

VEŽBA 10: OOP – Nasleđivanje**Primer 1 – Rad sa paketima, klasama i objektima i ponašanje nasleđenog metoda toString() u funkciji njegovog redefinisanja**

```
package com.asss.uup.covek;

public class Osoba {

    // polja
    private String ime;
    private String prezime;
    private int godiste;

    // konstruktori (generisani)
    public Osoba() {
    }

    public Osoba(String ime, String prezime, int godiste) {
        this.ime = ime;
        this.prezime = prezime;
        this.godiste = godiste;
    }

    // nestaticki (objektni) metod
    public void predstaviSe() {
        System.out.println("Dobar dan!");
        System.out.println("Ja sam " + ime + " " + prezime + ".");
        System.out.println("\n" + godiste + "-to sam godiste.");
    }

    // pristupni i metodi mutatori (generisani)
    public String getIme() { return ime; }

    public void setIme(String ime) { this.ime = ime; }

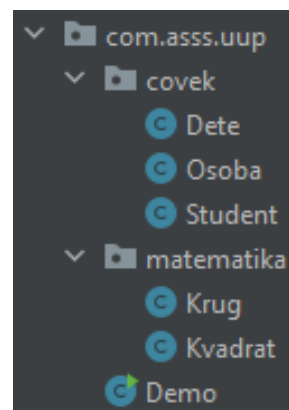
    public String getPrezime() { return prezime; }

    public void setPrezime(String prezime) { this.prezime = prezime; }

    public int getGodiste() { return godiste; }

    public void setGodiste(int godiste) { this.godiste = godiste; }

    // metod za prikaz (generisani)
    @Override
    public String toString() {
        return "Osoba{" +
            "ime=" + ime + '\'' +
            ", prezime=" + prezime + '\'' +
            ", godiste=" + godiste +
            '}';
    }
}
```



```
package com.asss.uup.covek;

public class Dete {

    // polja
    private String ime;
    private double visina;
    private double tezina;

    // konstruktori (generisani)
    public Dete() {
    }

    public Dete(String ime, double visina, double tezina) {
        this.ime = ime;
        this.visina = visina;
        this.tezina = tezina;
    }

    // pristupni i metodi mutatori (generisani)
    public String getIme() { return ime; }

    public void setIme(String ime) { this.ime = ime; }

    public double getVisina() { return visina; }

    public void setVisina(double visina) { this.visina = visina; }

    public double getTezina() { return tezina; }

    public void setTezina(double tezina) { this.tezina = tezina; }

    // metod za prikaz (generisani)
    @Override
    public String toString() {
        return "Dete{" +
            "ime='" + ime + '\'' +
            ", visina=" + visina +
            ", tezina=" + tezina +
            '}';
    }
}
```

```
package com.asss.uup.covek;

public class Student {

    // polja
    private String ime;
    private String prezime;
    private String brIndeksa;
    private int godinaStudija;

    // konstruktori (generisani)
    public Student() {
    }

    public Student(String ime, String prezime, String brIndeksa, int godinaStudija) {
        this.ime = ime;
        this.prezime = prezime;
        this.brIndeksa = brIndeksa;
        this.godinaStudija = godinaStudija;
    }

    // nestaticki (objektni) metod
    public void informacije() {
        System.out.print("Student " + ime + " " + prezime);
        System.out.print(" sa brojem indeksa " + brIndeksa);
        System.out.println(" je student " + godinaStudija + ". godine studija.");
    }

    // pristupni i metodi mutatori
    public String getIme() { return ime; }

    public void setIme(String ime) { this.ime = ime; }

    public String getPrezime() { return prezime; }

    public void setPrezime(String prezime) { this.prezime = prezime; }

    public String getBrIndeksa() { return brIndeksa; }

    public void setBrIndeksa(String brIndeksa) { this.brIndeksa = brIndeksa; }

    public int getGodinaStudija() { return godinaStudija; }

    public void setGodinaStudija(int godinaStudija) { this.godinaStudija = godinaStudija; }
}
```

```
package com.asss.uup.matematika;

public class Krug {

    // polje
    private double poluprecnik;

    // konstruktori (generisani)
    public Krug() {
    }

    public Krug(double poluprecnik) { this.poluprecnik = poluprecnik; }

    // nestaticki (objektni) metodi
    // umesto ...+ Math.PI treba da stoji ...* Math.PI (SETITE SE ZASTO!!!)
    public void obim() { System.out.println("Obim kruga je: " + 2 * poluprecnik + Math.PI); }

    public void površina() { System.out.println("Površina kruga je: " + poluprecnik * poluprecnik * Math.PI); }

    // pristupni i metodi mutatori
    public double getPoluprecnik() { return poluprecnik; }

    public void setPoluprecnik(double poluprecnik) { this.poluprecnik = poluprecnik; }

    // metod za prikaz (generisani)
    @Override
    public String toString() {
        return "Krug{" +
            "poluprecnik=" + poluprecnik +
            '}';
    }
}
```

```
package com.asss.uup.matematika;

public class Kvadrat {

    // polje
    private int stranica;

    // konstruktori (generisani)
    public Kvadrat() {
    }

    public Kvadrat(int stranica) {
        this.stranica = stranica;
    }

    // nestaticki (objektni) metod
    public void površina() { System.out.println("Površina kvadrata je " + Math.pow(stranica, 2)); }

    // pristupni i metod mutator
    public int getStranica() {
        return stranica;
    }

    public void setStranica(int stranica) {
        this.stranica = stranica;
    }
}
```

```
package com.asss.uup;

import com.asss.uup.covek.Dete;
import com.asss.uup.covek.Osoba;
import com.asss.uup.covek.Student;
import com.asss.uup.matematika.Krug;
import com.asss.uup.matematika.Kvadrat;

public class Demo {

    public static void main(String[] args) {

        Osoba osoba1 = new Osoba();
        osoba1.setIme("Mika");
        osoba1.setPrezime("Mikic");
        osoba1.setGodiste(99);

        osoba1.predstaviSe();

        String o = osoba1.toString();
        System.out.println(o);

        System.out.println(osoba1);

        Osoba osoba2 = new Osoba( ime: "Pera", prezime: "Peric", godiste: 97);

        System.out.println(osoba2.getIme() + " - " + osoba2.getPrezime() + " - " + osoba2.getGodiste());

        System.out.println("\n-----\n");

        Dete dete = new Dete( ime: "Djuka", visina: 1.5, tezina: 45);

        System.out.println(dete);

        System.out.println("\n-----\n");

        Student student = new Student( ime: "Laza", prezime: "Lazic", brIndeksa: "001/2020", godinaStudija: 1);

        student.informacije();

        System.out.println(student);

        System.out.println("\n-----\n");

        Krug krug = new Krug( poluprecnik: 7);

        krug.obim();
        krug.povrsina();
        System.out.println(krug);

        System.out.println("\n-----\n");

        Kvadrat kvadrat = new Kvadrat( stranica: 12);

        kvadrat.povrsina();
        System.out.println(kvadrat);

    }
}
```

```
Dobar dan!
Ja sam Mika Mikic.
'99-to sam godiste.
Osoba{ime='Mika', prezime='Mikic', godiste=99}
Osoba{ime='Mika', prezime='Mikic', godiste=99}
Pera - Peric - 97

-----

Dete{ime='Djuka', visina=1.5, tezina=45.0}

-----

Student Laza Lazic sa brojem indeksa 001/2020 je student 1. godine studija.
com.asss.uup.covek.Student@2b98378d

-----

Obim kruga je: 14.03.141592653589793
Povrsina kruga je: 153.93804002589985
Krug{poluprecnik=7.0}

-----

Povrsina kvadrata je 144.0
com.asss.uup.matematika.Kvadrat@29774679
```

