

UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA

Goran Savić

OBJEKTNO PROGRAMIRANJE

Java – primeri i zadaci

Novi Sad, 2010.

Primer Java klase

Pravougaonik.java

```
public class Pravougaonik {  
    private double a;  
    private double b;  
  
    public Pravougaonik(double aa, double bb) {  
        a = aa;  
        b = bb;  
    }  
  
    public Pravougaonik(Pravougaonik pp) {  
        a = pp.getA();  
        b = pp.getB();  
    }  
  
    public double getA() {  
        return a;  
    }  
  
    public double getB() {  
        return b;  
    }  
  
    public double getO() {  
        return 2*a + 2*b;  
    }  
  
    public double getP() {  
        return a*b;  
    }  
}
```

- Jedna klasa – jedan fajl sa ekstenzijom .java
- Naziv fajla i naziv klase moraju biti isti
- Kompletna klasa je u jednom fajlu – nisu razdvojeni zaglavlje i implementacija (kao .hpp i .cpp)
- Nije moguće definisati funkcije i promenljive izvan neke klase

- Prenos parametara za klasne tipove (npr. Pravougaonik) se uvek vrši po referenci - ne treba pisati znak & kao u C++
- Primitivni tipovi (npr. int) se prenose po vrednosti

Modifikator pristupa se piše ispred svakog polja i metode

Nema destruktora - poseban pozadinski proces oslobađa nekorišćenu memoriju

Pokretanje Java programa

Test.java

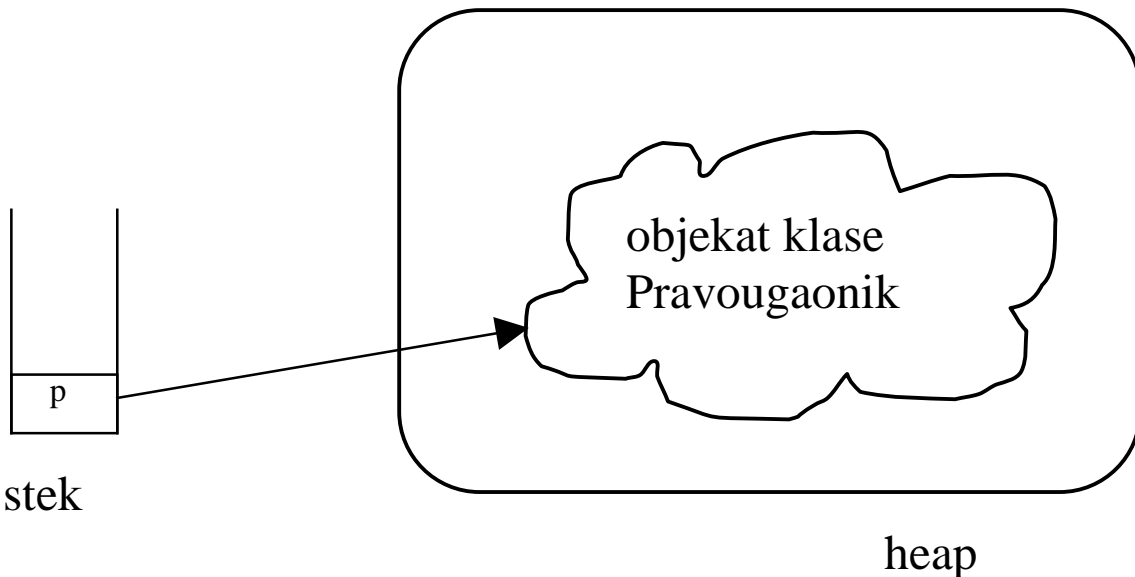
```
public class Test {  
    public static void main(String[] args) {  
  
        Pravougaonik p = new Pravougaonik(5,8);  
  
        System.out.println("Obim je: " + p.getO());  
        System.out.println("Povrsina je: " + p.getP());  
    }  
}
```

- Izvršavanje programa kreće od main metode u klasi
- Svaka klasa koja ima main metodu može da se izvrši

Promenljiva p je referenca na objekat klase Pravougaonik – ne treba pisati znak * kao u C++

Uvek se vrši dinamička alokacija memorije pri konstruisanju objekta

Ispis na konzolu



Zadatak:

U programskom jeziku Java napisati klase Krug i JSTrougao (jednakostranični trougao). Osim potrebnih polja, klase treba da imaju metode koje računaju obim, odnosno površinu. Napisati test program koji testira funkcionalnost ovih klasa.

