode je uokviren u "try-catch" blok. Metoda vraća učitani red slova kao *String* i kasnije ga ispisuje na ekranu.

Kada se neki red znakova učitava sa tastature, metoda "readLine" vraća *String* koji sadrži te znakove. Međutim, šta se dešava ako je potrebno sa tastature učitati neki broj? U tom slučaju, **učitani *String* se mora pretvoriti u broj**. Klase *Double*, *Integer* i *Boolean* koje su već pominjane u poglavlju o listama sadrže odgovarajuće statičke metode koje rade upravo konverziju iz *String* vrednosti u neki prost tip: "**parseDouble**", "**parseInt**" i "**parseBoolean**". Metode "parseDouble" i "parseInt" bacaju izuzetke tipa *NumberFormatException* ako uneti *String* ne može da se uspešno konvertuje u broj.

## Primer 2

Dodati u klasu *UcitavanjeSaTastature* i sledeće:

- Statičku metodu koja sa tastature učitava realan broj i ispisuje na ekranu njegov kvadrat uz odgovarajuću poruku.
- Statičku metodu koja sa tastature učitava ceo broj i ispisuje na ekranu da li je deljiv sa dva.

Napisati klasu *TestUcitavanjeSaTastature* koja poziva sve metode klase *UcitavanjeSaTastature*.

```

public static void ucitajBrojIispisiKvadrat() {
    BufferedReader br = new BufferedReader(
        new InputStreamReader(System.in));

    try{
        System.out.print("Unesite realan broj: ");
        String s = br.readLine();

        double d = Double.parseDouble(s);
        System.out.println("Kvadrat je: "+(d*d));

    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void ucitajBrojIispisiParnost() {
    BufferedReader br = new BufferedReader(
        new InputStreamReader(System.in));
}

```

```

try{
    System.out.print("Unesite ceo broj: ");
    String s = br.readLine();

    int i = Integer.parseInt(s);
    if (i%2==0)
        System.out.println("Broj je paran");
    else
        System.out.println("Broj je neparan");
} catch (Exception e) {
    System.out.println("Greska: "+e.getMessage());
}
}

public class TestUcitavanjeSaTastature {

    public static void main(String[] args) {

        UcitavanjeSaTastature.ucitajIIspisiString();

        UcitavanjeSaTastature.ucitajBrojIIspisiKvadrat();

        UcitavanjeSaTastature.ucitajBrojIIspisiParnost();

    }

}

```

Pored ovog načina, postoji još jedan način za učitavanje sa tastature. U paketu “java.util” se nalazi klasa **Scanner** koja omogućava malo jednostavnije učitavanje sa tastature ili iz tekstualnog fajla. Prva pogodnost koju ova klasa uvodi je ta što sve njene metode za učitavanje bacaju samo neproveravane izuzetke pa se “try-catch” blokovi ne moraju pisati. Ovo, međutim, i nije toliko velika prednost jer će u slučaju da bude bačen izuzetak (a nije napisan “try-catch” blok), program prekinuti sa radom. Druga pogodnost je ta što metode ove klase mogu da učitaju i istovremeno konvertuju učitani niz znakova u neki prost tip i to sve u jednom pozivu.

Pre učitavanja sa tastature potrebno je napraviti objekat klase Scanner. Konstruktor ove klase ima samo jedan ulazni parametar, a to je ulazni tok iz kojeg se želi čitati.

```
Scanner naziv_promenljive = new Scanner(System.in);
```

Za učitavanje jednog reda znakova, potrebno je pozvati **“nextLine” metodu**. Ova metoda vraća String koji sadrži sve znakove iz učitanoj reda teksta. **Metoda “nextInt”** učitava jedan red znakova sa tastature, pretvara ga u ceo broj i vraća taj broj kao povratnu vrednost. **Metode “nextDouble” i “nextBoolean”** vrše sličnu funkciju samo što se kao rezultat dobija realan broj tj. boolean vrednost.

