

Java funkcije

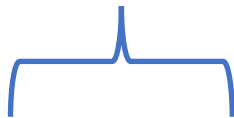
Pojam funkcije

- Funkcija (metoda) je grupa naredbi koje zajedno izvršavaju neki zadatak
- Program se može podeliti u više funkcija pri čemu svaka funkcija izvršava specifičan zadatak
- Funkcije su mehanizam za razbijanje složenih problema na jednostavnije delove
- Svaki Java program ima glavnu- main() funkciju i iz nje se pozivaju pomoćne funkcije
- Funkcije se definišu kao potprogrami koji na osnovu izvesnog broja argumenata (ulaznih parametara) daju jedan rezultat koji se naziva vrednost funkcije
- Funkcije pored vrednosti funkcije mogu dati i druge rezultate koji se nazivaju bočnim efektima
- Vrlo često se ne koristi vrednost funkcije već samo njeni bočni efekti

Definicija funkcije

tip povratne vrednosti ili void

formalni parametri funkcije



```
public static tipRezultata imeFunkcije( tip1 param1, tip2 param2, ..., tipN paramN)  
{  
  naredba1;  
  ....  
  naredban;  
}
```



telo funkcije

} zaglavlje (potpis) funkcije

Definicija funkcije

- Funkcije se dele na dve kategorije:
 - funkcije koje vraćaju vrednost pozivajućoj funkciji (main ili neka druga)
 - funkcije koje ne vraćaju vrednost pozivajućoj funkciji (void funkcije)

Delovi funkcije

- Kada funkcija ne vraća vrednost kao tip rezultata se navodi **void**, u protivnom se definiše tip podataka koga vraća funkcija
- imeFunkcije je identifikator funkcije
- Niz parametara predstavlja niz argumenat pomoću kojih se vrši unošenje podataka u funkciju na osnovu kojih se izračunava vrednost funkcije
- Telo funkcije predstavlja sadržaj funkcije koja se definiše
- U telu funkcije se pored ostalih naredbi može naći i naredba povratka u pozivajuću funkciju return

Parametri funkcije i lokalne promenljive

- Parametri koji su navedeni u zaglavlju funkcije nazivaju se formalni parametri funkcije
- Parametri koji se navode pri svakom obraćanju funkciji nazivaju se stvarni parametri funkcije
- Promenljive koje se definišu unutar tela funkcije nazivaju se lokalne promenljive
- Lokalne promenljive su “vidljive” samo unutar funkcije koja ih sadrži

Naredba return

- Ako se navede samo **return**; funkcija u kojoj se nalazi ova naredba prekida izvršenje i prelazi se na izvršavanje sledeće naredbe pozivajuće funkcije
- Ako se navede **return vrednost** ; tada funkcija vraća vrednost pozivajućoj funkciji
- Naredba **return vrednost**; se ne može napisati unutar void funkcije
- Ako se return izraz ne navede unutar tela funkcije tada se izvršavanje funkcije završava izvršavanjem poslednje naredbe te funkcije

Funkcija koja vraća vrednost

```
public static double saberi(double a, double b) {  
    System.out.println("-----");  
    System.out.println("Telo funkcije saberi");  
    double zbir = a + b;  
    System.out.println("-----");  
    return zbir; // vrati vrednost pozivajućoj funkciji  
}
```

```
public_static_method snippet
```


Funkcija koja ne vraća vrednost

```
public static void stampaj(double a) {  
    System.out.println(".....");  
    System.out.println("Telo funkcije Stampaj");  
    System.out.println(a);  
    System.out.println(".....");  
}
```

Poziv funkcije

```
public static void main(String[] args) {  
    System.out.println("////////////////////////////////////  
////////////////////////////////////");  
    System.out.println("Pocetak Main funkcije");  
    // poziv funkcije Saberi()  
    // promenljiva zbir prihvata povratnu vrednost  
    // 5 i 6 su stvarni parametri  
    double zbir = saberi(5, 6);  
    System.out.println("Povratak u Main funkciju");  
    //poziv void funkcije  
    stampaj(zbir);  
    System.out.println("Povratak u Main funkciju");  
    System.out.println("Kraj Main funkcije");  
    System.out.println("////////////////////////////////////  
////////////////////////////////////");  
}
```

