


Objektna metodologija

Paketi
Java biblioteka

Upotreba paketa

- Ako klasa pripada paketu `java.lang` Javine biblioteke klasa, dovoljno je navesti ime klase
pr. `String, Math, ...`
- Ako klasa pripada nekom drugom paketu, mora se navesti putanja do nje kroz hijerarhiju paketa, odvojena tačkama
pr. `java.awt.Font,
java.awt.Font f1;
f1= new java.awt.Font();`




Paketi - import instrukcija

- Instrukcija `import` uvozi specificiranu klasu ili, ako se navede `*` umesto imena klase, uvozi sve klase iz datog paketa (ali ne i iz podpaketa), u našu klasu

Sintaksa:

```
import imePaketa.ImeKlase;
ili import imePaketa.*;
```

- Nije moguće pisati `import java.util.Vec*;`
`// skracenje od Vector`
- `import` mora biti navedeno **ispred** ključne reči `class`



Paketi - import instrukcija

pr.

```
Ako ispred definicije klase A napišemo


import java.awt.Font;

tada je u klasi A, pri obraćanju klasi Font dovoljno
napisati samo ime klase Font, bez cele putanje
```

pr.

```
Font f2 = new Font( );
```

- samo `public` klase se mogu uvoziti



Paketi Java API

- Napravljene klase i interfejsi se mogu grupisati u zasebne celine koje se u Javi nazivaju **paketi**
- Svaka klasa pripada tačno jednom paketu
- Sve klase i interfejsi u jednom paketu (osim privatnih ugnjeđenih klasa i interfejsa) su vidljivi i u svim delovima paketa
- Postoje standardni paketi klasa koji se isporučuju zajedno sa Java prevodiocem
- Takvi standardni paketi čine **Java API** (Application Programming Interface)

5

Java biblioteka klasa – Java API

- Biblioteka sa preko 1000 klasa organizovanih u pakete po funkcionalnosti:
 - java.lang – jezgro Java jezika, paket dostupan bez `import`
 - java.awt
 - java.io
 - java.util
 - java.security
 - java.applet
 - java.beans
 - java.net
 - java.sql
 - javax.swing
 - java.text
 - java.rmi
 - java.time

6

Definicija paketa

- Instrukcija za definiciju paketa:
`package nazivPaketa;`
- U java source file-u može biti najviše jedna `package` instrukcija
- Obično je to prva instrukcija u java source file-u

Primer:

```
package MojiPaketi.DiplomskiRad.PrviDeo;
```

- Tačka kao simbol određuje hijerarhiju među paketima
- Ako u klasi (fajlu) nema definicije paketa, tada ta klasa pripada anonimnom paketu

7

Zadatak

```
// Vozila mogu biti putnička ili teretna. Imaju sopstvenu težinu, max broj putnika
// i prosečnu težinu putnika (za putnička) odnosno težinu tereta (za teretna).
// Izlistati sva vozila, zatim samo ona vozila koja mogu preći preko mosta
// zadate nosivosti.
import java.util.*;
class Vozilo {
    int težina; // sopstvena težina vozila
    Vozilo () {} // default konstruktor
    Vozilo (int t) {
        težina=t;
    }
    int ukTez() {
        return težina;
    }
    void prikaz() {
        System.out.println("\nTežina vozila " + težina);
    }
} // class
```

8

Zadatak

```

class PVozilo extends Vozilo {
    int brojPut;
    int prosTez;
    PVozilo () {}
    PVozilo (int t, int brP, int pT) {
        super(t);
        brojPut = brP;
        prosTez = pT;
    }
    int ukTez() {
        return super.ukTez() + brojPut*prosTez;
    }
    /* ili ovako:
    int pom;
    pom = super.ukTez() + brojPut*prosTez;
    return pom;
    */
}

```

9

Zadatak

```

void prikaz() {
    super.prikaz();
    System.out.println("Broj putnika " + brojPut);
    System.out.println("Prosečna težina putnika " + prosTez);
    System.out.println("Ukupna težina " + ukTez());
}
} // class
class TVozilo extends Vozilo {
    int terTez;
    TVozilo () {}
    TVozilo (int t, int tT) {
        super(t);
        terTez = tT;
    }
    int ukTez() {
        return super.ukTez() + terTez;
    }
}

```

10

Zadatak

```

void prikaz() {
    super.prikaz();
    System.out.println("Težina tereta " + terTez);
    System.out.println("Ukupna težina " + ukTez());
}
} // class
class Most {
    public static void main (String [] args) {
        Scanner ulaz=new Scanner(System.in);
        System.out.print("Unesite broj vozila: ");
        int broj = ulaz.nextInt();
        Vozilo nizV[] = new Vozilo[broj]; // kreiranje niza objekata
        System.out.println("Unesite podatke o vozilima: ");
    }
}

```

11

Zadatak

```

for (int i=0; i<broj; i++) {
    System.out.println("Težina vozila: ");
    int tez = ulaz.nextInt();
    System.out.println("Da li je putnicko(P) ili teretno(T)? ");
    String odg = ulaz.nextLine();
    if (odg.charAt(0) == 'P') {
        System.out.println("Prosečna težina putnika: ");
        int ptez = ulaz.nextInt();
        System.out.println("Broj putnika: ");
        int brp = ulaz.nextInt();
        nizV[i] = new PVozilo(tez, brp, ptez);
    }
    else {
        System.out.println("Težina tereta: ");
        int ttez = ulaz.nextInt();
        nizV[i] = new TVozilo(tez, ttez);
    }
} // else

```

12

Zadatak

```
} // for
System.out.println("\nSpisak svih vozila");
for (int i=0; i<brój; i++)
    nizV[i].prikaz( );
System.out.println("\nNosivost mosta: ");
int nosivost = ulaz.nextInt( );
System.out.println("Vozila koja mogu preći most nosivosti "
    + nosivost + " su: ");
for (int i=0; i<brój; i++) {
    if(nizV[i].ukTez( ) < nosivost)
        nizV[i].prikaz( );
} // for
} // main
} // class
```

13