### Primer 3

*Uraditi zadatak iz prethodna dva primera, ali korišćenjem Scanner klase i njenih metoda.*

```

public static void ucitajIIspisiString() {
    Scanner in = new Scanner(System.in);

```

```

        System.out.print("Unesite poruku: ");

        String s = in.nextLine();

        System.out.println("Uneli ste: "+s);
    }

```

*Kao što se vidi iz prethodnog koda, metode klase UcitavanjeSaTastature se, korišćenjem klase Scanner, mogu napisati veoma lako i brzo. Problem je u tome što, ako se desi neki izuzetak, program istog trenutka prekida sa radom. Zbog toga je ipak neophodno napisati "try-catch" blok oko svake metode koja može da baci neki izuzetak.*

```

import java.util.Scanner;

public class UcitavanjeSaTastature {

    public static void ucitajIISPisiString() {
        Scanner in = new Scanner(System.in);

        try {
            System.out.print("Unesite poruku: ");

            String s = in.nextLine();

            System.out.println("Uneli ste: "+s);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    public static void ucitajBrojIISPisiKvadrat() {
        Scanner in = new Scanner(System.in);

        try {
            System.out.print("Unesite realan broj: ");

            double d = in.nextDouble();

            System.out.println("Kvadrat je: "+(d*d));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    public static void ucitajBrojIISPisiParnost() {
        Scanner in = new Scanner(System.in);

        try {
            System.out.print("Unesite ceo broj: ");

            int i = in.nextInt();

            if (i%2==0)

```

```

        System.out.println("Broj je paran");
    else
        System.out.println("Broj je neparan");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

}

public class TestUcitavanjeSaTastature {

    public static void main(String[] args) {

        UcitavanjeSaTastature.ucitajIIispisiString();

        UcitavanjeSaTastature.ucitajBrojIIispisiKvadrat();

        UcitavanjeSaTastature.ucitajBrojIIispisiParnost();

    }

}

```

## Zadaci

### Zadatak 1

Napisati klasu **UcitavanjeSaTastature2** koja ima:

- Javnu statičku metodu koja sa tastature učitava neki tekst i na ekranu ispisuje broj rečenica u tom tekstu. Smatra se da se rečenica uvek završava tačkom.
- Javnu statičku metodu koja sa tastature učitava rečenicu i na ekranu ispisuje broj reči u toj rečenici.
- Javnu statičku metodu koja sa tastature učitava neki tekst i na ekranu ispisuje broj pojavljivanja reči “sneg” u toj rečenici. Potrebno je uzeti u obzir i sva pojavljivanja ove reči na početku ili na kraju rečenice. Rečenica se uvek završava tačkom.
- Javnu statičku metodu koja sa tastature učitava realan broj koji predstavlja neku distancu u metrima i na ekranu ispisuje tu istu distancu ali izraženu u kilometrima.
- Javnu statičku metodu koja sa tastature učitava više realnih brojeva. Brojevi se unose sve dok se umesto broja ne unese reč “kraj”. Metoda onda na ekranu ispisuje njihovu sumu.
- Javnu statičku metodu koja sa tastature učitava ceo broj i na ekranu ispisuje njegov faktorijel. Ako je uneti broj jednak nuli ili manji od nule, na ekranu se ispisuje jedinica.
- Javnu statičku metodu koja sa tastature učitava više celih brojeva. Brojevi se unose sve dok se umesto broja ne unese reč “kraj”. Metoda onda na ekranu ispisuje njihov proizvod.

Napisati klasu **TestUcitavanjeSaTastature2** poziva metode klase UcitavanjeSaTastature.

### Rešenje:

```

import java.io.*;

public class UcitavanjeSaTastature2 {

    public static void UcitajIIispisiBrojRecenica() {
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        try{
            System.out.print("Unesite tekst: ");

```

```

        String s = br.readLine();

        int brojac = 0;
        for (int i=0; i<s.length();i++)
            if (s.charAt(i) == '.')
                brojac++;

        System.out.println("Broj recenica u tekstu je: "+brojac);
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajIIspisiBrojReci() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    try{
        System.out.print("Unesite recenicu: ");

        String s = br.readLine();

        String[] reci=s.split(" ");
        System.out.println("Broj reci u recenici je: "+reci.length);
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajIIspisiBrojPonavljanjaReciSneg() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    try{
        System.out.print("Unesite recenicu: ");

        String s = br.readLine();

        String[] reci=s.split(" ");
        int brojac = 0;

        for(int i=0; i< reci.length;i++)
            if (reci[i].equalsIgnoreCase("sneg") ||
                reci[i].equals("sneg."))
                brojac++;

        System.out.println("Broj ponavljanja reci 'sneg' u recenici je: "
                           +brojac);
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajMetreIIspisiKilometre() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    try{
        System.out.print("Unesite distancu izrazenu u metrima: ");

        String s = br.readLine();

        double metri = Double.parseDouble(s);
        System.out.println("Distanca u kilometrima je: "+

