Visoka tehnička škola Niš

Studijski program:

Savremene računarske tehnologije

Internet programiranje

(6)

Osnove klasa u Javi

Prof. dr Zoran Veličković, dipl. inž. el.

Novembar, 2018.

Uvod u Javine klase

- **KLASA** predstavlja **LOGIČKU KONSTRUKCIJU** (celinu), na kojoj se zasniva programski jezik Java, kojom se definiše **OBLIK** i **NAČIN PONAŠANJA** obj.
- Već znamo iz osnova **OOP**-a da se **KLASOM** definišu **NOVI TIPOVI PODATKA** koji se nazivaju **OBJEKTI**.
- Jednom **istom klasom** se može formirati **VIŠE OBJEKATA**.
- Dakle, klasa predstavlja "**ŠABLON**" za formiranje objekta, a **objekat** je samo **PRIMERAK** (često se kaže **INSTANCA**) klase.
- Kao i u .NET-u klasa se **DEKLARIŠE** pomoću rezervisane (ključne) reči **class**.
- **KLASA** se sastoji od **PODATAKA** i **PROGRAMSKOG KODA** (koji se u kontekstu **OO** programiranja naziva **METODA**) koji koristi **TE podatke**.
- Klasa **MORA** da sadrži metodu pod nazivom **main()** samo ako u njoj **POČINJE Java program**.
- Setite se, izuzetak od ovog pravila su **APLETI** (posebna vrsta Java programa - izučavani u Internet tehnologijama) koji **NE POSEDUJU** metodu **main()**.

