

Početna grupa, 28.11.2015. – Metodi

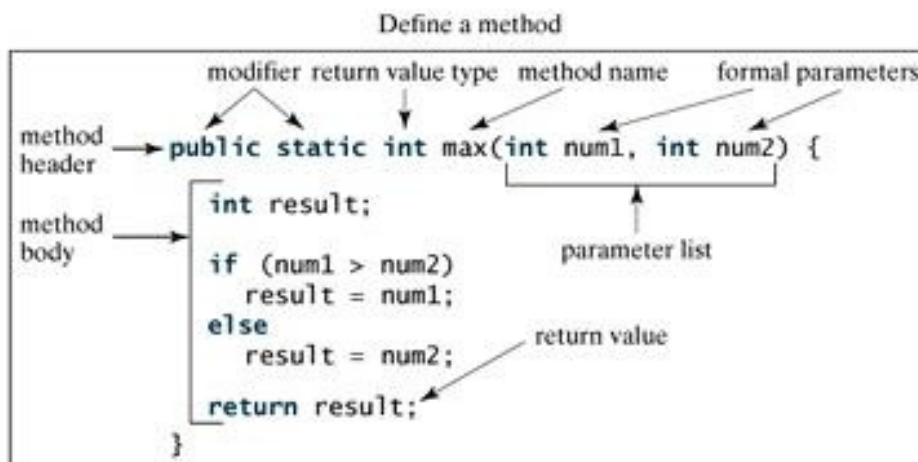
1. Metodi – opšti oblik metoda:

```
modifier returnType methodName(list of parameters)
{
    // Method body;
}
```

2. Ime metoda: početno slovo je malo, a zatim slijede slova, cifre ili simbol '_' . Primjeri dobrih imena:

```
run
runFast
getBackground
getFinalData
compareTo
setX
isEmpty
```

3. Primjer:



4. Ako želite da metod pozivate direktno, a ne preko promjenljive tipa klase, tada ga morate deklarirati kao `public static`.

5. Ako direktno pozivate metod `max`, tada klasa ima sljedeći izgled:

```
public class TestMax {
    /** Main method */
    public static void main(String[] args) {
        int i = 5;
        int j = 2;
        int k = max(i, j);
        System.out.println("The maximum between " + i +
            " and " + j + " is " + k);
    }

    /** Return the max between two numbers */
    public static int max(int num1, int num2) {
        int result;
        if (num1 > num2)
            result = num1;
    }
}
```

```

        else
            result = num2;

        return result;
    }
}

```

6. Ako direktno pozivate metod `printGrade`, tada klasa ima sljedeći izgled:

```

public class TestVoidMethod {
    public static void main(String[] args) {
        printGrade(78.5);
    }

    public static void printGrade(double score) {
        if (score >= 90.0) {
            System.out.println('A');
        }
        else if (score >= 80.0) {
            System.out.println('B');
        }
        else if (score >= 70.0) {
            System.out.println('C');
        }
        else if (score >= 60.0) {
            System.out.println('D');
        }
        else {
            System.out.println('F');
        }
    }
}

```

7. Metod koji je označen kao `static` može direktno pozivati samo druge `static` metode.

8. Primjeri metoda dati su u fajlu `Metodi2012.java`.

```

public class Metodi2012 {

    /**
     * @param args
     */
    // Za dato x, racuna x*x+2x+5
    // public i static su modifikatori
    // double je tip rezultata - return type
    // ime metoda> kvadrat
    // lista parametara: samo jedna element u listi i to je double x

    public static double kvadrat(double x)
    {
        double y; // lokalna promjenljiva, koristi se samo u metodu
        y = x*x+2*x+5;
        return y; // vracamo vrijednost onome ko je pozvao metod kvadrat
        (obicno je to neki drugi metod, npr. main)
        // return x*x+2*x+5; // moze i ovako, kao jedna naredba
    }

    // Izracunava i vraca zbir brojeva x i y
}

