

# KI103-DZ06-Branislav\_Manojlovic\_kp11

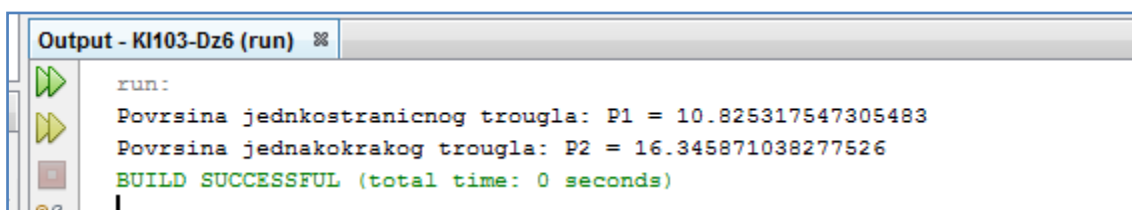
## Zadatak 1

Napraviti dve metode za izračunavanje površine jednakostraničnog i jednakokrakog trougla. Prikazati rad metoda preko Main-a.

### Rešenje:

```
package domacizadatak1;
/** @author Branislav */
public class Zadatak1 {
    //metoda koja izracunava povrsinu jednakostranicnog trougla
    public double povJednakostranicnog(double a){
        double P;
        P = (Math.pow(a, 2)*Math.sqrt(3))/4;
        return P;
    }
    //metoda koja izracunava povrsinu jednakokrakog trougla
    public double povJednakokrakog(double a, double b){
        double P;
        P = a/4*(Math.sqrt(4*Math.pow(b, 2)-Math.pow(a, 2)));
        return P;
    }
    public static void main(String[] args) {
        double a=5, b=7, P1, P2;
        Zadatak1 poziv = new Zadatak1();
        //pozivam funkciju "povJednakostranicnog" na izvršenje
        P1 = poziv.povJednakostranicnog(a);
        System.out.println("Povrsina jednakostranicnog trougla: P1 = " + P1);
        //pozivam funkciju "povJednakokrakog" na izvršenje
        P2 = poziv.povJednakokrakog(a, b);
        System.out.println("Povrsina jednakokrakog trougla: P2 = " + P2);
    }
}
```

Nakon pokretanja i izvršavanja programa, kao rešenje dobijam ispis na konzoli koji je prikazan na sledećoj slici. Test vrednosti su  $a = 5$  i  $b = 7$ .



Slika 1. Ispis rešenja u zadatku 1

## Zadatak 2

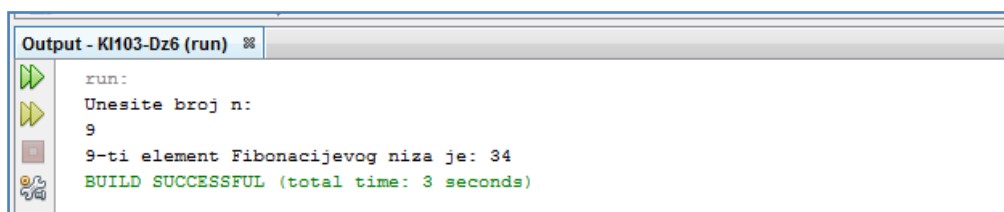
Napraviti metodu koja vraća N-ti element fibonačijevog niza. Svaki element fibonačijevog niza je jednak zbiru prethodna dva (Preskočiti 1 i 2). Koristiti petlju unutar metode. Prikazati rad metode preko Main-a.

### Rešenje:

```
package domacizadatak2;
import java.util.Scanner;
/** @author Branislav */
public class Zadatak2 {
    //metoda koja odredjuje N-ti element Fibonacijevog niza
    public int odredjivanjeElementa(int n){
        int f1=0, f2=1, k;
        //for petlja koja prolazi kroz Fibonacijev niz do zadatog elementa n i vraca poslednji
        for(int i=0; i<n; i++){
            k = f1;
            f1 = f2;
            f2 = k + f2;
        }
        return f1;
    }

    public static void main(String[] args) {
        int n, resenje;
        Scanner sc = new Scanner(System.in);
        System.out.println("Unesite broj n:");
        n = sc.nextInt();
        Zadatak2 poziv = new Zadatak2();
        //pozivam metodu na izvršenje
        resenje = poziv.odredjivanjeElementa(n);
        System.out.println(n + "-ti element Fibonacijevog niza je: " + resenje);
        sc.close();
    }
}
```

Nakon pokretanja programa i unosa vrednosti za broj n, dobijamo ispis kao na slici 2.