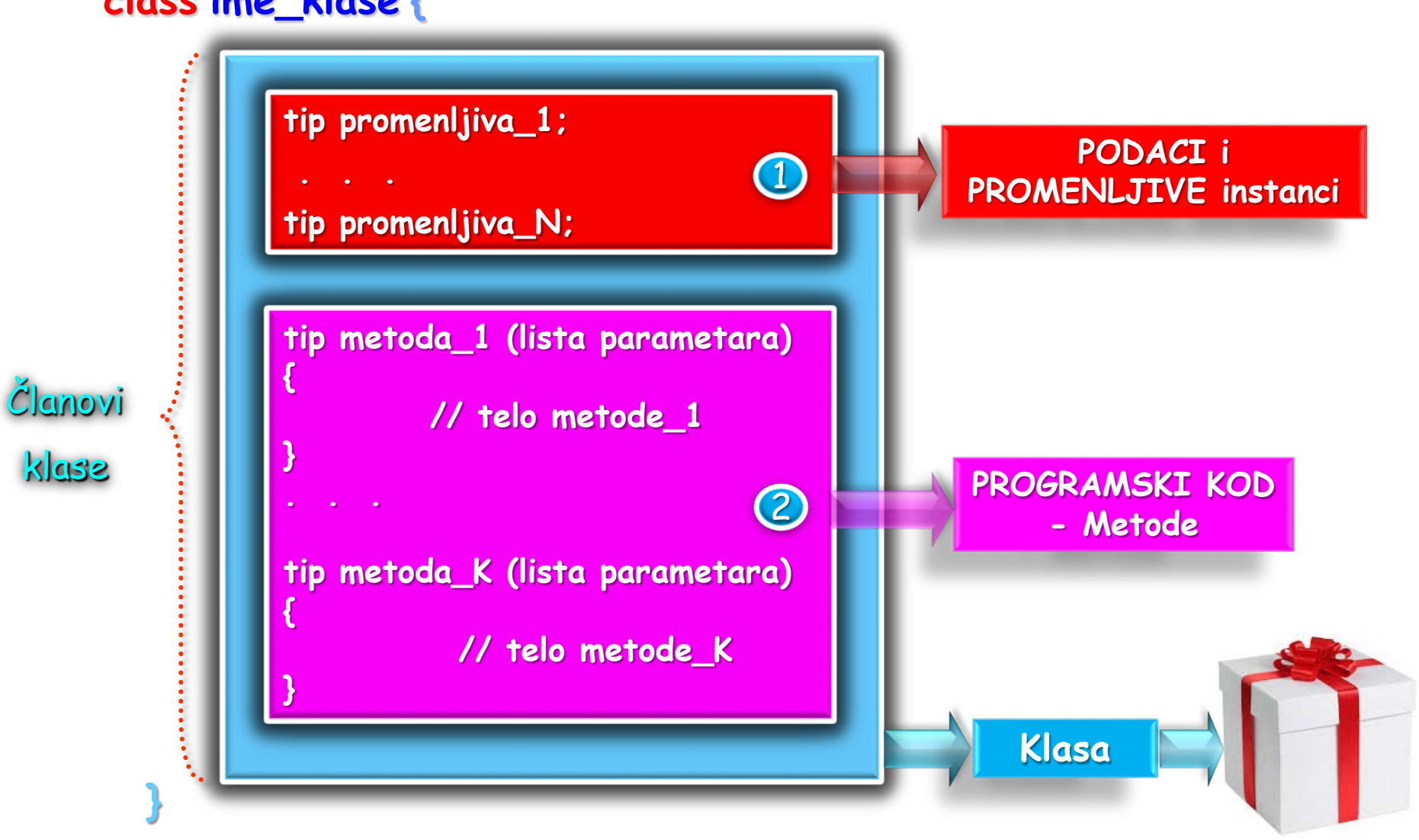
Klase su šabloni objekata

Analogija MODLICE za kolače sa KLASOM

Analogija utisnutog kolačića sa OBJEKTOM

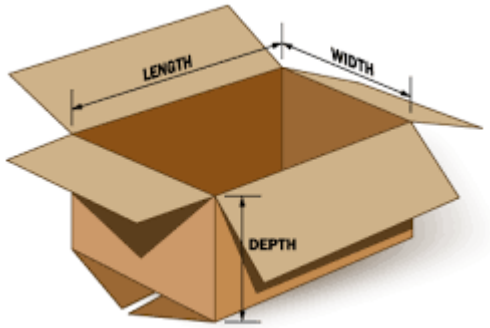
Analogija formiranja SLOŽENOG KOLAČIĆA (objekta) od nekoliko manje složenih kolačića (objekata)

Opšta definicije klase



Kutija - klasa bez metoda (1)

```
class Kutija {
        double širina;
        double visina;
        double dubina;
}
```



- Na ovaj način se formira **ŠABLON** za **NOVI TIP PODATAKA** (generičko – opšte ime ovog tipa je **OBJEKT**) pod nazivom **Kutija** (trenutno je bez metoda).
- **Na osnovu** formirane klase, **INICIJALIZACIJA OBJEKTA** se odvija u **DVA KORAKA**:

Korak ① **FORMIRANJE objekta** tipa **Kutija**:

```
Kutija mojaKutija = new Kutija();
```

Korak ② **DODELJIVANJE VREDNOSTI** promenljivama u objektu:

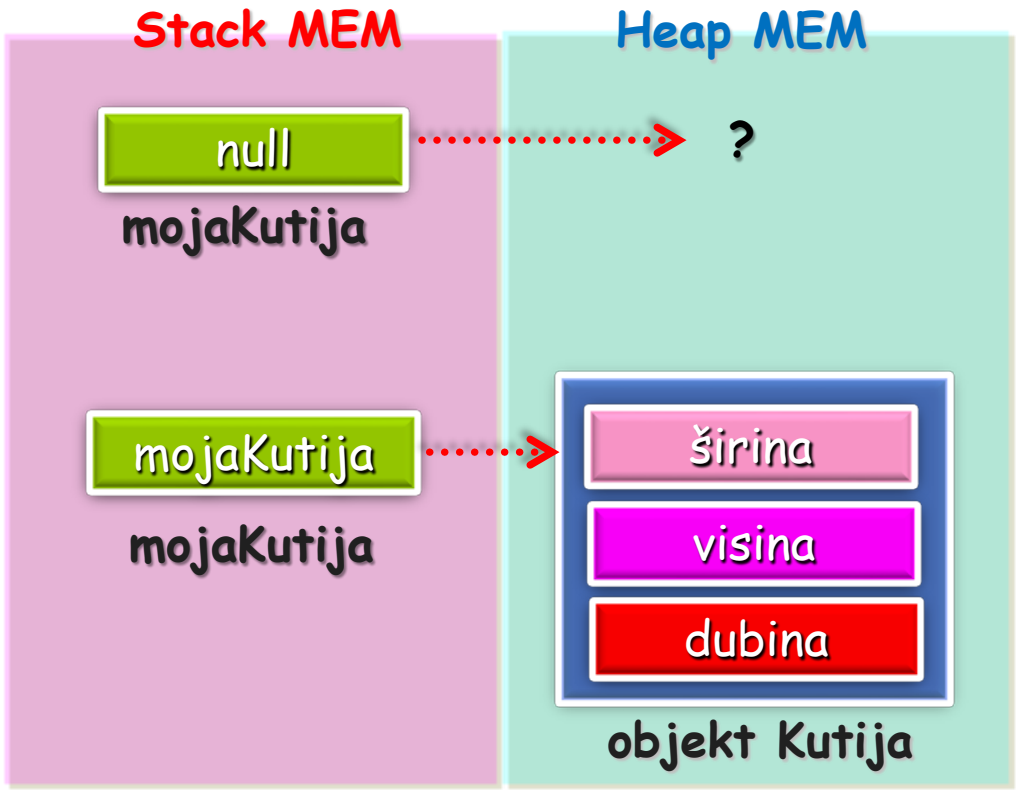
```
mojaKutija.širina = 100;
```

Kutija - klasa bez metoda (2)

„PROGRAMSKI ISKAZ“ „EFEKAT“

1) Kutija mojaKutija;

2) mojaKutija = new Kutija();



1. **Formiranje** objekta tipa **Kutija**
2. **Dodeljivanje** vrednosti deklarisanoj promenljivoj - inicijalizacija.

Operator new

- Operator **new dinamički dodeljuje** memoriju objektu na sledeći način:

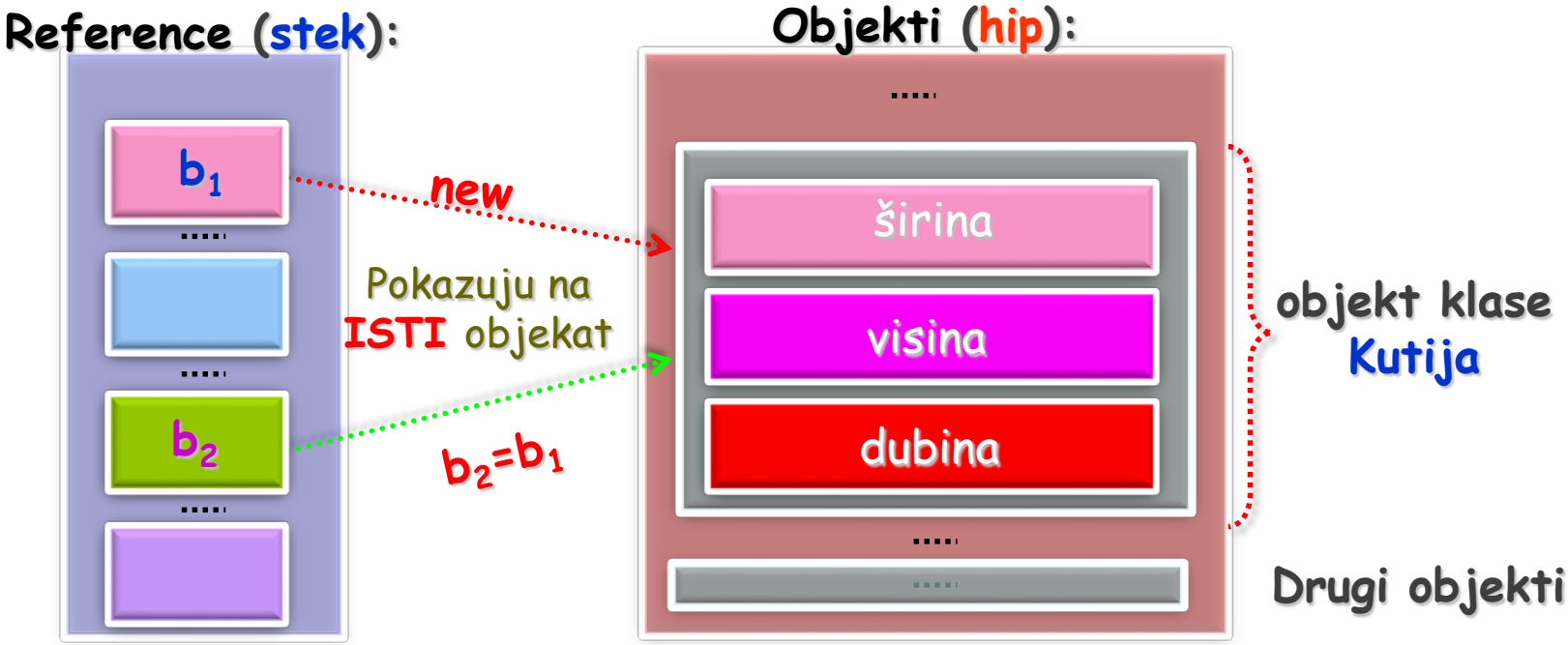
Promenljiva = new imeklase( );

