

4. Klase i objekti

Sadržaj

- Definisane klase
- Obeležja i metode
- Pozivanje metoda
- Statička obeležja i metode

Definisanje klase

- Klase su najčešće javne
- Prilikom deklaracije koristi se rezervisana reč `class`
- Po konvenciji, naziv klase počinje velikim slovom
- Svaka naredna reč u nazivu počinje velikim slovom
- Klase se smešta u fajl koji ima isti naziv kao naziv javne klase i koji ima ekstenziju `.java`

```
public class RoadBike {  
    // telo klase  
}
```

Specifikacija paketa

- Paketi sadrže logički povezane klase
- Odnos paket - klasa sličan je odnosu direktorijum - fajl
- Paketi se mogu ugneždavati
- Nazivi paketa po konvenciji se pišu samo malim slovima
- Prvi izraz u definiciji klase predstavlja specifikaciju pripadnosti paketu

```
package cycling.model;
```

```
public class RoadBike {  
    //...  
}
```

Importovanje klasa

- Ukoliko se unutar definicije jedne klase koristi klasa iz drugog paketa, mora se eksplicitno importovati
- Moguće je importovati čitav paket
- Podrazumeva se da je u svakoj klasi importovan paket `java.lang` koji sadrži često korišćene Java klase
 - `System`
 - `String`
 - `Object`
 - `Math`
 - ...

Primer importovanja klasa i paketa

```
package cycling.model;

import java.awt.Rectangle;
import java.util.*;

public class RoadBike {
    /* moguće korišćenje klase Rectangle,
       svih klasa iz paketa java.util i
       svih klasa iz paketa java.lang */
}
```

Deklaracija obeležja

- Opšti oblik deklaracije obeležja

```
<modPristupa> [static] [final] <tip> <naziv> [init]
```

- Saglasno principu enkapsulacije, obeležja klase se po pravilu deklarišu kao privatna
- Privatnim obeležjima moguć je pristup unutar klase u kojoj su deklarisani
- Ukoliko pri deklaraciji klase nije izvršena inicijalizacija, vrednost obeležja se postavlja na podrazumevanu vrednost tipa kojim je deklarisana

Primer deklaracije obeležja

```
package geometry;

public class Point {
    private int x;
    private int y;
    private boolean selected;
    //...
}
```


Deklaracija metoda u klasi

- Generalno, deklaracija metode može da sadrži:
 - modifikatore
 - povratni tip podatka
 - naziv metode
 - listu parametara u zagradama
 - listu izuzetaka
 - telo metode - kod smešten između vitičastih zagrada
- Po pravilu, metode se deklarišu kao javne, tako da im je moguće pristupiti iz bilo kod dela programa
- Konvencija imenovanja preporučuje da naziv metode počinje malim slovom, a svaka sledeća reč u nazivu velikim slovom

Metode pristupa

- Obično za svako obeležje klase postoje po dve metode pristupa
 - jedna vraća vrednost obeležja (engl. *getter*)
 - druga menja vrednost obeležja (engl. *setter*)
- Imenovanje metoda pristupa se vrši dodavanjem prefiksa *get* ili *set* nazivu obeležja
- Izuzetak su obeležja tipa `boolean`, gde se umesto *get* dodaje prefiks *is*

Primer metoda pristupa (1)

```
public class Point {  
    private int x;  
    ...  
    public int getX() {  
        return this.x;  
    }  
  
    public void setX(int x) {  
        this.x = x;  
    }  
}
```

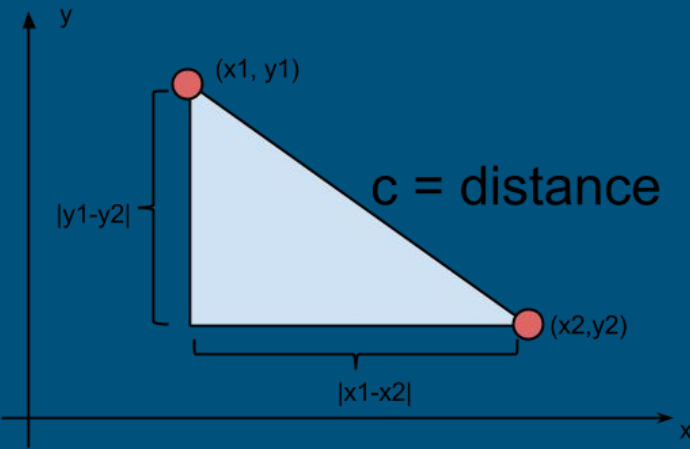
Primer metoda pristupa (2)

```
public class Point {
    private boolean selected;
    ...
    public boolean isSelected() {
        return this.selected;
    }

    public void setSelected(boolean selected) {
        this.selected = selected;
    }
}
```