Primer 2 – Primena nasleđivanja sa jednom natklasom i jednom potklasom

```
package com.asss.uup;

// natklasa
public class Osoba {

    // polja
    private String ime;
    private String prezime;
    private int godiste;

    // konstruktor
    public Osoba(String ime, String prezime, int godiste) {
        this.ime = ime;
        this.prezime = prezime;
        this.godiste = godiste;
    }

    // objektni metod
    public void predstaviSe() {
        System.out.println("Osoba: " + ime + " " + prezime + " rođena je " + godiste + " godine.");
    }

    // pristupni i metodi mutatori
    public String getIme() { return ime; }

    public void setIme(String ime) { this.ime = ime; }

    public String getPrezime() { return prezime; }

    public void setPrezime(String prezime) { this.prezime = prezime; }

    public int getGodiste() { return godiste; }

    public void setGodiste(int godiste) { this.godiste = godiste; }

    // metod za prikaz
    @Override
    public String toString() {
        return "Osoba{" +
            "ime=" + ime + '\n' +
            "prezime=" + prezime + '\n' +
            "godiste=" + godiste +
            '}';
    }
}
```

```
package com.asss.uup;

// potklasa klase Osoba (prosiruje klasu Osoba)
public class Student extends Osoba {

    // polje (implicitno su nasleđena sva polja klase Osoba)
    private String brIndeksa;

    // nije moguće napraviti ovakav konstruktor jer ga nema u natklasi
    // public Student() {
    // }

    // konstruktor
    public Student(String ime, String prezime, int godiste, String brIndeksa) {
        super(ime, prezime, godiste);
        this.brIndeksa = brIndeksa;
    }

    // nasleđen metod iz natklase (može se redefinisati, a ne mora)
    // @Override
    // public void predstaviSe() {
    //     super.predstaviSe();
    // }

    @Override
    public void predstaviSe() {
        System.out.println("Osoba: " + getIme() + " " + getPrezime() + " rođena je " + getGodiste() + " godine.");
        System.out.println("Broj indeksa: " + brIndeksa);
    }

    // pristupni i metodi mutatori
    public String getBrIndeksa() { return brIndeksa; }

    public void setBrIndeksa(String brIndeksa) { this.brIndeksa = brIndeksa; }
}
```



```
package com.asss.uup;

public class Main {

    public static void main(String[] args) {

        // pravljenje i prikaz objekata
        Osoba osoba = new Osoba( ime: "Mika", prezime: "Mikic", godiste: 2000);
        osoba.predstaviSe();

        System.out.println();

        Student student = new Student( ime: "Pera", prezime: "Peric", godiste: 2001, brIndeksa: "001/2020");
        student.predstaviSe();

        System.out.println();

        // promena vrednosti svojstava i prikaz takvih objekata
        osoba.setIme("Djuka");
        osoba.setPrezime("Djukic");
        osoba.setGodiste(2002);
        osoba.predstaviSe();

        System.out.println();

        student.setIme("Laza");
        student.setPrezime("Lazic");
        student.setGodiste(2003);
        student.setBrIndeksa("999/2019");
        student.predstaviSe();
    }
}
```

