

NAPOMENA: Vežbe podrazumevaju da je student ovladao teorijom iz dela “Uvod u programski jezik Java” (11_uvod-java.ppt), “Nizovi i Stringovi” (12_nizovi-i-stringovi.ppt), “Metode” (21_metode.ppt) i “Algoritmi” (23_Odabrani algoritmi.ppt).

Stringovi

Primer 1. Rad sa promenljivom tipa String. Definisanje i korišćenje funkcija (dužina karaktera i karakter na poziciji).

```
//definisanje promenljive tipa string
String a = "Ovo je tekst ";
System.out.println(a);

//konkatencija stringova
//ukoliko promenljiva nije string poziva se njena konverzija
String b = a + 5 + " zadatka";
System.out.println(b);

//prikaz broja karaktera Stringa
System.out.println("Broj elemenata Stringa a je:" + a.length());

//pristup odredjenom karakteru Stringa, indeks prvog karaktera je 0
System.out.println("Prvi karakter: " + a.charAt(0));
System.out.println("Drugi karakter: " + a.charAt(1));
//greska pristup nepostojecem elementu, prepravi a.length()-1
System.out.println("Poslednji elemenata Stringa a je:" + a.charAt(a.length()-1));
```

Primer 2. Rad sa promenljivom tipa String. Trimovanje (uklanjanje razmaka) i pribavljanje delova stringa.

```
//definisanje promenljive tipa string
String a = " Ovo je tekst 6 zadatka ";
System.out.println(a);

//Odsecanje praznih mesta sa pocetka
//i kraja stringa, whitespace karaktera
a = a.trim();
System.out.println(a);

//kreiranje podstringa
String podstring;
//podstring obuhvata karaktere od 4 pozicije
//indeksa pa do kraja stringa
podstring = a.substring(4);
System.out.println("Podstring 1: " + podstring);

//podstring obuhvata karaktere od 4 pa do 9 pozicije
//indeksa (ne ukljucuje 9 karakter) "je te"
podstring = a.substring(4, 9);
System.out.println("Podstring 2: " + podstring);
```

Objektno orijentisano programiranje 1

Stringovi i funkcije

Primer 3. Rad sa promenljivom tipa String. Korišćenje funkcija (provera sličnosti stringa).

```
//Definisanje promenljive tipa string
String a    = "Ovo Je Tekst";
String a1   = "ovo je tekst";

//Provera da li su stringovi sadrze isti niz karaktera
if(a.equals(a1))
    System.out.println("1. Nizovi a i a1 su slicni");
else
    System.out.println("1. Nizovi a i a1 nisu slicni");

if(a.equalsIgnoreCase(a1))
    System.out.println("2. Nizovi a i a1 su slicni");
else
    System.out.println("2. Nizovi a i a1 nisu slicni");

//prebacivanje velikih u mala slova
a = a.toLowerCase();
System.out.println(a);
a1 = a1.substring(0,5);
//Provera da li string a zapocinje sa karakterima iz stringa a1
if(a.startsWith(a1))
    System.out.println("Zapocinje");
else
    System.out.println("Ne zapocinje");

//Provera da li string a sadrzi rec
if(a.contains("tekst"))
    System.out.println("Sadrzi rec");
else
    System.out.println("Ne sadrzi rec");

//leksikografsko poredjenje stringova
if("B".compareTo("A")>0) //razlika pozicija B i A
    System.out.println("posle");
else if("B".compareTo("A")<0)
    System.out.println("pre");
else
    System.out.println("isti");
```

Primer 4. Rad sa promenljivom tipa String. Deljenje (splitovanje) teksta na delove u odnosu na *delimiter*).

Objektno orijentisano programiranje 1

Stringovi i funkcije

```
//Definisanje promenljive tipa string
String a = "Tekst koji se sastoji od 7 reci";
System.out.println(a);

//deljenje stinga na podstringove
//prikazati sve reci iz kojih se sastoji string a
String [] reci = a.split(" ");
System.out.println("String a se sa stoji od " + reci.length + " reci");

for(int i = 0; i < reci.length; i++ )
    System.out.println((i+1) + " rec u stringu a je: " + reci[i] );

String stringBrojeva = "14;2;-3;5";
String [] brojevi = stringBrojeva.split(";");
System.out.println("String a se sa stoji od " + brojevi.length + " brojeva");

for(int i = 0; i < brojevi.length; i++ )
    System.out.println((i+1) + " broj u stringu stringBrojeva je: " + brojevi[i] );
```

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 potprogrami (podalgoritmi) specifične namene.

Definicija funkcije:

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

Za svaku metodu koju definišemo moramo navesti njeno ime, skup ulaznih parametara i tip povratne vrednosti - rezultata funkcije. 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 parametrara 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...)**. Povratna vrednost može biti **primitivni tip**, **referenca na objekat(klasa, niz,mapa...)** ili **bez povratne vrednost** - **void**.

