



PREDMET

Osnovi Java Programiranja

Čas 5-6

**RAD SA OPERATORIMA, RAD SA MATEMATIČKIM FUNKCIJAMA
I JAVADOC**

Copyright © 2010 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2010 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

Oktobar 2011.

SADRŽAJ

Operatori.....	3
Primer aplikacije	4
Klasa Igrač	4
Klasa Main	5
Primer rada sa matematičkim funkcijama	6
Napomene	7

Čas 5-6

Rad sa operatorima, rad sa matematičkim funkcijama i JavaDoc

Operatori

Kategorija	Operator
Unarni	++ - + - ! ~ () .
Aritmetički	* / % + -
Pomerački	<< >> >>>
Poređenje	< <= > >= instanceof == !=
Bitske operacije	& ^
Logičke	&&
Kondicional	?:
Dodeljivanje	= += -= *= /= %=
Koma	,

Primer inkrementora i diskrementora

Vrednost X	Izraz	Vrednost X	Vrednost Y
5	Y = X++	6	5
5	Y = ++X	6	6
5	Y = X--	4	5

5	$Y = --X$	4	4
---	-----------	---	---

Upotreba kondicionala

```
int a;
int b;
. . .
int min = a < b ? a : b;
```

Upotreba operatora instanceof

```
if(dragan instanceof Student)
{
    Student draganS = (Student) dragan;
}
```

Operatori dodeljivanja

```
int a = 0;
a = a + 1;
a += 1;
a++;
```

```
int b = 8;
b *= 2;
```

Primer aplikacije

Simulacija igre. Dva igrača izvlače kartu naizmenično i upisuju im se koliko su dobili na karti. Pobjednik je igrač koji prvi sakupi više od 100 poena. Karte imaju vrednosti od 2 do 14 (As se računa kao 11).

Klasa Igrač

```
public class Igrac {
    private String naziv;
    private int poeni;
    private Random r = new Random();
    public Igrac() {
    }
    public Igrac(String naziv, int poeni) {
        this.naziv = naziv;
        this.poeni = poeni;
    }
    public Igrac(Igrac i) {
        this.naziv = i.naziv;
    }
}
```

```
        this.poeni = i.poeni;
    }

    public String getNaziv() {
        return naziv;
    }

    public void setNaziv(String naziv) {
        this.naziv = naziv;
    }

    public int getPoeni() {
        return poeni;
    }

    public void setPoeni(int poeni) {
        this.poeni = poeni;
    }

    @Override
    public String toString() {
        return "Igrac " + naziv + " je osvojio " + poeni + " poena";
    }

    public void bacanje() {
        poeni += r.nextInt(13)+2;
    }
}
}
```

Klasa Main

```
public class Main {
    public Main() {
        Igrac igracA = new Igrac();
        Igrac igracB = new Igrac();

        String nazivIgracaA = JOptionPane.showInputDialog("Unesite ime igraca");
        String nazivIgracaB = JOptionPane.showInputDialog("Unesite ime igraca");

        igracA.setNaziv(nazivIgracaA);
        igracB.setNaziv(nazivIgracaB);

        while (igracA.getPoeni() < 100 && igracB.getPoeni() < 100) {
```

```
        igracA.bacanje();
        System.out.println("IgracaA = " + igracA);
        igracB.bacanje();
        System.out.println("IgracaB = " + igracB);
    }
    String pobednik = igracA.getPoeni() > igracB.getPoeni() ?
    "Pobednik je " + igracA : "Pobednik je " + igracB;
    System.out.println(pobednik);
}
public static void main(String[] args) {
    new Main();
}
}
```

Primer rada sa matematičkim funkcijama

```
package matematika;
import javax.swing.JOptionPane;
public class Main {
    public Main() {
        odluka();
    }
    public static void main(String[] args) {
        new Main();
    }
    public void odluka() {
        int broj=0;
        double vrednost=0;
        double rezultat = 0;
        try {
            broj =
            Integer.parseInt(JOptionPane.showInputDialog("<html><font
            color=#FF0000><b>Izracunaj</b></font></html>\n 1. sinus\n
            2. kosinus\n" +
            " 3. zaokruzivanje na veci\n 4. Zaokruzivanje na manji\n
            5. Zaokruzivanje na pribliznu vrednost\n" +
            "6. apsolutna vrednost\n"));
            vrednost =
            Double.parseDouble(JOptionPane.showInputDialog("Unesite
            broj za koji se izracunava matematicka operacija"));
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Molio vas da nuensete
            pravilan broj");
        }
        odluka();
    }
}
```

```

    }
    if(broj == 1) rezultat=sinus(vrednost);
    if(broj == 2) rezultat=kosinus(vrednost);
    if(broj == 3) rezultat=zaokruzivanjeNaVeci(vrednost);
    if(broj == 4) rezultat=zaokruzivanjeNaManji(vrednost);
    if(broj == 5) rezultat=zaokruzivanje(vrednost);
    if(broj == 6) rezultat=apsolutnaVrednost(vrednost);
    JOptionPane.showMessageDialog(null, "Rezultat odabrane
matematicke operacije je: " + rezultat);
}
public double sinus(double value){
    return Math.sin(value);
}

private double kosinus(double vrednost) {
    return Math.cos(vrednost);
}

private double zaokruzivanje(double vrednost) {
    return Math.round(vrednost);
}

private double zaokruzivanjeNaManji(double vrednost) {
    return Math.floor(vrednost);
}

private double zaokruzivanjeNaVeci(double vrednost) {
    return Math.ceil(vrednost);
}

private double apsolutnaVrednost(double vrednost) {
    return Math.abs(vrednost);
}
}

```

Napomene

Trebalo bi obratiti pažnju na deo gde u okviru JOptionPane klase u stringu koristimo html kod.

```

JOptionPane.showInputDialog("<html><font
color=#FF0000><b>Izracunaj</b></font></html>\n 1. sinus\n 2. kosinus\n" +
" 3. zaokruzivanje na veci\n 4. Zaokruzivanje na manji\n 5. Zaokruzivanje
na pribliznu vrednost\n" +
"6. apsolutna vrednost\n"));

```

Upotrebom html koda možemo da uredimo dodatno izgled teksta koji se pojavljuje u dijalozima. HTML kod je sastavni deo java jezika tako da možemo da ga koristimo u svim vizuelnim komponentama gde se nešto predstavlja kao string.

U ovoj naredbi smo imali još jedan trik, kada tokom pisanja koda želimo da nastavimo string u drugom redu to može da se uradi tako što se dva stringa nalepe jedan na drugi pomoću + znaka. Na taj način možemo dugačak tekst na stringu u editoru da napišemo u više redova.

Sve matematičke funkcije su statične, to znači da nemoemo da instanciramo klasu Math da bi koristili te funkcije, odnosno pozivamo matematičke funkcije direktno pozivajuće ih iz naziva klase (bez kreiranja instanci).

U programu smo vršili konverziju string u int i u double pomoću naredbi Integer.parseInt i Double.parseDouble ovo su metode wrapper kalsa prostih promenljivi int i double. S obzirom da je parsiranje moguće samo kad je uneti string zaista broj ceo proces parsiranja smo smestili u try-catch mehanizam koji hvata ako dođe do greške u parsiranju i poziva dijalog kojim se obaveštavamo da nismo dobro uneli broj. Nakon neuspešnog unosa broja ponovo se prikazuje isti dijalog.