

Laboratorijske vežbe – dvočas br. 4

Zadatak 1

(radi ga laborant u saradnji sa studentima)

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

- Atribut niz koji predstavlja niz celih brojeva. *(deklarisanje niza)*
- Atribut brojac koji predstavlja trenutni broj elemenata u nizu. Početna vrednost je 0.
- Konstruktor u kome se niz kreira tako da mu maksimalni kapacitet bude jednak vrednosti koja se prosleđuje konstruktoru u obliku ulaznog argumenta. *(konstruktor, inicijalizacija niza, kapacitet niza)*
- Metodu koja na ekranu ispisuje prvi element niza. *(pristup elementima niza preko indeksa, indeksi kreću od nule)*
- Metodu koja na ekranu ispisuje maksimalni kapacitet niza. *(length - kapacitet niza)*
- Metodu za dodavanje novog elementa (tj. broja) u niz; broj koji se dodaje prosleđuje se metodi u vidu ulaznog argumenta. Pre unosa, potrebno je proveriti da li u nizu još ima praznih mesta. Ako nema, ispisati poruku o grešci na ekranu.
- Metodu za prikaz sadržaja niza na ekranu.
- Metodu za proveru da li se određeni broj nalazi u nizu; broj se prosleđuje metodi u vidu ulaznog argumenta; u slučaju da je traženi broj u nizu metoda vraća vrednost true, u suprotnom vraća false.
- Metodu za računanje zbira brojeva u nizu; izračunati zbir se vraća kao povratna vrednost metode.
- Metodu za pronalaženje najvećeg broja u nizu; pronađeni broj se vraća kao povratna vrednost metode.

Napraviti klasu **TestNizCelihBrojeva** koja kreira jedan objekat klase NizCelihBrojeva i postavlja kapacitet niza da bude 4. Uneti u niz brojeve 5 i 19 i ispisati sve elemente niza na ekranu. *(pozivanje konstruktora)*

```
class NizCelihBrojeva {  
  
    int[] niz;  
    int brojac = 0;  
  
    NizCelihBrojeva(int kapacitet) {  
        niz = new int[kapacitet];  
    }  
  
    void ispisiPrvi() {  
        System.out.println(niz[0]);  
    }  
  
    void ispisiKapacitet() {  
        System.out.println(niz.length);  
    }  
  
    void dodajUNiz(int b) {  
        if (brojac < niz.length) {  
            niz[brojac] = b;  
            brojac++;  
        }  
        else System.out.println("Greska! Niz je pun!");  
    }  
}
```

```

    }

    void ispisi() {
        for (int i = 0; i < brojac; i++)
            System.out.println(niz[i]);
    }

    boolean daLiSadrzi(int b) {
        for (int i = 0; i < brojac; i++)
            if (niz[i] == b) return true;
        //Sledeca komanda nije deo if ni for naredbe
        return false;
    }

    int zbiri () {
        int z = 0;
        for (int i=0; i<brojac; i++) z=z+niz[i];

        return z;
    }

    int maxElement () {
        int max = niz[0];
        for (int i=0; i<brojac; i++)
            if (niz[i] > max) max = niz[i];

        return max;
    }
}

class TestNizCelihBrojeva {

    public static void main(String[] args) {

        NizCelihBrojeva nc = new NizCelihBrojeva(4);

        nc.dodajUNiz(4);
        nc.dodajUNiz(19);

        nc.ispisi();

    }
}

```

Zadatak 2

(rade ga studenti sami)

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

- Atribut niz koji predstavlja niz realnih brojeva.
- Atribut brojac koji predstavlja trenutni broj elemenata u nizu. Početna vrednost je 0.
- Konstruktor u kome se niz kreira tako da mu maksimalni kapacitet bude jednak vrednosti koja se prosleđuje konstruktoru u obliku ulaznog argumenta.
- Metodu za dodavanje novog elementa (tj. broja) u niz; broj koji se dodaje prosleđuje se metodi u vidu ulaznog argumenta. Pre unosa, potrebno je proveriti da li u nizu još ima praznih mesta. Ako nema, ispisati poruku o grešci na ekranu.
- Metodu za računanje proizvoda brojeva u nizu; izračunatu vrednost ispisati na ekranu..

- Metodu za pronalaženje najmanjeg broja u nizu; pronađeni broj ispisati na ekranu.

Napraviti klasu **TestNizRealnihBrojeva** koja kreira jedan objekat klase **NizRealnihBrojeva** sa proizvoljnim kapacitetom. Potrebno je uneti nekoliko elemenata u niz, i ispisati na ekranu najmanji broj u nizu.

```
class NizRealnihBrojeva {

    double[] niz;
    int brojac = 0;

    NizRealnihBrojeva(int kapacitet){
        niz = new double[kapacitet];
    }

    void dodajUNiz(double b) {
        if (brojac < niz.length) {
            niz[brojac] = b;
            brojac++;
        }
        else System.out.println("Greska! Niz je pun!");
    }

    void ispisiProizvod () {
        double p = 1;
        for (int i=0;i<brojac;i++) p = p * niz[i];

        System.out.println(p);
    }

    void ispisiMinElement() {
        double min = niz[0];
        for (int i=0;i<brojac;i++)
            if (niz[i] < min) min = niz[i];

        System.out.println(min);
    }
}

class TestNizRealnihBrojeva {

    public static void main(String[] args) {

        NizRealnihBrojeva nr = new NizRealnihBrojeva(5);

        nr.dodajUNiz(12.22);
        nr.dodajUNiz(33.0);
        nr.dodajUNiz(1);

        nr.ispisiMinElement();
    }
}
```