Primer 1. Napisati funkciju koja ispisuje tekst "Hello World" (funkcija nema ulazne parametre i nema povratnu vrednost). Poziv funkcije.
Semantika rezervisane reči **static** ispred imena funkcije.

Objektno orijentisano programiranje 1


Stringovi i funkcije

```
public class Primer1 {  
  
    /*  
    Pozeljno je iznad funkcije napisati kratak komentar  
    kojim se objasnjava programska logika i svrha funkcije  
    */  
    // ispis Hello World teksta  
    static void pozdrav() {  
        System.out.println("Hello World");  
    }  
  
    public static void main(String[] args) {  
        //poziv staticke funkcije  
        pozdrav();  
        //ili na staticki nacin  
        Primer1.pozdrav();  
  
        //u slucaju kada funkcija pozdrav nije staticka  
        Primer2 objekat = new Primer2();  
        objekat.pozdrav();  
    }  
}
```

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

```
//izracunavanje kvadrata hipotenuze pravouglog trougla  
//ulazni parametri su duzine kateta a i b  
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**  (engl. *Type mismatch: cannot convert from...*). Takođe, do greške će doći i ako se zaboravi naredba **return** (engl. *This method must return a result of type...*).

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.

Objektno orijentisano programiranje 1

Stringovi i funkcije

Primer 3. Napisati program i funkciju u kojoj se vrednosti ulaznih parametara menjaju. Ispisati poruke na ekran kojima se potvrđuje da se parametri poziva funkcije prenose po vrednosti.

```
static void promenaVrednostiUlaznogParametaraFunkcije(int a){
    a = 5;
    System.out.println("Promenjena vrednost a unutar funkcije je: " + a);
}
public static void main(String[] args) {
    int a = 3;
    System.out.println("Inicijalna vrednost a je: " + a);
    promenaVrednostiUlaznogParametaraFunkcije(a);
    System.out.println("Vrednost a po povratku iz funkcije je: " + a);
}
```

Java dozvoljava definisanje više funkcija (engl. *method overloading*) sa istim imenom ako i samo ako se razlikuju po broju i/ili tipu ulaznih parametara. U telu jedne funkcije moguće je izvršiti poziv druge funkcije.

Primer 4. Napisati funkcije koje izračunavaju hipotenuzu pravouglog trugla. Posmatrati slučajeve kada se vrednosti kateta razlikuju i kada su vrednosti identične.

```
//ulazni parametri su duzine kateta a i b
static double vrednostHipotenuzePravouglogTrougla(double a, double b){
    double c = 0;
    c = Math.sqrt(a*a + b*b);
    return c;
}

//ulazni parametar je duzina kateta a=b
static double vrednostHipotenuzePravouglogTrougla(double ab){
    return vrednostHipotenuzePravouglogTrougla(ab, ab);
}
```

Primer 5. Rekurzivna funkcija je funkcija koja u svojoj definiciji poziva samu sebe. Koristeći rekurziju odrediti faktorijel od broja N = 10.

```
//Racunanje faktorijela od n.
static int fact(int n) {
    if (n <= 1) {
        // Osnovni slucaj: n=1
        return 1;
    }
    else {
        // rekurzivni slucaj: n > 1 then n! = n * (n-1)!
        return n * fact(n-1);
    }
}
```

Ulazni parametri funkcije pored primitivnih vrednosti mogu biti i reference na objekte. U slučaju da se kao parametar funkcije nađe java objekat (niz, lista, mapa, klasa...) tada se u funkciju po vrednosti prenosi referenca na objekat. Prenosjenje reference omogućuje promenu osobina samih objekata (vrednosti niza, liste...).

Objektno orijentisano programiranje 1

Stringovi i funkcije

Primer 6. Napisati program i funkciju koja menja vrednost prvog elementa niza na 100. Po povratku iz funkcije ispisati sve elemente niza.

```
//promena prvog elementa niza
static void promeniVrednostElNiza(int [] a){
    a[0]=100;
}

public static void main(String[] args) {
    int [] niz = { 1, 2, 3, 5 };
    promeniVrednostElNiza(niz);

    //primer foreach petlje
    for (int el : niz){
        System.out.println("Element niza a je "+ el);
    }
}
```

Primer 7. Sortiranje elemenata niza. Preuređenje niza tako da svi elementi nakon toga zauzimaju rastući ili opadajući poredak. Najmanji element se stavlja na prvo mesto niza. Zatim se prođe kroz preostali niz (od drugog elementa, pa na dalje) i traži se najmanji element koji se postavlja na drugu poziciju. Algoritam se ponavlja sve dok se ne prođe kroz ceo niz.