Nizovi

Nizovi primitivnih tipova

```
double a[] = new double[5];  
int b[] = {1, 7, 9, 3, 4};
```

Kreiranje niza

Kreiranje i inicijalizacija niza

Svaki niz ima polje length

```
for (int i = 0; i < b.length; i++) {  
    System.out.println("Element na poziciji "  
        + i + " je " + b[i]);  
}
```

Pristup elementu niza

Nizovi objekata

```
Pravougaonik p1 = new Pravougaonik(10,2);  
Pravougaonik p2 = new Pravougaonik(3,4);
```

```
Pravougaonik pravougaonici[] = new Pravougaonik[2];  
pravougaonici[0] = p1;  
pravougaonici[1] = p2;
```

```
for (int i = 0; i < pravougaonici.length; i++) {  
    System.out.println("Pravougaonik na poziciji "  
        + i + " ima površinu " + pravougaonici[i].getP());  
}
```

Zadatak:

U programskom jeziku Java napisati klasu koja u okviru main metode kreira niz od 5 objekata klase Pravougaonik i pronalazi pravougaonik sa najvećom površinom.

Kompozicija

Valjak.java

```
public class Valjak {
    private Krug b;
    private Pravougaonik m;

    public Valjak(double r, double h) {
        b = new Krug(r);
        m = new Pravougaonik(b.getO(),h);
    }

    double getR() {
        return b.getR();
    }

    double getH() {
        return m.getB();
    }

    double getP() {
        return 2*b.getP() + m.getP();
    }

    double getV() {
        return b.getP() * getH();
    }
}
```

Instanciranje objekata članova

TestValjak.java

```
public class Test {
    public static void main(String[] args) {
        Valjak v = new Valjak(2,4);
        System.out.println("Povrsina valjka je: " + v.getP());
        System.out.println("Zapremina valjka je: " + v.getV());
    }
}
```

Zadatak:

Napisati klasu PP3Prizma koja modeluje pravu pravilnu trostranu prizmu. Prizmu modelovati kao kompoziciju baze (objekat klase JSTrougao) i omotača (objekat klase Pravougaonik). Klasa treba da sadrži metode koje računaju površinu i zapreminu prizme. Napisati test program koji testira funkcionalnost klase.

Nasleđivanje

- U Javi se klase organizuju u pakete
- Hijerarhija foldera na disku mora da odgovara hijerarhiji paketa

GeometrijskaFigura.java

```
package geometrija.figura;  
  
public abstract class GeometrijskaFigura {  
    public abstract double getObim();  
    public abstract double getPovrsina();  
}
```

Apstraktna klasa

Apstraktna metoda

U Javi je svaka metoda virtuelna. Ne treba pisati virtual

Pravougaonik.java

```
package geometrija.figura;  
  
public class Pravougaonik extends GeometrijskaFigura {  
    protected double a;  
    protected double b;  
  
    public Pravougaonik(double a, double b) {  
        this.a = a;  
        this.b = b;  
    }  
  
    public double getA() {  
        return a;  
    }  
  
    public void setA(double a) {  
        this.a = a;  
    }  
  
    public double getB() {  
        return b;  
    }  
  
    public void setB(double b) {  
        this.b = b;  
    }  
  
    public String toString() {  
        return "Pravougaonik(a=" + a + ", b=" + b + ")";  
    }  
  
    public double getObim() {  
        return 2 * (a + b);  
    }  
}
```

Nasleđivanje

- Svaka klasa u Javi automatski nasleđuje klasu Object
- Redefinisanje metode toString() koja vraća String reprezentaciju objekta

```
    public double getPovrsina() {  
        return a * b;  
    }  
}
```

Kvadrat.java

```
package geometrija.figura;
```

```
public class Kvadrat extends Pravougaonik {  
    public Kvadrat(double a) {  
        super(a, a);  
    }  
  
    public void setA(double a) {  
        super.setA(a);  
        super.setB(a);  
    }  
  
    public void setB(double b) {  
        super.setA(b);  
        super.setB(b);  
    }  
  
    public String toString() {  
        return "Kvadrat(a=" + a + ")";  
    }  
}
```

Poziv konstruktora pretka
(konstruktor klase Pravougaonik)

Poziv metode pretka
(metoda u klasi Pravougaonik)