```

```

        (metri/1000));
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajViseRealnihBrojevaIIspisiSumu() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    boolean kraj = false;
    double suma = 0;

    while (!kraj) {
        try {
            System.out.print("Unesite realan broj: ");

            String s = br.readLine();

            if (s.equals("kraj"))
                kraj = true;
            else
                suma = suma + Double.parseDouble(s);

        } catch (Exception e) {
            System.out.println("Greska: "+e.getMessage());
        }
    }

    System.out.println("Suma brojeva je: "+suma);
}

public static void UcitajCeoBrojIIspisiFaktorijel() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    try {
        System.out.print("Unesite ceo broj: ");

        String s = br.readLine();

        int b = Integer.parseInt(s);
        int faktorijel = 1;

        for (int i=1; i<=b; i++)
            faktorijel=faktorijel*i;

        System.out.println("Faktorijel je: "+faktorijel);
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajViseCelihBrojevaIIspisiProizvod() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    boolean kraj = false;
    int proizvod = 1;

    while (!kraj) {
        try {
            System.out.print("Unesite ceo broj: ");

```

```

        String s = br.readLine();

        if (s.equals("kraj"))
            kraj = true;
        else
            proizvod = proizvod * Integer.parseInt(s);

    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

System.out.println("Proizvod brojeva je: "+proizvod);
}

}

public class TestUcitavanjeSaTastature2 {

    public static void main(String[] args) {

        UcitavanjeSaTastature2.UcitajIISPISIbrojRecenica();
        UcitavanjeSaTastature2.UcitajIISPISIbrojReci();
        UcitavanjeSaTastature2.UcitajIISPISIbrojPonavljanjaReciSneg();
        UcitavanjeSaTastature2.UcitajMetreIISPISIKilometre();
        UcitavanjeSaTastature2.UcitajViseRealnihBrojevaIISPISISumu();
        UcitavanjeSaTastature2.UcitajCeoBrojIISPISIFaktorijel();
        UcitavanjeSaTastature2.UcitajViseCelihBrojevaIISPISIProizvod();

    }

}

```

## Zadatak 2

Rešiti prethodni zadatak ali korišćenjem Scanner klase za učitavanje sa tastature.

### Rešenje:

```

import java.util.Scanner;

public class UcitavanjeSaTastature3 {

    public static void UcitajIISPISIbrojRecenica() {
        Scanner in = new Scanner(System.in);

        try {
            System.out.print("Unesite tekst: ");

            String s = in.nextLine();

            int brojac = 0;
            for (int i=0; i<s.length();i++)
                if (s.charAt(i) == '.')
                    brojac++;

            System.out.println("Broj recenica u tekstu je: "+brojac);
        } catch (Exception e) {
            System.out.println("Greska: "+e.getMessage());
        }
    }

    public static void UcitajIISPISIbrojReci() {
        Scanner in = new Scanner(System.in);
    }
}

```



```

    try{
        System.out.print("Unesite recenicu: ");

        String s = in.nextLine();

        String[] reci=s.split(" ");
        System.out.println("Broj reci u recenici je: "+reci.length);
    }catch (Exception e){
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajIIispisiBrojPonavljanjaReciSneg(){
    Scanner in = new Scanner(System.in);

    try{
        System.out.print("Unesite recenicu: ");

        String s = in.nextLine();

        String[] reci=s.split(" ");
        int brojac = 0;

        for(int i=0; i< reci.length;i++)
            if (reci[i].equalsIgnoreCase("sneg") ||
                reci[i].equals("sneg."))
                brojac++;

        System.out.println("Broj ponavljanja reci 'sneg' u recenici je: "
            +brojac);
    }catch (Exception e){
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajMetreIIispisiKilometre(){
    Scanner in = new Scanner(System.in);

    try{
        System.out.print("Unesite distancu izrazenu u metrima: ");

        double metri = in.nextDouble();

        System.out.println("Distanca u kilometrima je: "+
            (metri/1000));
    }catch (Exception e){
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajViseRealnihBrojevaIIispisiSumu(){
    Scanner in = new Scanner(System.in);

    double suma = 0;

    //Kraj se moze ocekivati kada se unese rec
    //"kraj". U tom slucaju ce "nextDouble" metoda
    //da baci izuzetak i taj iyuyetak ce da prekine
    //while petlju.
    try{
        while (true){
            System.out.print("Unesite realan broj: ");

            suma += in.nextDouble();
        }
    }
}