Izvršavanje programa pozivom funkcija

```
////////////////////////////////////  
Pocetak Main funkcije  
-----  
Telo funkcije saberi  
-----  
Povratak u Main funkciju  
.....  
Telo funkcije Stampaj  
11.0  
.....  
Povratak u Main funkciju  
Kraj Main funkcije  
////////////////////////////////////
```

Funkcija Maksimum()

```
public static int maksimum(int a, int b) {  
    int m = a > b ? a : b;  
    return m;  
}
```

```
public static void main(String[] args) {  
    // 5, 7  
    int m1 = maksimum(5, 7);  
    System.out.println(m1);  
}
```

Maksimalna vrednost 3 broja

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Unesi prvi ceo broj: ");
    int a = sc.nextInt();

    System.out.println("Unesi drugi ceo broj: ");
    int b = sc.nextInt();

    System.out.println("Unesi treci ceo broj: ");
    int c = sc.nextInt();
    sc.close();

    int m1 = maksimum(a, b);
    int m2 = maksimum(m1, c);

    System.out.format("max(%d,%d,%d)=%d\n", a,b,c,m2);
}
```

Funkcije klase Math

```
public static int citaj(Scanner sc) {  
  
    System.out.println("Unesi ceo broj: ");  
    int a = sc.nextInt();  
    return a;  
}
```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    int a = citaj(sc);  
    int b = citaj(sc);  
    int c = citaj(sc);  
    sc.close();  
    int m1 = Math.max(a, b);  
    int m2 = Math.max(m1, c);  
    System.out.format("max(%d,%d,%d)=%d\n", a,b,c,m2);  
}
```

Kvadratni koren

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Unesi realan broj");  
    double a = sc.nextDouble();  
    double b = Math.sqrt(a);  
    sc.close();  
    System.out.println("Koren iz broja: " + a + " je broj: " + b);  
}
```

Stepen realnog broja

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Unesite realni broj a: ");
    double a = sc.nextDouble();

    System.out.println("Unesite stepen n: ");
    int n = sc.nextInt();
    sc.close();

    double s = Math.pow(a, n);

    System.out.println("Rezultat stepenovanja je: " + s);
}
```


Primer 1

Napisati funkciju kojoj se prosleđuje ceo broj n a ona vraća zbir svih brojeva od 1 do broja n . Pozvati ovu funkciju.

Primer 1 -rešenje

```
public static int suma(int n){
    int s =0;
    int i = 1;
    while (i<=n) {
        s+=i;
        i++;
    }
    return s;
}
```

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Unesi ceo broj n: ");
    int n = sc.nextInt();
    sc.close();

    int s = suma(n);
    System.out.println("Zbir 1+....+ " + n + "=" + s);
}
```

Primer 2

Napisati funkciju kojom se štampaju svi stepeni broja 2 koji nisu veći od unapred zadate granice koja je veća od broja 2 i koja se prosleđuje kao argument funkciji.

Primer 2 – rešenje

```
public static void stampajStepeneDvojke(int granica) {  
    if (granica < 2) {  
        System.out.println("Greska: Granica mora biti veca od 2.");  
        return;  
    }  
  
    int broj = 2;  
    while (broj <= granica) {  
        System.out.println(broj);  
        broj *= 2;  
    }  
}
```

Primer 2 – rešenje

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Unesi granicu: ");  
    int granica = sc.nextInt();  
    sc.close();  
    stampajStepeneDvojke(granica);  
}
```

Primer 3

Funkciji se prosleđuje trocifreni broj. Funkcija štampa cifru stotina, cifru desetica i cifru jedinica.