- **Imenom klase** iza koga sledi „**( )**" poziva se spec. metoda - **KONSTRUKTOR**.
- **KONSTRUKTOR** je programski kod kojim se definiše **ŠTA** se tačno **dešava** kada se formira **JEDAN OD OBJEKATA KLASE**.
- Ako ne kreiramo konstruktor u izvornom kodu, Java obezbeđuje **PODRAZUMEVANI KONSTRUKTOR** (kao u prethodno prikazanom primeru).
- Iz .NET-a već znamo da se formiranim objektima može pristupiti samo preko **REFERENCI** koje su napravljene prilikom **kreiranja objekata**.
- Primer **formiranja REFERENCI**:

```
Kutija b1 = new Kutija();

Kutija b2 = b1;
```

Reference na objekte

- Promenljivoj **b2** **NIJE DODELJENA KOPIJA** objekta **b1** već mu se dodeljuje **REFERENCA** na **ISTI OBJEKAT**!
- Ovo je **suštinska razlika** između **VREDNOSNIH** i **REFERENCNIH** tipova podataka (dakle isto kao i u .NET-u).
- Sada bi trebalo da bude jasno **odakle potiče** naziv ovih tipova!



Dodavanje metoda klasi Kutija

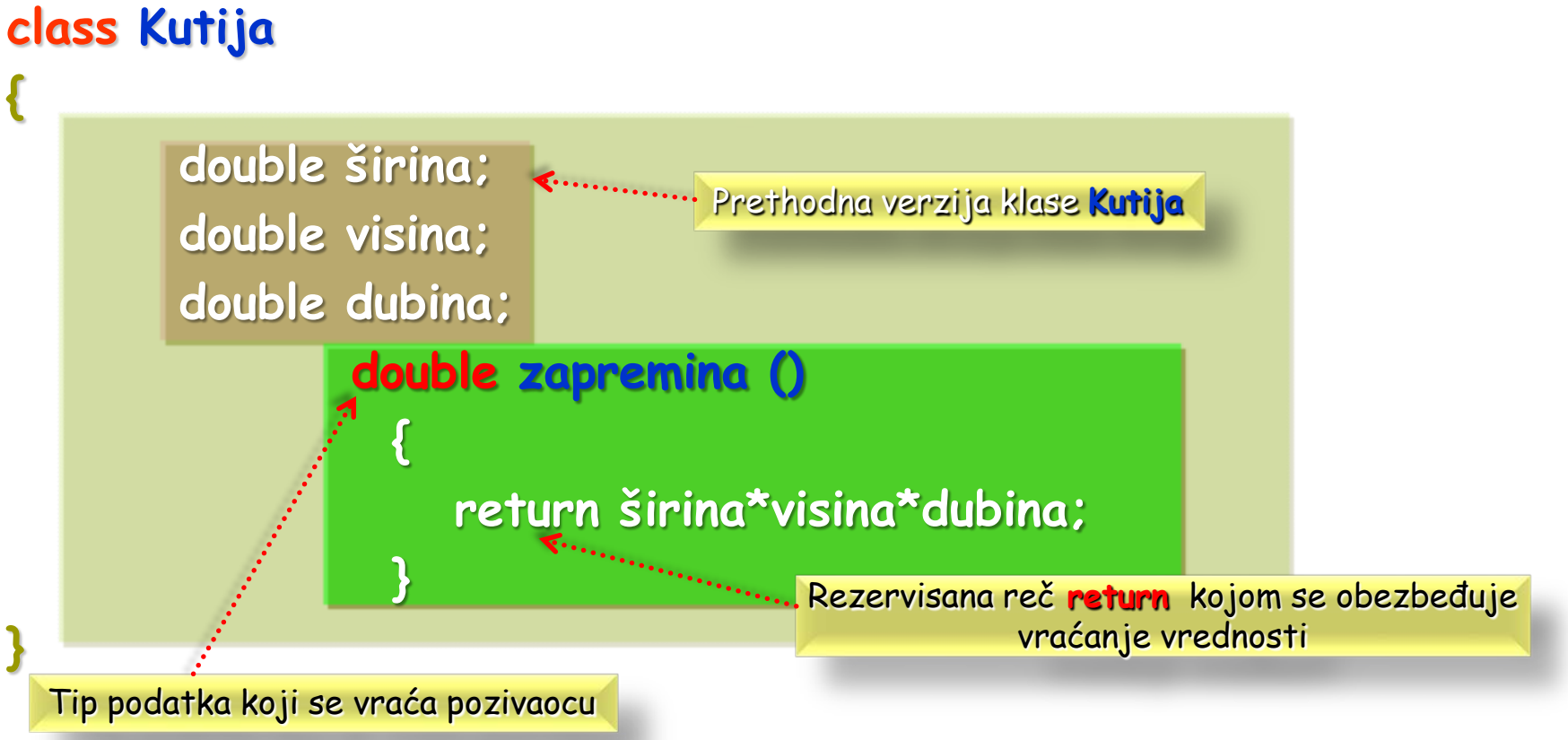
```
class Kutija {
        double širina;
        double visina;
        double dubina;

        // Dodavanje metode: zapremina()
        void zapremina()
         {
                System.out.print("Zapremina je:");
                System.out.print(širina*visina*dubina);
        }
}
```

Prethodna verzija klase **Kutija**

Dodato

- Dodata **METODA** **zapremina()** klasi **Kutija**

Vraćanje vrednosti



- Klasi **Kutija** je dodata metoda **zapremina()** koja **vraća vrednost** pozivnoj metodi (službena reč **return**).