```

```

        }
    } catch (Exception e) {
        System.out.println("Kraj ucitavanja");
    }

    System.out.println("Suma brojeva je: "+suma);
}

public static void UcitajCeoBrojIIspisiFaktorijel() {
    Scanner in = new Scanner(System.in);

    try {
        System.out.print("Unesite ceo broj: ");

        int b = in.nextInt();
        int faktorijel = 1;

        for (int i=1; i<=b; i++)
            faktorijel=faktorijel*i;

        System.out.println("Faktorijel je: "+faktorijel);
    } catch (Exception e) {
        System.out.println("Greska: "+e.getMessage());
    }
}

public static void UcitajViseCelihBrojevaIIspisiProizvod() {
    Scanner in = new Scanner(System.in);

    int proizvod = 1;

    try {
        while (true) {
            System.out.print("Unesite ceo broj: ");

            proizvod*= in.nextInt();
        }
    } catch (Exception e) {
        System.out.println("Kraj ucitavanja");
    }

    System.out.println("Proizvod brojeva je: "+proizvod);
}

}

public class TestUcitavanjeSaTastature3 {

    public static void main(String[] args) {

        UcitavanjeSaTastature3.UcitajIIspisiBrojRecenica();
        UcitavanjeSaTastature3.UcitajIIspisiBrojReci();
        UcitavanjeSaTastature3.UcitajIIspisiBrojPonavljanjaReciSneg();
        UcitavanjeSaTastature3.UcitajMetreIIspisiKilometre();
        UcitavanjeSaTastature3.UcitajViseRealnihBrojevaIIspisiSumu();
        UcitavanjeSaTastature3.UcitajCeoBrojIIspisiFaktorijel();
        UcitavanjeSaTastature3.UcitajViseCelihBrojevaIIspisiProizvod();
    }

}

```

### Zadatak 3

Napraviti klasu **NizInt** koja ima:

- Privatni atribut niz koji predstavlja niz celih brojeva.
- Javnu metodu koja sa tastature učitava ceo broj i inicijalizuje niz na taj kapacitet. Ako je uneti kapacitet nula ili manji od nule, potrebno je inicijalizovati niz na 10 elemenata. U slučaju da je bačen izuzetak zbog greške pri unosu kapaciteta sa tastature, potrebno je ispisati poruku o grešci i postaviti kapacitet niza na 10 elemenata.
- Javnu metodu koja, kada se pozove, sa tastature učitava sve elemente niza. Prvo je potrebno proveriti da li je niz inicijalizovan. Ako niz nije inicijalizovan, metoda baca izuzetak i ne vrši se učitavanje.
- Javnu metodu koja na ekranu ispisuje elemente niza. Ako niz nije inicijalizovan, metoda baca izuzetak .

Napraviti klasu **TestNizInt** koja kreira jedan objekat klase NizInt i poziva njegove metode za inicijalizaciju i unos elemenata, a nakon toga za ispis.

#### Rešenje:

```
import java.io.*;

public class NizInt {

    private int[] niz;

    public void unesiKapacitet() {
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        try{
            System.out.print("Unesite kapacitet niza: ");
            String s=br.readLine();

            int kapacitet = Integer.parseInt(s);
            if (kapacitet<=0)
                niz = new int[10];
            else
                niz = new int[kapacitet];

        } catch (Exception e) {
            System.out.println("Greska pri unosu kapaciteta: "
                +e.getMessage());
            niz = new int[10];
        }
    }

    public void unesiElemente() {
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        if (niz==null)
            throw new RuntimeException("Niz nije inicijalizovan");

        for(int i=0; i<niz.length;i++){
            try{
                System.out.print("Unesite "+i+". element niza: ");
                String s=br.readLine();

                int b = Integer.parseInt(s);
                niz[i] = b;
            } catch (Exception e) {
                System.out.println("Greska pri unosu "+i+". elementa: "
                    +e.getMessage());
            }
        }
    }

    public void ispisi() {
```

```

        if (niz==null)
            throw new RuntimeException("Niz nije inicijalizovan");

        for (int i=0;i<niz.length;i++)
            System.out.println(i+". element:"+niz[i]);
    }

}

public class TestNizInt {

    public static void main(String[] args) {

        NizInt ni = new NizInt();

        ni.unesiKapacitet();
        ni.unesiElemente();
        ni.ispisi();

    }

}

