

# Paketi

Sve klase u Javi organizovane su u **pakete** (engl. *package*). Paketi su slični direktorijumima koji se koriste za organizovanje podataka na disku. Pod paketom se podrazumeva način grupisanja međusobno srodnih klasa i interfejsa.

Ako program koristi samo nekoliko klasa, nije potrebno praviti paket. Međutim, to nije slučaj sa programima koji imaju veliki broj klasa. Radi jednostavnijeg snalaženja, te klase je potrebno grupisati u pakete, po nameni.

Paketi su korisni iz sledećih razloga:

- omogućavaju organizovanje klasa u module (engl. *units*),
- smanjuju problem konflikta naziva,
- omogućavaju bolju zaštitu klasa, promenljivih i metoda,
- mogu se koristiti za identifikaciju klasa.

Sve standardne klase koje postoje u Javi takođe su organizovane u pakete. Na primer, klase **String** i **StringBuffer** nalaze se u paketu **java.lang**. Ovaj paket nije potrebno naznačavati u kôdu jer je on automatski dostupan u svim programima.

U tabeli 7.1 prikazani su standardni Java paketi.

**Tabela 7.1 Standardni Java paketi**

PAKET	OPIS
java.applet	Klase koje su neophodne za izradu Java apleta.
java.awt	Klase koje pomažu pri pisanju platformski nezavisnih aplikacija sa grafičkim korisničkim interfejsom (GUI).
java.beans	Klase koje podržavaju razvoj JavaBeans aplikacija.
java.io	Klase za izvođenje ulazno-izlaznih operacija.
java.lang	Sadrži osnovne Java klase kao što su Object i System.
java.net	Klase koje se koriste za izradu mrežnih aplikacija. Koriste se u sprezi sa java.io radi čitanja i upisivanja podataka sa mreže.

Tabela 7.1 Standardni Java paketi (nastavak)

PAKET	OPIS
java.rmi	Klase za udaljeni pristup.
java.security	Klase neophodne za izgradnju Java sigurnosnih aplikacija.
java.sql	Klase za pristup bazama podataka.
java.text	Klase koje se koriste za aplikacije koje ne zavise od jezika.
java.util	Sadrži alate i strukture podataka.

Za sve osnovne tipove podataka u Javi, postoje **klase omotači** (engl. *wrapper class*), koje sadrže osnovni tip i dodatne metode za manipulaciju tim osnovnim tipom. To su sledeće klase: **Boolean, Character, Byte, Short, Integer, Long, Float, Double**. Sve te klase se nalaze u paketu **java.lang** i koriste se ako postoji potreba, na primer, za izradu osnovnog tipa od podatka tipa string (metode tipa **parseInt**, **parseDouble**).

Ako su klase u drugom paketu definisane pomoću rezervisane reči **public**, tada se u novi program te klase dodaju preko iskaza **import**. Ovaj iskaz treba da se nalazi na početku datoteke s definicijom klase, ali iza iskaza za definisanje paketa.

Opšti oblik uvoženja klasa iz paketa izgleda ovako:

```
import naziv_paketa.naziv_klase;
```

Klasa se pakuje tako što se u datoteku u kojoj se nalazi klasa na početku doda iskaz za pakovanje. Za to se koristi rezervisana reč **package** i mora biti prva u toj datoteci.

Opšti oblik izrade paketa izgleda ovako:

```
package naziv_paketa[ '.' ime_potpaketa...];
```

## Rešeni zadaci

1. Definirati klasu **Student** u okviru paketa **fakultet** koja sadrži:

- privatni podatak **ime** tipa *String*,
- privatni podatak **prezime** tipa *String*,
- privatni podatak **brojIndeksa** tipa *int*,
- odgovarajuće metode **get** i **set** za sve podatke. Nedoovoljene vrednosti za podatke **ime** i **prezime** su *null* stringovi, a podatak **brojIndeksa** mora biti veći od 100 a manji od 1000.
- metodu **ispisi** koja ispisuje podatke o studentu.