Osoba: Mika Mikic rođena je 2000 godine.

Osoba: Pera Peric rođena je 2001 godine.
Broj indeksa: 001/2020

Osoba: Djuka Djukic rođena je 2002 godine.

Osoba: Laza Lazic rođena je 2003 godine.
Broj indeksa: 999/2019

Primer 3 – Primena nasleđivanja sa jednom natklasom i više potklasa

```
package com.asss.uup;

// natklasa
public class BMW {

    // polja
    private TipVozila tipVozila;
    private String model;
    private int snaga;
    private int zapremina;

    // konstruktor
    protected BMW(String model, int snaga, int zapremina) {
        this.model = model;
        this.snaga = snaga;
        this.zapremina = zapremina;
    }

    // pristupni i metodi mutatori
    public TipVozila getTipVozila() { return tipVozila; }

    public void setTipVozila(TipVozila tipVozila) { this.tipVozila = tipVozila; }

    public String getModel() { return model; }

    public void setModel(String model) { this.model = model; }

    public int getSnaga() { return snaga; }

    public void setSnaga(int snaga) { this.snaga = snaga; }

    public int getZapremina() { return zapremina; }

    public void setZapremina(int zapremina) { this.zapremina = zapremina; }

    // metod za prikaz
    @Override
    public String toString() {
        return "BMW (" + this.getTipVozila().getTip() + "): " + getModel() + " (" + getSnaga() + "/" + getZapremina() + ")";
    }
}
```

```
package com.asss.uup;

/*
enum tip
posebna vrsta klase koja nema "obicne" objekte
vec precizno definisane "konstante enumeracije"
(ispod je slozeni oblik koji ima svoja polja i konstruktor,
a moze biti i bez toga - samo nazivi konstanti)
*/

public enum TipVozila {

    // konstante enumeracije
    AUTOMOBIL ("automobil"), MOTOR ("motor");

    // polje
    private String tip;

    // konstruktor
    TipVozila(String tip) { this.tip = tip; }

    // pristupni i metod mutator
    public String getTip() { return tip; }

    public void setTip(String tip) { this.tip = tip; }
}
```

```
package com.asss.uup;

// potklasa klase BMW
public class Automobil extends BMW {

    // konstruktor
    protected Automobil(String model, int snaga, int zapremina) {
        super(model, snaga, zapremina);
        this.setTipVozila(TipVozila.AUTOMOBIL);
    }

    // nema potrebe za ovim metodom jer nasledjuje taj metod iz natklase
    // @Override
    // public String toString() {
    //     return "Automobil{" +
    //         "model='" + model + '\'' +
    //         '}';
    // }
}
```

```
package com.asss.uup;

// potklasa klase BMW
public class Motor extends BMW {

    // konstruktor
    protected Motor(String model, int snaga, int zapremina) {
        super(model, snaga, zapremina);
        this.setTipVozila(TipVozila.MOTOR);
    }

    // nema potrebe za ovim metodom jer nasledjuje taj metod iz natklase
    // @Override
    // public String toString() {
    //     return "Motor{" +
    //         "model='" + model + '\'' +
    //         '>';
    // }
}
```

```
package com.asss.uup;

public class Main {

    public static void main(String[] args) {

        Automobil automobil = new Automobil( model: "M5 Competition", snaga: 625, zapremina: 4395);
        Motor motor = new Motor( model: "M1000RR", snaga: 212, zapremina: 999);

        System.out.println(automobil);

        System.out.println();

        System.out.println(motor);

    }
}
```

BMW (automobil): M5 Competition (625/4395)

BMW (motor): M1000RR (212/999)