```

```

// Return type> double
// Ime metoda: zbir
// Lista parametara> 2 parametra x i y, oba tipa double
public static double zbir(double x, double y)
{
    double r;
    r = x+y;
    return r;
}

// Stampati kvadrate svih cijelih brojeva od 1 do n
// Return type: void (jer metod ne vraca vrijednost)
public static void stampaDoN(int n)
{
    int i;
    for (i = 1; i<=n; i++)
    {
        System.out.println(i*i);
    }
    return; // moze i bez ove naredbe
}

// Stampati sve brojeve iz segmenta [a,b] koji su djeljivi sa tri
// Return type: void (jer metod ne vraca vrijednost)
public static void stampaDjeljivihSa3(int a, int b)
{
    int i;
    for(i = a; i<=b; i++)
    {
        if (i%3 == 0)
        {
            System.out.println(i);
        }
    }
}

// Izracunava zbir kvadrata brojeva od 1 do n
public static int zbirKvadrata(int n)
{
    int zbir = 0, i;
    for(i=1; i<=n; i++ )
    {
        zbir += i*i;
    }
    return zbir;
}

// Izracunava zbir cifara cijelog broja n
public static int zbirCifara(int n)
{
    int m = Math.abs(n), zbir = 0;;
    while (m != 0)
    {
        zbir += m%10;
        m /= 10;
    }
    return zbir;
}

```

```

public static void main(String[] args) {
    double a = 5, rezultat = 0;
    // Pozivamo metod kvadrat vise puta
    rezultat = kvadrat(a); // vrijednost promjenljive a se kopira u
memoriju argumenta (parametra) x u metodu kvadrat
    System.out.println(rezultat);
    rezultat = kvadrat(a+5); // vrijednost izraza a+5 se kopira u memoriju
argumenta (parametra) x u metodu kvadrat
    System.out.println(rezultat);
    rezultat = kvadrat(4); // vrijednost 4 se kopira u memoriju argumenta
(parametra) x u metodu kvadrat
    System.out.println(rezultat);

    // Pozivamo metod zbir vise puta
    rezultat = zbir(10,5);
    System.out.println(rezultat);
    rezultat = zbir(a,a+5);
    System.out.println(rezultat);
    rezultat = zbir(a,-13);
    System.out.println(rezultat);

    System.out.println(zbir(4,8)); // direktno stampamo rezultat
izvršavanja metoda zbir

    // Pozivamo metod zbirKvadrata vise puta
    System.out.println(zbirKvadrata(3));
    System.out.println(zbirKvadrata(4));
    System.out.println(zbirKvadrata(5));

    // Pozivamo metod stampaDoN
    // Obratite paznju da se poziva drugacije od prethodnih metoda, jer je
return type ovog metoda void
    stampaDoN(12);

    // Pozivamo metod zbirCifara
    System.out.println("Zbir cifara je " + zbirCifara(-4531));
}
}

```

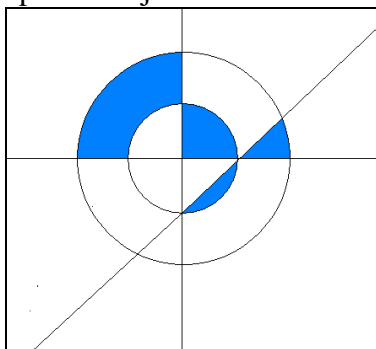
Zadaci za vježbu (metodi)

1. Napisati metod `double promjena(double x, double a)` koji vraća broj $a \cdot x$, ako je x nenegativan i a/x , ako je x negativan.
2. Napisati metod `int zbirIzIntervala(int a, int b)` koji vraća zbir svih cijelih brojeva iz intervala $[a, b]$
3. Napisati metod `double duzinaDuzi(double x1, double y1, double x2, double y2)` koji vraća dužinu duži čije su krajnje tačke $A(x1, y1)$ i $B(x2, y2)$.
4. Napisati metod `boolean pripadaIntervalu(int a, int b, int x)` koji vraća `true` ako broj x pripada intervalu $[a, b]$ i vraća `false` ako ne pripada.
5. Napisati metod `void spisakDjelilaca(int n)` koji štampa sve pozitivne djelioce broja n .
6. Napisati metod `int brojDjelilaca(int n)` koji vraća broj pozitivnih djelilaca broja n .