- Ovo je najčešće korišćena verzija metoda.

Konstruktori klasa

```
class Kutija {
        double širina;
        double visina;
        double dubina;
   Kutija() {
                širina=10.0;
                visina=10.0;
                dubina=10.0;
             }
   double zapremina ()
             {
                return širina*visina*dubina;
             }
}
```

KLASA Kutija

KONSTRUKTOR klase Kutija, inicijalizuje vrednost pojedinih parametara. Ima isto ime kao i KLASA!

METODA klase Kutija zapremina()

SCHILDT

Parametarski konstruktori

PARAMETARSKI KONSTRUKTOR
Kutija(s,v,d), pri inicijalizaciji definiše vrednosti parametara s, v i d

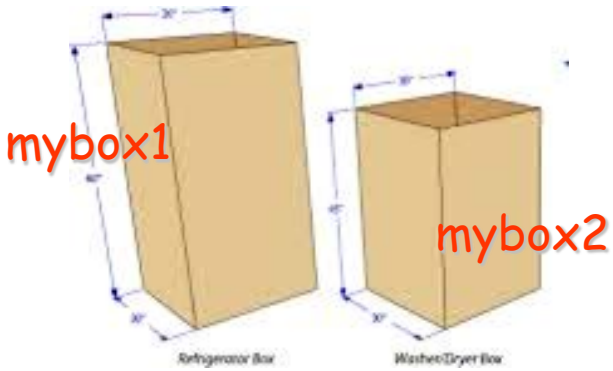
```
class Kutija {
        double širina;
        double visina;
        double dubina;
   Kutija(double s, double v, double d) {
        širina=s;
        visina=v;
        dubina=d;
  }
   double zapremina () {
                return širina*visina*dubina;
   }
}
```

Primer: parametarski konstruktori

```
class BoxDemo7 {
    public static void main (String args[]) {
    Kutija mybox1 = new Kutija (10, 20, 15);
    Kutija mybox2 = new Kutija (3, 6, 9);
    double vol;
    vol = mybox1.zapremina();
    System.out.println(" Zapremina je " + vol);
    vol = mybox2.zapremina();
    System.out.println("Zapremina je " + vol);
  }
}
```

Deklarišu se DVE INSTANCE tipa Box: mybox1 i mybox2. Prva ima parametre 10, 20 i 15, a druga 3, 6 i 9.