Napisati javnu klasu **TestStudent** u okviru paketa **fakultet** koja pravi objekat klase **Student**. Uneti i ispisati podatke o studentu.

**Rešenje:**

```
// Datoteka Student.java
package fakultet;

public class Student {
    // Deklarisanje podataka
    private String ime;
    private String prezime;
    private int brojIndeksa;

    // Metoda za pristup podatku ime
    public String getIme() {
        return ime;
    }

    // Metoda za postavljanje podatka ime
    public void setIme(String ime) {
        if (ime == null)
            System.out.println("Greška: Ime ne sme biti null.");
        else
            this.ime = ime;
    }

    // Metoda za pristup podatku prezime
    public String getPrezime() {
        return prezime;
    }

    // Metoda za postavljanje podatka prezime
    public void setPrezime(String prezime) {
        if (prezime == null)
            System.out.println("Greška: Prezime ne sme biti null.");
        else
            this.prezime = prezime;
    }

    // Metoda za pristup podatku brojIndeksa
    public int getBrojIndeksa() {
        return brojIndeksa;
    }

    // Metoda za postavljanje podatka brojIndeksa
    public void setBrojIndeksa(int brojIndeksa) {
        if (brojIndeksa > 100 && brojIndeksa < 1000)
            this.brojIndeksa = brojIndeksa;
        else
            System.out.println("Greška: Broj indeksa mora biti između 100 i 1000.");
    }
}
```

```
// Datoteka TestStudent.java
package fakultet;

public class TestStudent {
    public static void main(String[] args) {
        // Izrada objekta klase Student
        Student st = new Student();

        // Postavljanje podataka o studentu
        st.setIme("Laza");
        st.setPrezime("Lazić");
        st.setBrojIndeksa(186);

        // Štampanje podataka o studentu
        System.out.println("Podaci o studentu su: ");
        System.out.println("Ime: " + st.getIme());
        System.out.println("Prezime: " + st.getPrezime());
        System.out.println("Broj indeksa: " + st.getBrojIndeksa());
    }
}
```

### Izlazni rezultati:

Podaci o studentu su:

Ime: Laza

Prezime: Lazić

Broj indeksa: 186

2. Definirati klasu **Kosarkas** u paketu **sportisti** koja sadrži:

- privatni podatak **imePrezime** tipa *String*,
- privatni podatak **tim** tipa *String*,
- privatni podatak **licnaGreska** tipa *int*,
- odgovarajuće javne metode **get** i **set** za ove podatke. Nedoovoljene vrednosti za podatke **imePrezime** i **tim** su *null* stringovi, a **licnaGreska** mora biti veća od 1 i manja od 5. U slučaju pogrešnog unosa ovih vrednosti, potrebno je obavestiti korisnika odgovarajućom porukom.
- redefinisati metodu **toString** klase **Object**. Ova metoda treba da vraća *String* koji sadrži vrednosti ime i prezime košarkaša, tim za koji igra i broj ličnih grešaka.

Napisati javnu klasu **TestKosarkas** u paketu **sportisti** koja pravi objekat klase **Kosarkas**. Uneti i ispisati podatke o košarkašu.