```
//sortiranje elemenata niza u rastucem
static void selectionSort(int[] niz) {
    for (int i = 0; i < niz.length - 1; i++) {
        for (int j = i; j < niz.length; j++) {
            if (niz[i] > niz [j]) {
                int t = niz[i];
                niz[i] = niz[j];
                niz[j] = t;
            }
        }
    }
}
```

Zadaci

Zadatak 1. Napisati funkciju koja za prosleđeni tekst ispisuje:

- Slova u obrnutom redosledu, jedno pored drugog.
- Reči u obrnutom redosledu, jedno ispod drugog.

Tekst je:

Norvežanin Kjetil Jansrud osvojio je zlatnu medalju u superveleslalomu na Zimskim olimpijskim igrama u Sočiju pošto je za 30 stotih delova sekunde bio brži od drugoplasiranog Amerikanca Endrjua Vajbrehta.

Objektno orijentisano programiranje 1

Stringovi i funkcije

Zadatak 2. Napisati funkciju koja kreira matricu B čije vrednosti se dobijaju parsiranjem stringa "4,3,2,1;0,1,0;1,2,3,4". Pri čemu se karakter ";" koristi kao delimiter vrste a "," kao delimiter kolone. Ispiši sve elemente matrice B, paziti pritom da svaka vrsta nema isti broj kolona.

Zadatak 3. Napisati program i odgovarajuće funkcije koji za niz A:

- određuje najveći element niza (funkcija najvećiEl)
- određuje najmanji element niza (funkcija najmanjiEl)
- određuje srednju vrednost niza (funkcija srednjaVrednost)
- menja vrednost svakog elementa niza tako što negativnim elementima niza dodaje srednju vrednost niza a za pozitivne elemente niza oduzima srednju vrednost niza (funkcija promeniSve).

Zadatak 4. Napisati program i odgovarajuće funkcije koji za nizove A i B izračunavaju niz C. Niz C se dobija tako što se korespondentni elementi nizova A i B sabiraju. $c1 = a1+b1$; $c2 = a2+b2$;... Nizove A i B kreirati u main-u, a zatim proslediti kao parametre funkcije, ime funkcije odabrati proizvoljno.

Zadatak 5. Po ugledu na zadatak 2, kreirati program koji ispisuje podatke o artiklima. Za očitane podatke program nudi različite opcije ispisa i izlaz iz programa.

- Ispisuje spisak naziva svih artikala
- Ispisuje podatke određenog artikla
- Ispisuje samo artikle koji su na akciji
- Ispisuje samo artikle od određenog proizvođača

Tekst je:

```
"Coko plazma | s01 | Bambi | 85.30 | akcija\nSmoki | s02 | Stark | 55.00 | nije na  
akciji\nCipsi | s03 | Marbo | 115.20 | nije na akciji\nKrem Bananica | s04 |  
Stark | 11.00 | akcija\n"
```

Objektno orijentisano programiranje 1

Stringovi i funkcije

Zadatak 6. Napisati program koji realizuje igricu potapanja brodića. Teren predstavlja dvodimenzionalna matrica popunjena nulama, a na mestima gde je postavljen brodić sadrži broj jedan. Jedno polje terena predstavlja jedan element matrice. Igrica se igra tako što dva igrača pokrenu isti program na zasebnim računarima. Rezultati pokušaja igrača se međusobno saopštavaju usmenim putem. Omogućiti sledeće opcije:

- a. Početni meni
 - i. Unos dimenzija terena tj. dvodimenzionalne matrice
 - ii. Unos brodića veličine 4, 3, 2 i 1 polja. Omogućiti izbor horizontalne / vertikalne orijentacije brodića na terenu.
- b. Meni tokom igre
 - i. Unos broj reda i kolone za pokušaj potapanja protivničkog broda
 - ii. Ispis terena sa postavljenim brodićima i terenom sa pokušajima obaranja protivničkih brodića (inicijalno su 0, promašaj x a pogodak 1). Obratiti pažnju da se ispis ponavlja nakon svakog unosa pokušaja potapanja protivničkog broda.

```
*** Potapanje brodića ***
```

```
Teren sa brodićima:
```

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 1 1 0 0
0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 1
```

```
Teren sa pokušajima:
```

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 x x x x 0 0
0 x 0 0 0 0 0 0 0 0
0 x 0 0 0 0 0 0 0 0
0 x 0 0 x 1 1 x 0 0
0 x 0 0 0 0 0 0 0 0
```

```
Unesite broj reda i broj kolone u obliku [red,kolona]:
```

Napomena: Ceo program realizovati upotrebom funkcija i nizova.

Objektno orijentisano programiranje 1
Stringovi i funkcije