7. Napisati metod `int zbirDjelilaca(int n)` koji vraća zbir pozitivnih djelilaca broja n manjih od n .
8. Napisati metod `int zbirCifara(int n)` koji vraća zbir cifara broja n .
9. Napisati metod `int najvecaCifra(int n)` koji vraća najveću cifru broja n .
10. Napisati metod `int minMaxCifra(int n)` koji vraća zbir najveće i najmanje cifre broja n .
11. Napisati metod `double stepen(double x, int n)` koji vraća x^n . Ne koristiti metod `pow` iz klase `Math`.
12. Napisati metod `void print3K(int a, int b)` koji štampa sve cijele brojeve iz intervala $[a,b]$ koji su oblika $3k+1$, za neko cjelobrojno k .
13. Napisati metod `long fakt(long n)` koji vraća $n!$ ($n! = 1*2*...*n$).
14. Napisati metod `boolean stepenDvojke(int n)` koji vraća `true` ako je n stepen broja 2 i `false` ako n nije stepen broja 2.
15. Napisati metod `int minStepenDvojke(int n)` koji vraća najmanji prirodan broj k takav da n nije veći od broja 2^k .
16. Napisati metod `boolean trougaoPostoji(double x1, double y1, double x2, double y2, double x3, double y3)` koji vraća `true` ako postoji trougao sa tjemena $(x1,y1)$, $(x2,y2)$, $(x3,y3)$ i `false` ako trougao ne postoji.
17. Napisati metod `double površinaTrougla(double x1, double y1, double x2, double y2, double x3, double y3)` koji vraća površinu trougla sa tjemena $(x1,y1)$, $(x2,y2)$, $(x3,y3)$, ako trougao postoji i 0 ako trougao ne postoji.
18. Napisati metod `double redEX(double x, double eps)` koji vraća zbir reda $\sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \dots$. Sumiranje se prekida kada tekući sabirak po apsolutnoj vrijednosti postane veći od vrijednost `eps`.
19. Napisati metod `double red(double x, double eps)` koji vraća zbir reda $\sum_{n=0}^{\infty} (-1)^n (n+1)x^{2n}$, $-1 < x < 1$. Sumiranje se prekida kada tekući sabirak po apsolutnoj vrijednosti bude veći od vrijednost `eps`.
20. Napisati metod `double prosjekOcjena(int n)` koji provjerava da li je broj n pozitivan i ako jeste, učitava n cijelih brojeva koje predstavljaju ocjene iz matematike za n učenika vašeg odjeljenja i izračunava prosječnu ocjenu za tih n učenika. Ako n nije pozitivan, metod vraća 0.
21. Napisati metod `void manjiOdX(int x)` koji učitava cijele brojeve, sve dok se ne učitao broj koji je veći od x , a zatim štampa broj učitanih brojeva, broj učitanih parnih brojeva i zbir svih učitanih brojeva.
22. Napisati metod `int nadjiZbir(int n)` koji sa standardnog ulaza (pomoću klase `Scanner`) učitava n cijelih brojeva i vraća njihov zbir.
23. Napisati metod `int nadjiNajveci(int n)` koji sa standardnog ulaza (pomoću klase `Scanner`) učitava n cijelih brojeva i vraća najveći od učitanih brojeva.
24. Napisati metod `int najveciNeparniDjelilac(int n)` koji vraća najveći neparni pozitivni djelilac broja n .
25. Napisati metod `int nadjiNajveciTrougao(int n)` koji sa standardnog ulaza (pomoću klase `Scanner`) učitava n trojki pozitivnih cijelih brojeva (a, b, c) i vraća površinu najvećeg

trougla sa stranicama a , b i c . Napomena: Napišite metod koji računa površinu trougla i koristite ga u metodi `nadjiNajveciTrougao`.