**Rešenje:**

```
// Datoteka Kosarkas.java
package sportisti;

public class Kosarkas {
    // Deklarisanje podataka
    private String imePrezime;
    private String tim;
    private int licnaGreska;

    // Metoda za pristup podatku imePrezime
    public String getImePrezime() {
        return imePrezime;
    }

    // Metoda za postavljanje podatka imePrezime
    public void setImePrezime(String imePrezime) {
        if (imePrezime == null)
            System.out.println("Greška: vrednost podatka ime i prezime ne sme biti null.");
        else
            this.imePrezime = imePrezime;
    }

    // Metoda za pristup podatku tim
    public String getTim() {
        return tim;
    }

    // Metoda za postavljanje podatka tim
    public void setTim(String tim) {
        if (tim == null)
            System.out.println("Greška: vrednost podatka tim ne sme biti null.");
        else
            this.tim = tim;
    }

    // Metoda za pristup podatku licnaGreska
    public int getLicnaGreska() {
        return licnaGreska;
    }

    // Metoda za postavljanje podatka licnaGreska
    public void setLicnaGreska(int licnaGreska) {
        if (licnaGreska < 1 || licnaGreska > 5)
            System.out.println("Greška: vrednost podatka licna greška mora imati vrednost između 1 i 5.");
        else
            this.licnaGreska = licnaGreska;
    }
}
```

```
// Redefinisana metoda
public String toString(){
    return "Ime i prezime: " + imePrezime + "\nTim: " + tim
        + "\nLične greške: " + licnaGreska;
}
}

// Datoteka TestKosarkas.java
package sportisti;

import java.io.*;

public class TestKosarkas {
    public static void main(String[] args) throws Exception {
        BufferedReader ulaz = new BufferedReader(
            new InputStreamReader(System.in));

        // Izrada objekta klase Kosarkas
        Kosarkas k = new Kosarkas();

        // Postavljanje podataka o košarkašu
        System.out.print("Unesite ime i prezime košarkaša: ");
        k.setImePrezime(ulaz.readLine());
        System.out.print("Unesite tim košarkaša: ");
        k.setTim(ulaz.readLine());
        System.out.print("Unesite broj ličnih grešaka: ");
        k.setLicnaGreska(Integer.parseInt(ulaz.readLine()));

        // Štampanje podataka o košarkašu
        System.out.println("\nPodaci o košarkašu su: ");
        System.out.println(k);
    }
}
```

### Izlazni rezultati:

```
Unesite ime i prezime košarkaša: Vlade Divac
Unesite tim košarkaša: Srbija
Unesite broj ličnih grešaka: 2
Podaci o košarkašu su:
Ime i prezime: Vlade Divac
Tim: Srbija
Lične greške: 2
```

### 3. Napisati klasu **Kvadrat** u paketu **geometrija** koja sadrži:

- privatni podatak **stranica** tipa *double* za dimenziju stranice kvadrata,
- parametrizovan konstruktor za inicijalizaciju stranice kvadrata,
- odgovarajuće metode **get** i **set** za podatak **stranica**,

- metodu **obim** za izračunavanje obima kvadrata,
- metodu **povrsina** za izračunavanje površine kvadrata.

Napisati klasu **TestKvadrat** u paketu **geometrija** u kojoj se pravi objekat klase **Kvadrat** i izračunava obim i površina kvadrata ako je **stranica** = 5.0 cm.

### Rešenje:

```
// Datoteka Kvadrat.java
package geometrija;

public class Kvadrat {
    // Deklarisanje podatka stranica
    private double stranica;

    // Podrazumevani konstruktor
    public Kvadrat(double stranica) {
        this.stranica = stranica;
    }

    // Metoda za pristup podatku stranica
    public double getStranica() {
        return stranica;
    }

    // Metoda za postavljanje podatka stranica
    public void setStranica(double stranica) {
        this.stranica = stranica;
    }

    // Metoda za izračunavanje obima kvadrata
    double obim() {
        return 4 * stranica;
    }

    // Metoda za izračunavanje površine kvadrata
    double povrsina() {
        return Math.pow(stranica, 2);
    }
}

// Datoteka TestKvadrat.java
package geometrija;

public class TestKvadrat {
    public static void main(String[] args) {
        // Izrada objekta klase Kvadrat
        Kvadrat k = new Kvadrat(5.0);
    }
}
```

```

// Štampanje rezultata
System.out.println("Obim kvadrata za stranicu a = "
    + k.getStranica() + " iznosi: " + k.obim());
System.out.println("Površina kvadrata za stranicu a = "
    + k.getStranica() + " iznosi: " + k.povrsina());
}
}

```

### Izlazni rezultati:

Obim kvadrata za stranicu a = 5.0 iznosi: 20.0

Površina kvadrata za stranicu a = 5.0 iznosi: 25.0

### Zadaci za vežbu

1. Napisati klasu **Casopis** u paketu *novine* koja sadrži:

- podatak *naslov* tipa *String*,
- podatak *izdavac* tipa *String*,
- podatak *issn* tipa *String* koji predstavlja jedinstvenu šifru časopisa,
- odgovarajuće *get* i *set* metode za sve podatke,
- redefinisanu metodu *toString* klase **Object** koja vraća jedan znakovni niz. Taj znakovni niz sadrži podatke o časopisu: naslov, naziv izdavača i ISSN broj.

