Objektna metodologija

Nasleđivanje (*extends*)
Metodi preklapanja (*override* metodi)
Poziv metoda iz nadklase (*super*)

Nasleđivanje (inheritance)

- Nasleđivanje je relacija (odnos) između dve klase u kojoj je jedna klasa nadklasa a druga podklasa
  - Podklasa nasleđuje se atribute (promenljive, obeležja) i metode (funkcije) svoje nadklase, ali definiše i svoje specifične atribute i metode

Primer:
- Klasa Student nasleđuje klasu Osoba
  - time nasleđuje sve atribute klase Osoba,
    npr.: *mat. broj, ime, adresu, datum rođenja,*
    kao i sve njene metode
- Međutim, definiše i svoje specifične atribute i metode:
  - *broj indeksa, semestar, prosečna ocena,* kao i
  - metode za pristup i obradu tih atributa, koje su karakteristične samo za objekte podklase Student

2

Nasleđivanje – osnovni pojmovi

- Nasleđivanje – relacija među klasama u kojoj podklasa nasleđuje sve atribute i metode iz nadklase, dodajući svoje specifične atribute i metode, koji opisuju razlike podklase u odnosu na njenu nadklasu.
- Nasleđivanjem se definiše hijerarhija među klasama koja je specijalizacija (u smeru na dole), odnosno generalizacija (u smeru ka gore)
- Osnovni termini
  - Nadklasa ili superklasa
  - Podklasa
- Javina biblioteka klasa je uređena uz pomoć relacije nasleđivanja u hijerarhiju klasa
- Superklasa svih klasa Javine biblioteke je klasa Object, koja je na vrhu hijerarhije klasa.

Nasleđivanje – osnovni pojmovi

- Java podržava samo jednostruko nasleđivanje. To znači da svaka klasa može imati tačno jednog roditelja i više potomaka.
- Višestruko nasleđivanje, dopušta da jedna klasa može da ima više roditelja i više potomaka.
- Neki programski jezici omogućavaju višestruko nasleđivanje, ali Java ne.
- Višestruko nasleđivanje, koje nije implementirano u Javi, realizuje se ovde korišćenjem koncepta interfejsa.

Nasleđivanje
Definicija podklase

class *ImePodklase* **extends** *ImeNadklase* {

// telo podklase (ima standardnu strukturu klase)

}

Primer:

```
class Vojnik {.....}            // nadklasa (roditelj)
class Oficir extends Vojnik {   // podklasa (dete)
      .....
}
```

Nasleđivanje
Metodi preklapanja (override methods)

- Metodi preklapanja su metodi sa istim imenom i potpisom (listom argumenata) u klasi i njenoj podklasi.
- Java interpreter realizuje metode preklapanja pretraživanjem odozdo naviše u stablu nasleđivanja.
  - to omogućava pravilno izvršavanje metoda preklapanja, jer se za svaku instancu klase realizuje baš njen metod - njen "model ponašanja".
- Koncept polimorfizma realizovan je preko metoda preklapanja
  - to omogućava izvođenje istih operatora (operacija) na različitim objektima.

Poziv metoda iz superklase - rezervisana reč super

- Reč super je referenca na nadređeni objekat
- Koristi se u podklasi pri pozivu nekog metoda iz nadklase, uključujući i metod konstruktor
- Dve vrste sintakse:
  - `super.imeMetoda(lista_arg);`
    - poziv metoda preklapanja iz nadklase
  - `super(lista_arg);`
    - poziv konstruktora nadklase

7

Primer - nasleđivanje – klase Vojnik i Oficir

Definisati klase za opis vojnika i oficira.

- Obeležja svih su ime, tezina i visina.
- Oficiri imaju još i obeležje čin
- U aplikaciji treba izdvojiti vojnike i oficire sa tezinom od 70 do 95 kg i visinom preko 170 cm

```
import java.util.*;
class Vojnik {
  String ime;
  int tez, vis;
  Vojnik (String ip, int t, int v) {
     ime = ip;
     tez = t;
     vis = v;
  }
```

8

Primer nasleđivanje – klase Vojnik i Oficir

```
Vojnik (String ip) {          // overload konstruktor
    ime = ip;
  }
boolean odgovara( ) {
  if ( (tez > 70 && tez < 95) && (vis > 170) )
      return true;
  else
      return false;
  }
  void prikaziV( ) {
   System.out.print("\nIme i prezime: " + ime);
   System.out.print(", " + tez + "kg");
   System.out.print(", " + vis + "cm");
  }
} // class Vojnik
```

9

Primer nasleđivanje – klase Vojnik i Oficir

```
class Oficir extends Vojnik {        // Oficir nasleđuje Vojnik
 String cin;
 Oficir (String ip, int t, int v, String cin) {
    super(ip, t, v);                 //poziv konstruktora superklase
    this.cin = cin;
 }
 Oficir (String ip, String cin) {    // overload konstruktor
   super(ip);                        //poziv 2. konstruktora superklase
   this.cin = cin;
 }
 void prikaziV( ) {                  // prikaziV() je metod preklapanja
    super.prikaziV( );               // poziv metoda iz nadklase
    System.out.print(", " + cin);
 }
```

10

Primer nasleđivanje – klase Vojnik i Oficir

```
public static void main (String [ ] args) {
  Scanner ulaz = new Scanner(System.in);
  System.out.print("Unesite broj vojnih lica: ");
  int broj = ulaz.nextInt( );
  Vojnik nizVoj[ ] = new Vojnik[broj];
                    //nizVoj je niz objekata klase Vojnik
  System.out.println("Unesite podatke o vojnim licima: ");
  for (int i = 0; i < broj; i++) {
    System.out.println("Ime i prezime: " );
    String ip = ulaz.nextLine( );
    System.out.println("Tezina: " );
    int t = ulaz.nextInt( );
    System.out.println("Visina: " );
    int v = ulaz.nextInt( );
    System.out.println("Da li je oficir?(D/N) " );
    String odg = ulaz.nextLine( );
```

11

Primer nasleđivanje – klase Vojnik i Oficir

```
if (odg.charAt(0) == 'D') {
      System.out.println("Cin: " );
      String ci = ulaz.nextLine( );
      nizVoj[i] = new Oficir(ip, t, v, ci);        // kreiranje objekta Oficir
  }
  else
       nizVoj[i] = new Vojnik(ip, t, v);           // kreiranje objekta Vojnik
  }

   // ispis svih vojnih lica

  System.out.println("\nVojna lica: ");
  for (int i = 0; i < broj; i++) {
        nizVoj[i].prikaziV( );                     // primer polimorfizma
  }
```

12

Primer nasleđivanje – klase Vojnik i Oficir

```
System.out.println
            ("\nVojna lica koja odgovaraju kriterijumu izbora: ");
 for (int i = 0; i < broj; i++){
     if (nizVoj[i].odgovara( ))
         nizVoj[i].prikaziV( );
  } // for
 }  // main
}   // class Oficir
```

Komentar:

- U ovom primeru je klasa Oficir aplikacija (ima main metod) i koristi klasu Vojnik, dok klasa Vojnik nije izvršna klasa
- Drugo, ravnopravno, rešenje bi bilo sa dve klase, Vojnik i Oficir, i trećom klasom - aplikacijom, recimo klasom Vojska, koja ima samo main metod i koristi klase Vojnik i Oficir

13