Slika 2. Ispis rešenja u zadatku 2

## Zadatak 3

Napravi metodu koja prima n brojeva. Svaki paran broj koji metoda primi treba da pomnoži sa 3 i dodati na zbir. Zbir vratiti. Prikazati rad metode preko Main-a.

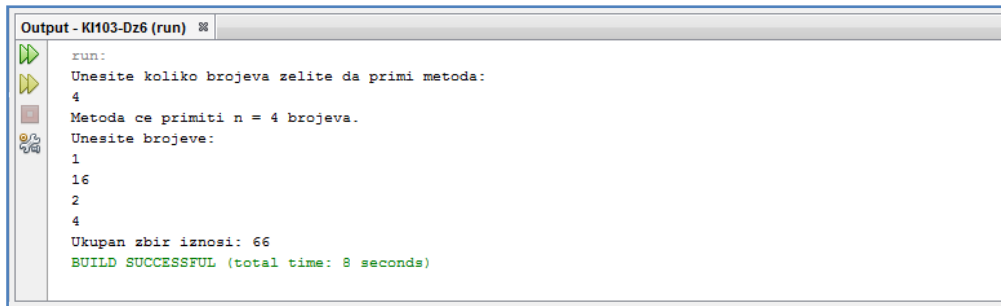
### Rešenje:

```
package domacizadatak3;
import java.util.Scanner;
/** @author Branislav */
public class Zadatak3 {
    //metoda koja vrsi sabiranje brojeva po zadatim parametrima
    public int sabiranje(Scanner sc,int n){
        int zbir=0, x;
        System.out.println("Unesite brojeve: ");
        for(int i=0; i<n; i++){
            x = sc.nextInt();
            if(x%2==0){
                zbir += x*3;
            }
        }
        return zbir;
    }

    public static void main(String[] args) {
        int n, x, UkupanZbir;
        Scanner sc = new Scanner(System.in);
        Zadatak3 poziv = new Zadatak3();
        System.out.println("Unesite koliko brojeva zelite da primi metoda:");
        n = sc.nextInt();
        System.out.println("Metoda ce primiti n = " + n + " brojeva.");
        //pozivam metodu za sabiranje i prosledjujem joj podatak o kolicini brojeva za sabiranje
        UkupanZbir = poziv.sabiranje(sc, n);
        System.out.println("Ukupan zbir iznosi: " + UkupanZbir);
    }
}
```

Nakon pokretanja programa, program traži od korisnika da unese podatak o tome koliko brojeva će biti obrađeno u programu. Posle toga se od korisnika traži da unese koji brojevi će biti obrađeni. Nakon izvršenja programa, kao rešenje dobija se ispis na konzoli kao na slici 3. Test podaci upotrebljeni u ovom zadatku su:

- Količina brojeva je: 4,
- Zadati brojevi su: 1, 16, 2, 4.



The screenshot shows an IDE output window titled "Output - K1103-Dz6 (run)". On the left, there is a vertical toolbar with icons for running (green play button), stepping through (yellow play button), and debugging (red bug icon). The output text is as follows:

```
run:
Unesite koliko brojeva zelite da primi metoda:
4
Metoda ce primiti n = 4 brojeva.
Unesite brojeve:
1
16
2
4
Ukupan zbir iznosi: 66
BUILD SUCCESSFUL (total time: 8 seconds)
```

Slika 3. Ispis rešenja u zadatku 3

## Zadatak 5

Napraviti metodu koja štampa kvadrat karakterom (\*) veličine prosleđenog parametra (int). Primer ako je unos 3:

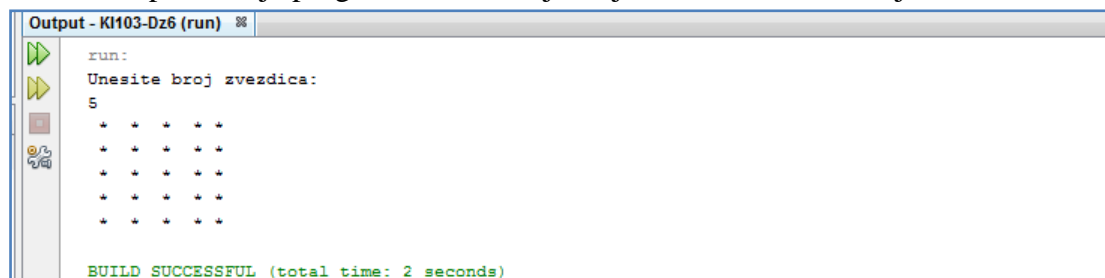
```
***  
***  
***
```

Prikazati rad metode preko Main-a.