Napisati klasu **TestCasopis** u paketu *novine* koja pravi objekat klase **Casopis**. Ispisati podatke o časopisu na ekranu.

2. Napisati javnu klasu **Student** u okviru paketa *studije* koja sadrži:

- privatni podatak *ime* tipa *String*,
- privatni podatak *prezime* tipa *String*,
- privatni podatak *indeks* tipa *int*,
- privatnu metodu *proveriIndeks* koja kao ulazni argument dobija ceo broj koji predstavlja indeks studenta i proverava da li je u opsegu od 200 do 1000. Ako je uslov ispunjen metoda vraća *true*, a u suprotnom *false*.
- odgovarajuće metode *get* i *set* metode za sve podatke. Nedozvoljene vrednosti za *ime* i *prezime* su *null* stringovi.
- zaštićenu metodu *ispisi* koja ispisuje podatke o studentu.

Napisati javnu klasu **TestStudent** u paketu *studije* koja pravi objekat klase **Student**. Uneti i ispisati podatke o studentu.

3. Napisati javnu klasu **Osoba** u okviru paketa *ljudi* koja ima:

- privatni podatak *ime* tipa *String*,
- privatni podatak *prezime* tipa *String*,
- privatni podatak *jmbg* koji predstavlja matični broj tipa *String*,



- privatnu metodu **proveriJmbg** koja kao ulazni argument dobija znakovni niz (string) koji predstavlja JMBG osobe i proverava da li taj string ima tačno 13 znakova. Ako je string različit od *null* i ima tačno 13 znakova, metoda vraća *true*, a u suprotnom *false*.
- odgovarajuće metode **get** i **set** za sve podatke. Nedozvoljene vrednosti za **ime** i **prezime** su *null* stringovi.
- zaštićenu metodu **ispisi** koja ispisuje podatke o osobi.

Napisati javnu klasu **TestOsoba** u paketu **provera** koja pravi objekat klase **Osoba**. Uneti i ispisati podatke o osobi.

4. Definisati klasu **Proizvod** u okviru paketa **proizvodi** koja ima:

- podatak **naziv** tipa *String*,
- podatak **cena** tipa *double*,
- odgovarajuće metode **get** i **set** za sve podatke. Podatak **cena** mora biti veća od nule, a **naziv** ne sme biti *null*.
- redefinisani metodu **toString** klase **Object**. Ova metoda treba da vraća *String* koji sadrži vrednosti svih podataka proizvoda.
- zaštićenu metodu **ispisi** koja ispisuje podatke o proizvodu.

Napisati javnu klasu **TestProizvod** u paketu **proizvodi** koja pravi objekat klase **Proizvod**. Uneti i ispisati podatke o proizvodu.

5. Definisati klasu **JednodimenzioniNiz** u okviru paketa **polje** koja ima:

- podatak **n** tipa *int* koji predstavlja dimenziju niza,
- metodu **unesiNiz** za unos **n** elementa niza,
- metodu **prikaziNiz** za štampanje elemenata niza,
- metodu **presekDvaNiza** za određivanje preseka dva niza dimenzije **n**,
- metodu **unijaDvaNiza** za određivanje unije dva niza dimenzije **n**,
- metodu **razlikaDvaNiza** za određivanje razlike dva niza dimenzije **n**.

Napisati javnu klasu **TestJednodimenzioniNiz** u paketu **polje** koja pravi objekat klase **JednodimenzioniNiz**. Uneti tri niza i naći  $(A \cup B) \cap C$ .