26. Napisati metod `boolean jeProst(int n)` koji za dati broj n provjerava da li je prost, i ako jeste, vraća `true`, a ako nije, vraća `false`.
27. Napisati metod `int prostInterval(int a, int b)` koji štampa sve proste brojeve iz intervala $[a, b]$, $a \leq b$.
28. Napisati metod `void prostiDjelioци(int n)` koji za dati broj n štampa sve proste djelioce broja n .
29. Napisati metod `int gcd(int n, int m)` koji vraća najveći zajednički djelilac brojeva m i n . (Pogledati na internetu Euklidov algoritam).
30. Prirodan broj n je Armstrongov ako je jednak zbiru kubova svojih cifara. Npr. 371 je Armstrongov, jer je $3^3 + 7^3 + 1^3 = 371$. Napisati metod `boolean isArmstrong(int n)` koji za dati broj n provjerava da li je Armstrongov, i ako jeste, vraća `true`, a ako nije, vraća `false`.
31. Prirodan broj n je Hemingov ako svi njegovi prosti djelioци pripadaju skupu $\{2, 3, 5\}$. Prvi Hemingov broj je 2, pa zatim idu 3, $4 = 2 * 2$, 5, $6 = 2 * 3$, $8 = 2 * 2 * 2$, $9 = 3 * 3$, $10 = 2 * 5$, $12 = 2 * 2 * 3$, $15 = 3 * 5$, itd. Npr. 14 nije Hemingov broj jer je $14 = 2 * 7$, pa prosti djelioци broja 14 su 2 i 7, a 7 ne pripada skupu $\{2, 3, 5\}$. Napisati metod `boolean isHemming(int n)` koji za dati broj n provjerava da li je Hemingov, i ako jeste, vraća `true`, a ako nije, vraća `false`.
32. Napisati metod `int kiselina(int h, int s, int o)` koji za 3 data pozitivna cijela broja h , s i o koji redom predstavljaju broj molekula vodonika (H), sumpora (S) i kiseonika (O), vraća koliko se najviše molekula sumporne kiseline (H_2SO_4) može dobiti od datih molekula. Npr., ako je $h=4$, $o=3$ i $s=10$, odgovor je 2.
33. Napisati metod `double presjekIntervala(double a, double b, double c, double d)` koji vraća dužinu presjeka intervala $[a, b]$ i $[c, d]$, $a \leq b$, $c \leq d$.
34. Napisati metod `double unijaIntervala(double a, double b, double c, double d)` koji vraća dužinu najmanjeg intervala koji sadrži i $[a, b]$ i $[c, d]$, $a \leq b$, $c \leq d$.
35. Napisati metod `boolean unutra(double x, double y)` koji za date realne brojeve x i y provjerava da li tačka sa koordinatama (x, y) pripada osjenčenom dijelu ravni. Centar oba kruga je u tački $(0, 0)$, poluprečnici su im redom 4 i 6, dok je prava data jednačinom $x - y - 4 = 0$. Podsjetite se da je krug skup tačaka u ravni koje su na rastojanju r od date tačke tj. centra kruga. Metod vraća `true` ako tačka pripada datoj oblasti i `false` ako ne pripada.



36. Napisati metod `boolean isInsideTriangle(double a1, double a2, double b1, double b2, double c1, double c2, double x, double y)`, koji provjerava da li postoji trougao čija su tjemena $A(a_1, a_2)$, $B(b_1, b_2)$ i $C(c_1, c_2)$ i ako postoji vraća `true` ako tačka (x, y) pripada unutrašnjosti trougla, uključujući i stranice trougla.