

# Programski jezik Java

Interno gradivo za predmet  
Algoritmi in programski jeziki (4. letnik)

**Razredi**  
(nepreciščeno besedilo)



bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Razred

- Deklaracija atributov
- Deklaracija metod
  - Metode *setter*
  - Metode *getter*
  - Uporabne (lastne) metode
- Konstruktorji

```
public class Primer{  
    Deklaracija atributov;  
    Deklaracija konstruktorjev;  
    Deklaracija metod:  
        Metode za nastavljanje vrednosti atributov;  
        Metode za vračanje vrednosti atributov;  
        Uporabne metode;  
}
```



bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Deklaracija atributov

- Vsak atribut je deklariran kot spremenljivka
- Deklaracije atributov niso nič drugega kot deklaracije spremenljivk
- Pri OO upoštevamo princip skrivanja informacij zato dodamo dostopno določilo
  - Objekt se obnaša kot črna škatla, tako da navzven niso vidne podrobnosti, povezane z njegovo realizacijo.
- Primer:
  - `private int matStev;`
  - `private String ime;`
  - `private String priimek;`
  - `private int stUr;`

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Deklaracija metod



- Če atributom deklariramo določilo private potrebujemo metode s pomočjo katerih omogočimo drugim razredom nadzorovan dostop do njihovih vrednosti
- Metode s katerimi nastavljamo vrednosti posameznih atributov
  - Metode setter
- Metode, ki vračajo vrednosti posameznih atributov
  - Metode getter

bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Metode setter



- Tipično imajo en parameter
- Telo metode ima en prireditveni stavek
  - priredimo vrednost parametra ustreznemu atributu
- Metoda mora biti javna
- Primer:
  - ```
public void vpisiMatSt(int ms){  
    matStev=ms;  
}
```
- Če je ime parametra enako imenu atributa mramo uporabiti referenco this (pred imenom atributa)
- Primer:
  - ```
public void vpisiMatSt(int matStev){  
    this.matStev=matStev;  
}
```

bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Metode getter



- Vračajo vrednosti posameznih atributov
- Primer:
  - ```
public int vrniMatStev(){  
    return matStev;  
}
```

bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Uporabne (lastne) metode

- Primer:
  - ```
public void izpisVse(){  
    System.out.println("Matična številka:"+matStev);  
    System.out.println("Številko ur:"+stUr);  
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Konstruktorji

- Konstruktorji so posebne metode, ki se kličejo, ko z operatorjem **new** kreiramo nov objekt.
- Glavna naloga
  - Inicializacija atributov objekta
- Koncept večkratnega definiranja metod - overloading
  - V istem razredu lahko deklariramo več različnih konstruktorjev
    - Morajo se razlikovati v seznamu parametrov

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Konstruktorji

- Primer:
  - ```
public Delavec(){  
}
```
- Ta konstruktor deluje enako kot konstruktor, ki ga Java dodeli avtomatsko, če v razredu ni nobenega konstruktorja.
- Uporaben v primeru ko želimo poleg ostalih konstruktorjev imeti konstruktor, ki vzpostavi začetne vrednosti atributov.

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Konstruktorji

- Primer:
  - ```
public Delavec(int ms, String p, String i){  
    matStev=ms;  
    priimek=p;  
    lme=i;  
}
```
- Ime razreda == ime konstruktorja
- Če hočemo, da drugi razredi kreirajo objekte tipa Delavec mora biti konstruktor javen (public)
- Z razliko od ostalih metod ob deklaraciji konstruktorja ne navajamo podatkovni tip rezultata
- Vrednosti atributov posredujemo preko parametrov
- Telo konstruktorja sestavljajo prireditveni stavki

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Konstruktorji

- Če so imena parametrov enaka imenom atributov je potrebna referenca **this**
- Z referenco **this** ločimo imena atributov od imen parametrov
- Primer:
  - ```
public Delavec(int matStev, String priimek, String ime){  
    this.matStev=matStev;  
    this.priimek=priimek;  
    this.ime=ime;  
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Konstruktorji

- Napaka
- ```
public Oseba(String i, String n){  
    String ime=i;  
    String naslov=n;  
}
```
- Ime in naslov sta krajevni (lokalni) spremenljivki konstruktorja vidni le v njegovem stavčnem bloku
- Prevajalnik ne bo javil napake, stanje predmeta pa ostane nedoločeno.
- Takšne napake je težko odkriti.
- Takšnim napakam se izognemo z uporabo reference **this**.

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Konstruktorji - povzetek



- Ime konstruktorja je enako imenu razreda
- Razred ima lahko enega ali več konstruktorjev, ki se razlikujejo v podpisu (številu ali podatkovnem tipu parametrov)
- Razred je lahko celo brez konstruktorjev, v tem primeru mu pripada privzeti konstruktor
- Konstruktorji ne vračajo nobene vrednosti - njihova naloga je, da vzpostavijo začetno stanje predmeta
- Konstruktor se lahko pokliče le s pomočjo operatorja new in ga ne moremo poklicati kot navadne metode

bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Končna rešitev



- ```
public class Delavec{
    private int matStev;
    private String ime;
    private String priimek;
    private int stUr;
}
public Delavec(){
}
public Delavec(int ms, String p, String i){
    matStev=ms;
    priimek=p;
    ime=i;
}
public Delavec(int ms, String p, String i, int u){
    matStev=ms;
    priimek=p;
    ime=i;
    stUr=u;
}
```

bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Končna rešitev



- ```
public void vpisiMatStev(int matStev){
    this.matStev=matStev;
}
public void vpisiPriimek(String priimek){
    this.priimek=priimek;
}
public void vpisiIme(String ime){
    this.ime=ime;
}
public void vpisiStUr(int stUr){
    this.stUr=stUr;
}
public int vrniMatStev(){
    return matStev;
}
```

bosjan.vouk@tsc.si

---

---

---

---

---

---

---

---

## Končna rešitev

```
public String vrniPriimek(){
    return priimek;
}
public String vrnilme(){
    return ime;
}
public int vrniStUr(){
    return stUr;
}
public double izracunajBrutoOD(double cenaUre){
    return stUr*cenaUre;
}
public void izpisiVse(){
    System.out.println("Maticna stevilka:"+matStev);
    System.out.println("Priimek in ime:"+priimek+" "+ime);
    System.out.println("Stevilo ur:"+stUr+"ur");
}
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Testiranje rešitve

```
public class TestDelavec{
    public static void main(String[] args){
        Delavec d1=new Delavec();
        d1.izpisiVse();
        System.out.println();
        d1.vpisiMatStev(101);
        d1.vpisiPriimek("Bevk");
        d1.vpisiIme("Janko");
        d1.vpisiStUr(182);
        System.out.println("Mat. stev.: "+d1.vrniMatStev());
        System.out.println("Priimek in ime"+d1.vrniPriimek()+" "+d1.vrnilme());
        System.out.println("Stevilo ur: "+d1.vrniStUr());
        System.out.println();
        d1.izpisiVse();
    }
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Testiranje rešitve

```
System.out.println("Placa: "+d1.izracunajBrutoOD(8.25)+" EUR");
System.out.println();
Delavec d2=new Delavec(234, "Novak", "Janez");
d2.vpisiStUr(174);
d2.izpisiVse();
System.out.println("Placa: "+d2.izracunajBrutoOD(12.5)+" EUR");
System.out.println();
Delavec d3=new Delavec(314,"Tavcar","Zdenka",188);
d3.izpisiVse();
System.out.println("Placa: "+d3.izracunajBrutoOD(14.0)+"EUR");
}
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Primer

- V 1. letnik fakultete je bilo lani vpisano ST\_STUD študentov, katerih podatki (vpisna številka, priimek, ime in ocene osmih opravljenih izpitov) so shranjeni v tabeli studenti. Pri tem veljajo naslednje deklaracije:

```
class Student
{
    private int vpisnaStevilka;
    private String ime;
    private String priimek;
    private int[] ocene = new int[8];
}
class TestStudent
{
    final int ST_STUD=10;
    Student[] studenti = new Student[ST_STUD];
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Primer

- Izdelaj razred Delavec, v katerem bo vsak delavec predstavljen kot objekt z atributi matična številka (celo število, ki je enolični identifikator delavca), priimek (niz znakov), ime (niz znakov) in število ur (celo število, ki pove, koliko ur je delavec prebil pri delu).
- Konstruktor naj omogoča inicializacijo vseh atributov razen števila ur, ki se vpiše naknadno z metodo vpisiStUr.
- Za izpis vseh atributov objekta napiši metodo izpisiVse.

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Rešitev

```
public class Delavec
{
    private int matStev;
    private String priimek;
    private String ime;
    private int stUr;
    Delavec(int ms, String p, String i)
    {
        matStev=ms;
        priimek=p;
        ime=i;
    }
    public void vpisiStUr(int u)
    {
        stUr=u;
    }
}

public void izpisiVse()
{
    System.out.println("Maticna
    stevilka: "+matStev);
    System.out.println("Priimek in ime:
    "+priimek+" "+ime);
    System.out.println("Stevilo ur:
    "+stUr);
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## Naloga

- Iz predhodne naloge uporabi razred Delavec in napiši program, ki omogoča shranjevanje podatkov za 100 zaposlenih.

```
class TestZaposleni
{
    final int ST_DEL=100;
    Delavec[] delavci= new Delavec[ST_DEL];
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---

## class TestZaposleni

```
import java.util.*;
class TestZaposleni
{
    public static void main(String[] arg){
        final int ST_DEL=100;
        int matStev, stUr;
        String primek, ime;
        Delavec[] delavci= new Delavec[ST_DEL];
        Scanner vhod= new Scanner(System.in);

        for (int i = 0; i < ST_DEL; i++)
        {
            System.out.println("Vnesi ime");
            ime=vhod.next();
            System.out.println("Vnesi priimek");
            primek=vhod.next();
            System.out.println("Vnesi matricno stevilko");
            matStev = vhod.nextInt();
            System.out.println("Vnesi stevilo ur");
            stUr = vhod.nextInt();
            delavci[i] = new Delavec(matStev,primek,ime);
            delavci[i].vpisiStUr(stUr);
        }
        for (int i = 0; i < ST_DEL; i++)
        {
            System.out.println("Podatki o "+(i+1)+"- uporabniku so sledeci:");
            delavci[i].izpisiVse();
        }
    }
}
```

bosjan.vouk@tsc.si



---

---

---

---

---

---

---

---