```

#### Zadatak 4

Napraviti klasu **NizDouble** koja ima:

- Privatni atribut niz koji predstavlja niz realnih brojeva.
- Javnu metodu koja sa tastature učitava ceo broj i inicijalizuje niz na taj kapacitet. Ako je uneti kapacitet nula ili manji od nule, potrebno je baciti izuzetak sa odgovarajućom porukom. U slučaju da je bačen izuzetak zbog greške pri unosu kapaciteta sa tastature, potrebno je ispisati poruku o grešci.
- Javnu metodu koja, kada se pozove, sa tastature učitava sve elemente niza. Prvo je potrebno proveriti da li je niz inicijalizovan. Ako niz nije inicijalizovan, metoda baca izuzetak i ne vrši se učitavanje.
- Javnu metodu koja sumira elemente niza i vraća njihovu sumu. Ako niz nije inicijalizovan, metoda baca izuzetak.

Napraviti klasu **TestNizDouble** koja kreira jedan objekat klase NizDouble i poziva njegove metode za inicijalizaciju i unos elemenata, a nakon toga ispisuje sumu elemenata niza na ekranu.

#### Rešenje:

```

import java.io.*;

public class NizDouble {

    private double[] niz;

    public void unesiKapacitet() {
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        try{
            System.out.print("Unesite kapacitet niza: ");
            String s=br.readLine();

            int kapacitet = Integer.parseInt(s);
            if (kapacitet<=0)
                throw new RuntimeException("Kapacitet niza mora biti "+
                    "veci od nule");

            niz = new double[kapacitet];
        } catch (Exception e) {
            System.out.println("Greska pri unosu kapaciteta: "
                +e.getMessage());
        }
    }
}

```

```

    }

    public void unesiElemente() {
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        if (niz==null)
            throw new RuntimeException("Niz nije inicijalizovan");

        for(int i=0; i<niz.length;i++){
            try{
                System.out.print("Unesite "+i+". element niza: ");
                String s=br.readLine();

                double element = Double.parseDouble(s);
                niz[i] = element;
            } catch (Exception e) {
                System.out.println("Greska pri unosu "+i+". elementa: "
                    +e.getMessage());
            }
        }
    }

    public double suma() {
        if (niz==null)
            throw new RuntimeException("Niz nije inicijalizovan");

        double suma = 0;
        for (int i=0;i<niz.length;i++) suma+=niz[i];

        return suma;
    }
}

public class TestNizDouble {

    public static void main(String[] args) {

        NizDouble nd = new NizDouble();

        nd.unesiKapacitet();
        nd.unesiElemente();
        System.out.println("Suma elemeata niza je: "+nd.suma());

    }
}

```

## Zadatak 5

Napraviti javnu klasu **ParkingMesto** koja ima:

- Privatni atribut slobodno koji predstavlja indikator zauzetosti parking mesta. Ovaj indikator ima vrednost TRUE ako je mesto slobodno a FALSE ako nije.
- Privatni atribut registarskiBroj koji predstavlja registarski broj vozila koje se nalazi na tom parking mestu (String).
- Odgovarajuće javne get i set metode za ova dva atributa.
- Redefinisani toString metodu klase Object koja vraća String koji sadrži sve podatke o parking mestu. Ako je parking mesto slobodno, vraća se String "Slobodno", a ako nije, onda se vraća podatak o registarskom broju kola koja se nalaze na tom mestu.

Napraviti javnu klasu **Parking** koja ima:

- Privatni atribut mesta koji predstavlja niz objekata klase ParkingMesto.