### Rešenje:

```
package domacizadatak5;  
import java.util.Scanner;  
/** @author Branislav */  
public class Zadatak5 {  
    //metoda koja ispisuje zvezdice po zadatom broju n  
    public void ispisZvezdica(int n){  
        //spoljasnja for petlja koja ispisuje zvezdice po vertikali  
        for(int i=1; i<=n; i++){  
            //ugnjezdena for petlja koja ispisuje zvezdice po horizontali  
            for(int j=1; j<=n; j++){  
                System.out.print(" * ");  
            }  
            System.out.println("");  
        }  
        System.out.println(" ");  
    }  
    public static void main(String[] args) {  
        int n;  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Unesite broj zvezdica:");  
        n = sc.nextInt();  
        Zadatak5 poziv = new Zadatak5();  
        //pozivam metodu na izvršenje  
        poziv.ispisZvezdica(n);  
    }  
}
```

Nakon pokretanja programa i zadavanja da je br. zvezdica 5, dobijam sledeće rešenje.



Slika 4. Ispis rešenja u zadatku 5

## Zadatak 6

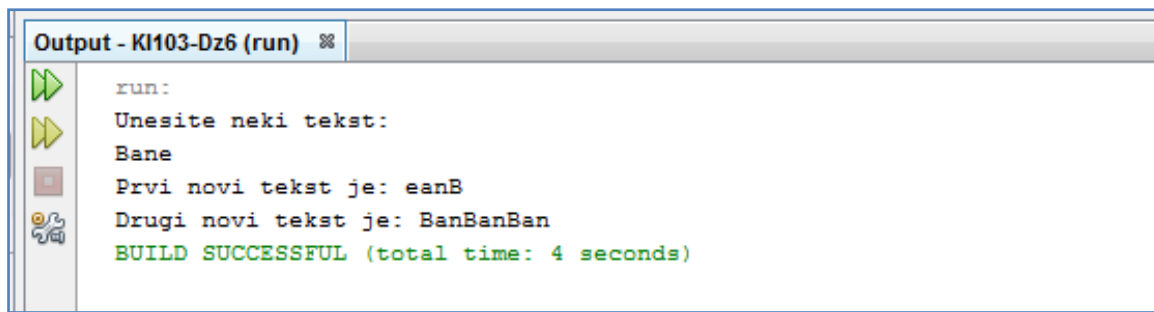
Napraviti metodu koja menja prvi i zadnji karakter String-a i metodu koja uzima prva tri karaktera String i potom vraća 3 puta ta tri karaktera. Primer: Java Rezultat: JavJavJav. Prikazati rad metode preko Main-a.

### Rešenje:

```
package domacizadat6;
import java.util.Scanner;
/** @author Branislav */
public class Zadatak6 {
    //metoda koja vrsi zamenu prvog i poslednjeg karaktera
    public void zamenaKaraktera(String tekst){
        String noviTekst;
        //uzimanje karaktera sa prvog i poslednjeg mesta u stringu i njihova zamena mesta
        noviTekst = tekst.charAt(tekst.length() - 1) + tekst.substring(1, tekst.length() - 1) +
        tekst.charAt(0);
        System.out.println("Prvi novi tekst je: " + noviTekst);
    }
    //metoda koja uzima prva tri karaktera i spaja ih u nov string
    public void uzimanjePrvaTriKarakta(String tekst){
        String noviTekst2;
        //spajanje izvucenog podstringa tri puta
        noviTekst2 = tekst.substring(0, 3) + tekst.substring(0, 3) + tekst.substring(0, 3);
        System.out.println("Drugi novi tekst je: " + noviTekst2);
    }

    public static void main(String[] args) {
        String tekst;
        Scanner sc = new Scanner(System.in);
        System.out.println("Unesite neki tekst:");
        tekst = sc.nextLine();
        Zadatak6 poziv = new Zadatak6();
        //pozivam prvu metodu
        poziv.zamenaKaraktera(tekst);
        //pozivam drugu metodu
        poziv.uzimanjePrvaTriKarakta(tekst);
    }
}
```

Nakon pokretanja programa od korisnika se zahteva da unese neki tekst – string. Nakon izvršenja programa, dobija se ispis prikazan na slici 5. Kao test podatak za string je uzet tekst: Bane.



```
run:
Unesite neki tekst:
Bane
Prvi novi tekst je: eanB
Drugi novi tekst je: BanBanBan
BUILD SUCCESSFUL (total time: 4 seconds)
```

Slika 5. Ispis rešenja zadatka 6

**Napomena:** Pored ovog tekstualnog fajla, prilažem i kompletan projekat u zipovanom fajlu: KI103-Dz6.zip.

MSc Branislav Manojlović