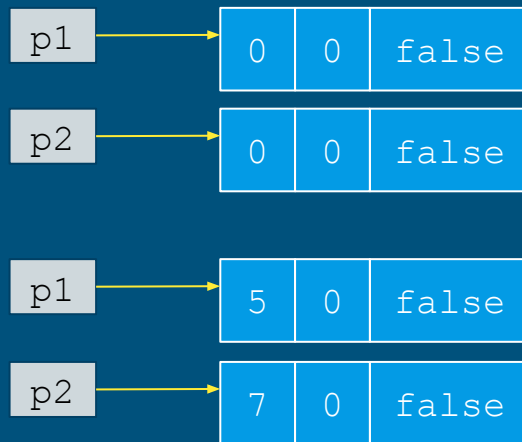
Metoda za izračunavanje udaljenosti tačaka

```
public class Point {  
    ...  
    public double distance(Point p) {  
        int dx = this.x - p.x;  
        int dy = this.y - p.y;  
        double d = Math.sqrt(dx * dx + dy * dy);  
        return d;  
    }  
}
```



Pozivanje metoda

```
public class TestPoint {  
    public static void main(String[] args) {  
        Point p1 = new Point();  
        Point p2 = new Point();  
        p1.setX(5);  
        p2.setX(7);  
        double dist = p1.distance(p2);  
        if (p1.getX() < p2.getX())  
            ...  
    }  
}
```



Podrazumevani konstruktor

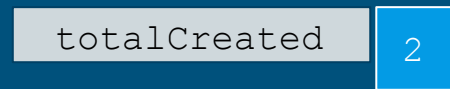
- Konstruktor je specijalna metoda koja vrši inicijalizaciju objekta
- Konstruktor ima isti naziv kao klasa i ne specificira povratni tip podatka
- Ukoliko definicija klase ne sadrži konstruktor, Java kompajler generiše podrazumevani javni konstruktor koji nema parametre i ima prazno telo
- NAPOMENA: Više reči o konstruktorima biće u narednim lekcijama.

Statička obeležja i metode

- Kada postoji obeležje koje je zajedničko za sve objekte klase, ono se deklariše kao statičko
- Statičko obeležje se inicijalizuje prilikom učitavanja definicije klase, pri pokretanju programa
- Pristup statičkom obeležju se po pravilu vrši korišćenjem statičkih metoda
- Statičke metode mogu biti pozvane i kada nije kreiran nijedan objekat klase
- Pristup statičkim metodama i obeležjima vrši se navođenjem naziva klase

Deklaracije statičkih obeležja i metoda

```
public class Point {  
    private static int totalCreated;  
    ...  
    public static int getTotalCreated() {  
        return totalCreated;  
    }  
    ...  
}
```



Poziv staticke metode

```
public class TestPoint {  
    public static void main(String[] args) {  
        Point p1 = new ...  
        ...  
        System.out.println(Point.getTotalCreated());  
    }  
}
```

Primeri statičkih metoda

- Metoda `main`
 - ulazna tačka programa
 - u trenutku poziva u memoriji ne postoji nijedan kreiran objekat
- Metode u klasi `Math`
 - za pozive metoda koje predstavljaju matematičke funkcije
 - izbegava se nepotrebno kreiranje *dummy* objekta potrebnih za poziv nestatičke metode
- **UPOZORENJE: Statičke metode nisu u skladu sa OO pristupom razvoju softvera i treba ih koristiti samo kada za to postoje opravdani razlozi.**

Rezime

- Definicija klase obuhvata deklaracije obeležja i metoda
- Klase se grupišu u pakete
- Za svako obeležje se po pravilu deklariraju metode pristupa
- Statička obeležja imaju jednu vrednost na nivou klase
- Statičke metode mogu da se pozovu kada ne postoji ni jedan objekat klase