Krug.java

```
package geometrija.figura;
```

```
public class Krug extends GeometrijskaFigura {  
    protected double r;  
  
    public Krug(double r) {  
        this.r = r;  
    }  
  
    public double getR() {  
        return r;  
    }  
  
    public void setR(double r) {  
        this.r = r;  
    }  
  
    public String toString() {  
        return "Krug(r=" + r + ")";  
    }  
  
    public double getObim() {  
        return 2 * r * Math.PI;  
    }  
}
```

Klasa Math sadrži statičko polje PI

```

    public double getPovrsina() {
        return Math.pow(r, 2) * Math.PI;
    }
}

```

Klasa Math sadrži statičku metodu pow za stepenovanje broja

Test.java

```

package test;

import geometrija.figura.*;

public class Test {

    public static void main(String[] args) {
        Test t = new Test();
        t.testiraj();
    }

    private void testiraj() {
        Pravougaonik p = new Pravougaonik(2,3);
        Krug k = new Krug(4);

        ispisiPovrsinu(p);
        ispisiPovrsinu(k);
    }

    private void ispisiPovrsinu(GeometrijskaFigura g) {
        System.out.println("Povrsina je: " + g.getPovrsina());
    }
}

```

Klasa Test nije u istom paketu kao klase koje predstavljaju figure

Kada se koriste klase iz drugih paketa, moraju se navesti import deklaracijom

Pozvaće se odgovarajuća metoda klase naslednice, jer su sve metode virtualne

Zadatak:

Napisati apstraktnu klasu GeometrijskoTelo sa apstraktnim metodama za izračunavanje površine, odnosno zapremine. Napisati klasu Kvadar koja nasleđuje klasu GeometrijskoTelo i redefiniše potrebne metode. Napisati klasu Kocka koja nasleđuje klasu Kvadar.

Gotove klase

Klasa String

- String u Javi nije samo niz karaktera, nego je posebna klasa
- Inicijalizacija

```
String s1 = new String("tekst");  
String s2 = "tekst";
```

- Poređenje stringova:

```
s1.equals(s2);  
  
s1.equalsIgnoreCase(s2);
```

Ne koristiti == za poređenje stringova, jer se tako porede samo reference, a ne sadržaji

- Stringovi se mogu spajati operatorom +

```
String tekst = "Povrsina je: " + v.getPovrsina();
```

- Korisne metode klase String
 - str.length()
 - str.charAt(i)
 - str.indexOf(s)
 - str.substring(a,b)
 - str.substring(a)
 - str.equals(s)
 - str.equalsIgnoreCase(s)
 - str.startsWith(s)

Klasa ArrayList

- Predstavlja kolekciju elemenata

```
Pravougaonik p1 = new Pravougaonik(2,3);  
Pravougaonik p2 = new Pravougaonik(5,2);  
Pravougaonik p3 = new Pravougaonik(4,6);  
  
ArrayList lista = new ArrayList();
```

Kreiranje kolekcije

```

lista.add(p1);
lista.add(p2);
lista.add(p3);

for (int i = 0; i < lista.size(); i++) {

    Pravougaonik p = (Pravougaonik) lista.get(i);
    System.out.println("Povrsina je: " + p.getPovrsina());
}

lista.remove(p2);

```

Dodavanje elemenata u kolekciju

Preuzimanje broja elemenata u kolekciji

EksPLICITNA konverzija tipa jer metoda `get` vraća `Object`

Preuzimanje i-tog elementa

Izbacivanje elementa iz kolekcije

Tipizirana klasa `ArrayList`

- Predstavlja kolekciju elemenata definisanog tipa

```

Pravougaonik p1 = new Pravougaonik(2,3);
Pravougaonik p2 = new Pravougaonik(5,2);
Pravougaonik p3 = new Pravougaonik(4,6);

ArrayList lista<Pravougaonik> = new ArrayList<Pravougaonik>();

lista.add(p1);
lista.add(p2);
lista.add(p3);

for (int i = 0; i < lista.size(); i++) {

    Pravougaonik p = lista.get(i);
    System.out.println("Povrsina je: " + p.getPovrsina());
}

```

Pri deklaraciji se navodi tip elemenata koji će se nalaziti u kolekciji

Nema potrebe za eksplicitnom konverzijom, jer sada metoda `get` vraća objekat definisane klase

- Može se koristiti posebna sintaksa `for` petlje za prolazak kroz kolekciju

```

for (Pravougaonik p: lista) {
    System.out.println("Povrsina je: " + p.getPovrsina());
}

```