Programski jezik Java

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



Razred

- Deklaracija atributov
- Deklaracija metod

 - Metode getter
 - Uporabne(lastne) metode
- Konstruktorji

• public class Primer{

Deklaracija ributov;
Deklaracija ributov;
Deklaracija nestod:
Metode za nastavljanje vrednosti atributov;
Uporabne metode;

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 in stUr;

•	•	•		
	٠		0	
•		0	0	•
•			•	
	0	۰	•	•
	•	•	0	
•	•	•	9	
	•		•	

Deklaracija metod

- Če atributom deklariramo določilo private potrebujemo metode s pomočjo katerih omogočimo drugim razredom nadzorovan
- Metode s katerimi nastavljamo vrednosti posameznih atributov

dostop do njihovih vrednosti

- Metode setter
- · Metode, ki vračajo vrednosti posameznih atributov
 - Metode getter

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)
- - public void vpisiMatSt(int matStev){ this.matStev=matStev;

Metode getter

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

				-
	ō			
•	0		•	
•	0	•		
	0	•		
D (•	0		
•		0		
•		0		



Uporabne (lastne) metode

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

bostjan.vouk@tsc.s

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
 - V istem razredu lahko deklariramo več različnih konstruktorjev
 - Morajo se razlikovati v seznamu parametrov

bostjan.vouk@tsc.s

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.

bostjan.vouk@tsc

_	_	_			
•	•	•			
•	•	•			
۰	•	•	•	•	
•	•	•	•		
•	0	•	•		
•	•	•			
•	•				





Konstruktorji

- Primer:
- public Delavec(int ms, String p, String i){
 matStev=ms;
 priimek=p;
 - . Ime=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 bostjan 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;

bostjan.vouk@tsc.s

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.

bostjan.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

bostjan.vouk@tsc.s

Končna rešitev

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

stUr=u;

bostjan.vouk@tsc.s

Končna rešitev

}

 public void vpisiMatStev(int matStev){ this.matStev=matStev;

this.matStev=matStev

- public void vpisiPriimek(String priimek){ this.priimek=priimek;
- public void vpisilme(String ime){ this.ime=ime;

public void vpisiStUr(int stUr){
 this.stUr=stUr;

 public int vrniMatStev(){ return matStev;

}

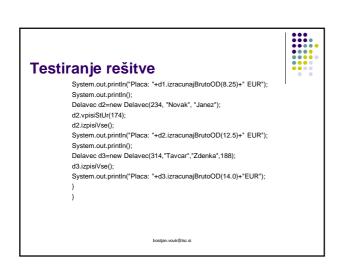
bostjan.vouk@tsc

•	•	•			
	ĕ		•		
•	•		0	•	
•			۰		
		•	•	•	
-	Ξ	=	2		
•	ĕ	_	ě		

_	_

Foncing resites public String vmilrriimek(){ return priimek; } public String vmilme(){ return ime; } public int vmiStUr(){ return stUr; } public double izracunajBrutoOD(double cenaUre){ return stUr*cenaUre; } public void izpisiVse(){ System.out.println("Maticna stevilka:"+matStev); System.out.println("Triimek in ime:"+priimek+" "+ime); System.out.println("Stevilo ur:"+stUr+"ur");

 public class TestDelavec{ public static void main(String[] args){ Delavec d1=new Delavec(); d1.izpisiVse(); System.out.println(); d1.vpisiMatStev(101); 	
d1.izpisiVse(); System.out.println();	
System.out.println();	
d1 uniciMotStoy/101):	
u i.vpisiiviatotev (101),	
d1.vpisiPriimek("Bevk");	
d1.vpisilme("Janko");	
d1.vpisiStUr(182);	
System.out.println("Mat. stev.: "+d1.vrniMatStev());	
System.out.println("Priimek in ime"+d1.vrniPriimek()+" "+d1.vrniIme());	
System.out.println("Stevilo ur: "+d1.vrniStUr());	
System.out.println();	
d1.izpisiVse();	



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];

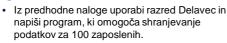
Primer

- 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.

bostjan.vouk@tsc.s

Rešitev

Naloga



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

#