Unesi trocifreni broj:

768

Cifra stotina: 7

Cifra desetica: 6

Cifra jedinica: 8

Primer 3 –rešenje

```
public static void stampajCifre (int broj) {  
    if (broj <100 || broj > 999) {  
        System.out.println("Niste uneli trocifreni broj");  
        return;  
    }  
    int s = broj /100;  
    int d = (broj %100)/10;  
    int j = broj %10;  
  
    System.out.println("Cifra stotina: " + s);  
    System.out.println("Cifra desetica: " + d);  
    System.out.println("Cifra jedinica: " + j);  
}
```

Primer 3 –rešenje

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Unesi trocifreni broj: ");  
    int broj = sc.nextInt();  
    sc.close();  
    stampajCifre(broj);  
}
```


Primer 4 - Linija

```
public static void Linija(char karakter, int n)
{
    for (int i = 0; i < n; i++) {
        System.out.print(karakter);
    }
    System.out.println();
}

public static void main(String[] args) {
    Linija('.', 50);
    Linija('/', 50);
    Linija('\\', 50);
}
```

Primer 5

Java funkcija pronalazi trocifrene brojeve koji su jednaki zbiru kubova svojih cifara

```
public static void pronadjiTrocifreneBrojeve() {
    int cifraStotina, cifraDesetica, cifraJedinica, zbirKubova;
    for (int broj = 100; broj < 1000; broj++) {
        cifraStotina = broj / 100;
        cifraDesetica = (broj / 10) % 10;
        cifraJedinica = broj % 10;

        zbirKubova = (int)Math.pow(cifraStotina, 3)
            + (int)Math.pow(cifraDesetica, 3)
            + (int)Math.pow(cifraJedinica, 3);

        if (broj == zbirKubova) {
            System.out.println(broj);
        }
    }
}

public static void main(String[] args) {
    pronadjiTrocifreneBrojeve();
}
```

Prosleđivanje parametara po vrednosti

```
public static void razmeni(int a, int b) {  
    int temp = a;  
    a = b;  
    b = temp;  
    System.out.format("U funkciji: a=%d b=%d%n",a,b);  
}  
public static void main(String[] args) {  
    int x = 5;  
    int y = 10;  
    System.out.println("Pre zamene: x = " + x + ", y = " + y);  
  
    razmeni(x, y);  
    System.out.println("Posle zamene: x = " + x + ", y = " + y);  
}
```

U Javi se primitivni tipovi (kao što su int, float, boolean, itd.) ne mogu direktno proslediti po referenci!

Rekurzivne funkcije

- Rekurzivna funkcija je funkcija koja poziva sama sebe

$$a^0 = 1$$

$$a^n = a^{n-1} \cdot a$$

$$f(n) = a^n$$

$$f(n) = f(n - 1) \cdot a$$

Rekurzivna funkcija koja pronalazi n-ti stepen celog broj a

```
public static int power(int a, int n) {  
    if (n == 0) {  
        return 1;  
    }  
    return a * power(a, n - 1);  
}
```

Poziv rekurzivne funkcije

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Unesite osnovu a: ");
    int a = sc.nextInt();
    System.out.print("Unesite stepen n: ");
    int n = sc.nextInt();
    sc.close();
    int rezultat = power(a, n);
    System.out.println(a + " na " + n + " je " + rezultat);
}
```

Napisati java funkciju za pronalaženje n-tog člana fibonacijevog niza: 1,1,2,3,5,8,...

$$f(1) = 1, f(2) = 1$$

$$f(3) = f(2) + f(1) = 2$$

$$f(4) = f(3) + f(2) = 2 + 1 = 3$$

$$f(n) = f(n-1) + f(n-2), \text{ za } n > 2$$


```
public static int fibonacci(int n) {
    if (n <= 2) {
        return 1;
    } else {
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Unesite poziciju n: ");
    int n = sc.nextInt();
    sc.close();
    int Fn = fibonacci(n);
    System.out.println("Fibonaccijev broj na poziciji " + n + " je: " + Fn);
}
```