Pristup metodi zapremina() klase Kutija



Preklapanje konstruktora (1)

```
class Kutija {
  double širina;  double visina;  double dubina;
Kutija(double w, double h, double d)
    {
       širina = w;
       visina = h;
       dubina = d;
    }
Kutija()
    {
       širina = -1;      // koristi -1 da se ukaže
       visina = -1;      // na neautorizovanu
       dubina = -1;      // kutiju
    }
Kutija(double len)
    { širina = visina = dubina = len; }
double zapremina() {
        return širina * visina * dubina;    } }
```

I. 3-parametra

II. Bez parametara

III. 1-parametar

Mogu se definisati dva ili više **konstruktora** sa **ISTIM IMENOM (Kutija)** sve dok se njihove dekleracije razlikuju (sa ili bez parametra).

Ove metode **(Kutija)** nazivaju se **PREKLOPLJENE** (overloaded) metode.

SCHILDT

Preklapanje konstruktora (2)

```
class PreklapanjeKonstruktora {
  public static void main (String args[]) {
    Kutija mybox1 = new Kutija(10, 20, 15);
    Kutija mybox2 = new Kutija();
    Kutija mycube = new Kutija(7);
    double vol;
  zapremima = mybox1.zapremina();
    System.out.println("Zapremina mybox1 je " + zapremima);
 zapremima = mybox2.zapremina();
    System.out.println(" Zapremina mybox2 je " + zapremima);
 zapremima = mycube.zapremina();
    System.out.println(" Zapremina mycube je " + zapremima);
  }
}
```

Tri poziva PREKLOPLJENOG konstruktora Kutija.

Koja će se metoda Kutija pozvati?

Šta je izlaz?

Preklapanje metoda

- Mogu se definisati **VIŠE METODA** sa **istim imenom** sve dok im se **dekleracija parametara razlikuju**!
- U tom slučaju **metodi su preklopljeni** (engl. overload).
- Na osnovu **tipa i/ili broja argumenata** Java određuje **koja** će se metoda zapravo pozvati!

Preklapanje metoda test(1)

```
class Preklopljene_metode {
  void test() {
    System.out.println("Nema parametara");
  }
void test(int a) {
    System.out.println("a: " + a);
  }
void test(int a, int b) {
    System.out.println("a i b: " + a + " " + b);
  }
double test(double a) {
    System.out.println("double a: " + a);
    return a*a;
   }
}
```

Metode sa **ISTIM IMENOM**, a različitim POTPISOM METODE – **brojem ili tipom** parametara!

SCHILDT

Preklapanje metoda test(2)

```
class Preklapanje {
  public static void main(String args[]) {
  Preklopljene_metode obj = new Preklopljene_metode();
    double result;
// poziv svih verzija test()
    obj.test();
    obj.test(10);
    obj.test(10, 20);
    rezultat = obj.test(123.25);
    System.out.println("Rezultat obj.test(123.25): " + rezultat);
  }
}
```

Metode **test()** sa različitim "**POTPISOM**" - OVERLOAD metode.

Primer klase Faktorijel

```
class Faktorijel {
    int fakt (int n) {
          int rezultat;
          if (n == 1)
              return 1;
          rezultat = fakt (n-1) * n;
          return rezultat;
    }
}

class Rekurzija {
     public static void main (String args[]) {
          Faktorijel f = new Faktorijel ();
          System.out.println("Faktorijel broja 3 je " + f.fakt(3));
          System.out.println("Faktorijel broja 5 je " + f.fakt(5));
          System.out.println("Faktorijel broja 10 je " + f.fakt(10));
    }
}
```

Uporedite rešenje sa prethodnih predavanja.

Klasa Faktorijel u Eclipse-u