

Metode (Funkcije)

Funkcija (metoda) je izdvojeni skup programskog koda koji se može pozvati (izvršiti) u bilo kom trenutku u programu. Dekompozicijom programskog koda u manje izdvojene celine (funkcije) eliminiše se potreba za njegovim ponavljanjem što doprinosi da sam program bude organizovaniji i čitljiviji. Funkcije se mogu najlakše objasniti ako se posmatraju kao podprogrami (podalgoritmi) specifične namene. Funkcija je rutina koja izračunava neku vrednost na osnovu date informacije.

Definicija funkcije:

```
static povratni_tip ime_funkcije(parametri) {  
    programski kod  
}
```

***Ključna reč static je obavezna za definisanje statičkih funkcija**

Za svaku funkciju koju definišemo moramo navesti njeno ime, skup ulaznih parametara i tip povratne vrednosti - rezultata funkcije. Preporučljivo je (**ali ne i obavezno**) da ime funkcije počne malim slovom. Broj i tip ulaznih parametara je proizvoljan, što implicira da je moguće kreirati funkcije koje nemaju ulazne parametre (samo otvorena i zatvorene zagrade se navode). Svaki od ulaznih parametara definiše se preko njegovog tipa i imena. Svaka funkcija vraća najviše jednu povratnu vrednost čiji tip mora da se poklopi sa navedenim *povratni_tip*-om u definiciji funkcije.

Ulazni parametri mogu biti **primitivni tip** ili **referenca na objekat (klasa, niz, mapa...)**.

Rezultat može biti **primitivni tip**, **referenca na objekat (klasa, niz, mapa...)** ili **bez povratne vrednosti - void**.

NAPOMENA Funkcija mora biti definisana (telo funkcije - njena realizacija) negde u klasi (okviru) ali se ne sme nalaziti u main metodi. Tamo se samo pozivaju metode.

Primer funkcije koja ispisuje tekst "Hello World" (funkcija nema ulazne parametre i nema povratnu vrednost).

```
static void pozdrav(){  
    System.out.println("Hello World");  
}
```

Ukoliko funkcija vraća povratnu vrednost, u telu funkcije **mora** postojati naredba **return** iza koje sledi povratna vrednost. Funkcije se mogu pozivati (izvršiti) u bilo kom trenutku u programu, potrebno je samo da se navede ime funkcije i njeni parametri (ukoliko postoje).

Primer funkcije koja na osnovu unetih vrednosti kateta izračunava hipotenuzu pravouglog trugla (funkcija sa dva ulazna parametra i povratnom vrednošću tipa double).

```
static double vrednostHipotenuzePravouglogTrougla(double a, double b){
```

```

    double c = 0;
    c = Math.sqrt(a*a + b*b);
    return c;
}

```

Ukoliko se tip povratne vrednost iz tela funkcije ne poklopi sa tipom povratne vrednosti navedenim u naslovu funkcije ili ako implicitna konverzija tipova nije moguća, alat Eclipse će prijaviti **grešku**.

```

static int vrednostHipotenuzePravouglogTrougla(double a, double b){
    double c = 0;
    c = Math.sqrt(a*a + b*b);
    return c; //GRESKA
}

```

Takođe, do greške će doći i ako se zaboravi naredba **return**.

Sve vrednosti parametara koje se navode u trenutku poziva funkcije prenose se po vrednosti u samu funkciju. Java na stek kopira vrednosti parametara iz poziva funkcije kao vrednosti ulaznih parametara. Nemoguće je promeniti prosleđenu promenljivu iz poziva funkcije.

```

static int vrednostHipotenuzePravouglogTrougla(double a, double b){           //GRESKA
    double c = 0;
    c = Math.sqrt(a*a + b*b);
}

```

Java dozvoljava definisanje više funkcija (method overloading) sa istim imenom ako i samo ako:

- a) razlikuju se po broju i/ili tipu ulaznih parametara

U telu jedne funkcije moguće je izvršiti poziv druge funkcije.