- Javnu metodu koja sa tastature učitava ceo broj i inicijalizuje atribut mesta na taj kapacitet. Ako je uneti kapacitet nula ili manji od nule, potrebno je inicijalizovati mesta na 20 elemenata. U slučaju da je bačen izuzetak zbog greške pri unosu kapaciteta sa tastature, potrebno je ispisati poruku o grešci i postaviti kapacitet parkinga na 20 elemenata. U svakom slučaju, potrebno je inicijalizovati svako pojedinačno parking mesto tako da bude slobodno (čak i u slučaju bačenog izuzetka).
- Javnu metodu koja, kada se pozove, sa tastature učitava podatke o svakom parking mestu. Prvo je potrebno proveriti da li je niz inicijalizovan. Ako niz nije inicijalizovan, metoda baca izuzetak i ne vrši se učitavanje. Ako niz jeste inicijalizovan učitavanje se vrši na sledeći način: podaci za svako parking mesto se unose u vidu jednog String-a. Ako String ima vrednost "slobodno" (nebitno da li piše velikim ili malim slovima), parking mesto treba da bude slobodno a registarski broj treba da bude null String. Ako uneti String ima bilo koju drugu vrednost, to je onda registarski broj vozila na tom mestu (npr. "BG 123-456").
- Javnu metodu koja na ekranu ispisuje sva parking mesta. Ako atribut mesta nije inicijalizovan, metoda baca izuzetak sa odgovarajućom porukom.

Napraviti javnu klasu **TestParking** koja kreira jedan objekat klase Parking, inicijalizuje ga na 5 mesta i unosu u njega podatke o tri vozila za prva tri parking mesta, dok se preostala dva parking mesta smatraju slobodnim.

### Rešenje:

```
public class ParkingMesto {

    private boolean slobodno;

    private String registarskiBroj;

    public String getRegistarskiBroj() {
        return registarskiBroj;
    }

    public void setRegistarskiBroj(String registarskiBroj) {
        this.registarskiBroj = registarskiBroj;
    }

    public boolean isSlobodno() {
        return slobodno;
    }

    public void setSlobodno(boolean slobodno) {
        this.slobodno = slobodno;
    }

    public String toString(){
        if (slobodno)
            return "Slobodno";
        else
            return registarskiBroj;
    }

}

import java.io.*;

public class Parking {

    private ParkingMesto[] mesta;

    public void unesiKapacitet(){
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        try{
            System.out.print("Unesite kapacitet parkinga: ");
            String s=br.readLine();
```

```

        int kapacitet = Integer.parseInt(s);
        if (kapacitet<=0)
            mesta = new ParkingMesto[20];
        else
            mesta = new ParkingMesto[kapacitet];
    } catch (Exception e) {
        System.out.println("Greska pri unosu kapaciteta parkinga: "
            +e.getMessage());
        mesta = new ParkingMesto[20];
    }

    for(int i=0;i<mesta.length;i++){
        mesta[i] = new ParkingMesto();
        mesta[i].setSlobodno(true);
    }
}

public void unesiElemente() {
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));

    if (mesta==null)
        throw new RuntimeException("Parking nije inicijalizovan");

    for(int i=0; i<mesta.length;i++){
        try{
            System.out.print("Unesite podatke za "+i+
                ". parking mesto: ");
            String s=br.readLine();

            if (s.equalsIgnoreCase("slobodno")) {
                mesta[i].setSlobodno(true);
                mesta[i].setRegistarskiBroj(null);
            }
            else{
                mesta[i].setSlobodno(false);
                mesta[i].setRegistarskiBroj(s);
            }
        } catch (Exception e) {
            System.out.println("Greska pri unosu "+i+". elementa: "
                +e.getMessage());
        }
    }
}

public void ispisi() {
    if (mesta==null)
        throw new RuntimeException("Parking nije inicijalizovan");

    for (int i=0;i<mesta.length;i++)
        System.out.println(i+". parking mesto:"+mesta[i]);
}

}

public class TestParking {

    public static void main(String[] args) {

        Parking p = new Parking();

        p.unesiKapacitet();
    }
}

```

```

        p.unesiElemente();
        p.ispisi();
    }
}

```

## Zadatak 6

Napisati klasu **Knjiga** koja ima:

- Privatni atribut naslov.
- Privatni atribut ISBN (String).
- Privatni atribut autor.
- Javne get i set metode za ove attribute. Nedorazumljive vrednosti za sva tri atributa su null String-ovi. U slučaju unosa ovih vrednosti, potrebno je baciti izuzetak sa odgovarajućom porukom.
- Redefinisano metodu toString klase Object koja vraća String sa svim podacima o knjizi uz odgovarajući tekst.
- Redefinisano equals metodu klase Object koja kao ulazni argument dobija objekat klase Object. Metoda prvo proverava da li je uneti objekat klase Knjiga, pa ako nije baca izuzetak. Metoda vraća true ako su ISBN i ISBN unete knjige jednaki. Ako ISBN i ISBN unete knjige nisu jednaki metoda vraća false.

Napraviti klasu **Biblioteka** koja ima:

- Privatni atribut knjige koji predstavlja listu objekata klase Knjiga. Ovu listu je potrebno odmah inicijalizovati.
- Javnu metodu koja sa tastature učitava sve podatke o knjizi, i dodaje novu knjigu u biblioteku. Dodavanje se vrši samo ako u biblioteci već ne postoji knjiga sa istim ISBN. Ako knjiga sa istim ISBN postoji, potrebno je baciti izuzetak sa odgovarajućom porukom.
- Javnu metodu koja sa tastature učitava podatke o više knjiga i unosi ih u bazu (broj knjiga koje je potrebno uneti nije unapred poznat). Posle unosa svake knjige, potrebno je pitati korisnika da li želi da unese još jednu knjigu. Ako korisnik odgovori sa "DA", vrši se unos još jedne knjige, a u suprotnom se izvršavanje metode prekida.
- Javnu metodu koja na ekranu ispisuje sadržaj biblioteke. Ako je biblioteka prazna, ispisuje se poruka o tome.

Napraviti klasu **TestBiblioteka** koja kreira jedan objekat klase Biblioteka, u njega unosi dve knjige i ispisuje na ekranu sadržaj biblioteka.

## Rešenje:

```

public class Knjiga {

    private String naslov;
    private String autor;
    private String ISBN;

    public String getAutor() {
        return autor;
    }
    public void setAutor(String autor) {
        if (autor == null)
            throw new RuntimeException("Autor ne sme da bude null");

        this.autor = autor;
    }
    public String getISBN() {
        return ISBN;
    }
    public void setISBN(String isbn) {
        if (isbn == null)
            throw new RuntimeException("ISBN ne sme da bude null");

        ISBN = isbn;
    }
    public String getNaslov() {
        return naslov;
    }
}

```



```

    public void setNaslov(String naslov) {
        if (naslov == null)
            throw new RuntimeException("Naslov ne sme da bude null");

        this.naslov = naslov;
    }

    public String toString(){
        return "Naslov: "+naslov+" Autor: "+autor+" ISBN: "+ISBN;
    }

    public boolean equals (Object o){
        if (!(o instanceof Knjiga))
            throw new RuntimeException("Morate uneti objekat klase "+
                "Knjiga");

        Knjiga k = (Knjiga)(o);
        if (ISBN.equals(k.getISBN()))
            return true;
        else
            return false;
    }
}

import java.io.*;
import java.util.LinkedList;

public class Biblioteka {

    private LinkedList <Knjiga> knjige = new LinkedList <Knjiga>();

    public void ucitajKnjigu(){
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        try{
            Knjiga k = new Knjiga();

            System.out.print("Unesite naslov knjige: ");
            k.setNaslov(br.readLine());

            System.out.print("Unesite autora knjige: ");
            k.setAutor(br.readLine());

            System.out.print("Unesite ISBN knjige: ");
            k.setISBN(br.readLine());

            if (knjige.contains(k))
                throw new RuntimeException("Ta knjiga vec postoji u "+
                    "biblioteci");

            knjige.add(k);
        } catch (Exception e) {
            System.out.println("Greska: "+e.getMessage());
        }
    }

    public void ucitajViseKnjiga(){
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));

        boolean kraj = false;

```

```

        while (!kraj){
            ucitajKnjigu();
            try{
                System.out.print("Da li zelite da unesete jos neku "
                    +"knjigu(DA/NE)? ");
                String s = br.readLine();

                if (!s.equals("DA"))
                    kraj = true;
            } catch (Exception e){
                System.out.println("Greska: "+e.getMessage());
            }
        }

        public void ispisi(){
            if (knjige.size() == 0)
                System.out.println("Bibiblioteka je prazna");
            else{
                for (int i=0;i<knjige.size();i++)
                    System.out.println(knjige.get(i));
            }
        }
    }

    public class TestBiblioteka {

        public static void main(String[] args) {

            Biblioteka b = new Biblioteka();

            b.ucitajViseKnjiga();
            b.ispisi();

        }
    }

```