

# Object Orient Programming Concepts

- Kas ir objekts?
- Kas ir klase?
- Kas ir mantošana?
- Kas ir pakotne?

# Kas ir objektu orientēta programmēšanas valoda?

- Programmēšanas valoda tiek saukta par objektu orientētu, ja tā atbalsta četras objekta īpašības:
  - abstrakcija
  - iekapsulēšana
  - polimorfisms
  - Mantošana
- Objektorientēta programmatūras izstrāde ir metode, kas programmatūras arhitektūru veido ņemot par pamatu objektus, ar kuriem darbojas katra sistēma vai apakšsistēma (nevis uz funkcijām, ko attiecīgajai sistēmai paredzēts veikt).

# Kas ir klase?

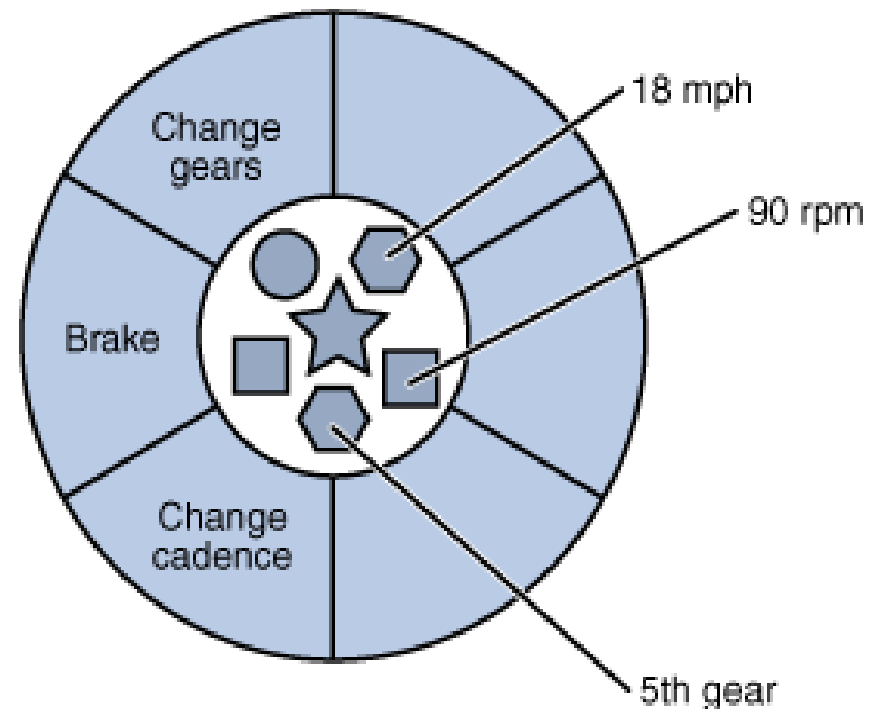
- Klase apraksta gan datus, gan objektu izturēšanos. Klases definīcija ir projekts/plāns (blueprint) uz kā pamata tiek radīti objekti.

```
class Bicycle {  
    int cadence = 0;  
    int speed = 0;  
    int gear = 1;  
    void changeCadence(int newValue) {  
        cadence = newValue;  
    }  
    void changeGear(int newValue) {  
        gear = newValue;  
    }  
    void speedUp(int increment) {  
        speed = speed + increment;  
    }  
    void applyBrakes(int decrement) {  
        speed = speed - decrement;  
    }  
}
```

# Kas ir objekts?

(abstraction)

- **Objekts** ir klases instance. To viennozīmīgi identificē **vārds** un tam katru brīdi ir noteikts stāvoklis, ko nosaka tā atribūtu vērtības :
  - Stāvokli glabā laukos (mainīgajos)
  - Objekta izturēšanos definē šim objektam pielietojamo metožu kopa
  - Metodes ir primārais objektu savstarpējās komunikācijas mehānisms
- Datu iekapsulēšana – iekšējā stāvokļa slēpšana un visu darbību realizācija izmantojot metodes



## Kas ir interfeiss?

- Metodes veido objekta interfeisu ar ārējo pasauli
- Interfeiss iekapsulē objekta kodu un datus.

```
interface Bicycle {  
    void changeCadence(int newValue);  
    void changeGear(int newValue);  
    void speedUp(int increment);  
    void applyBrakes(int decrement);  
}
```

# Kāpēc nepieciešama datu iekapsulēšana?

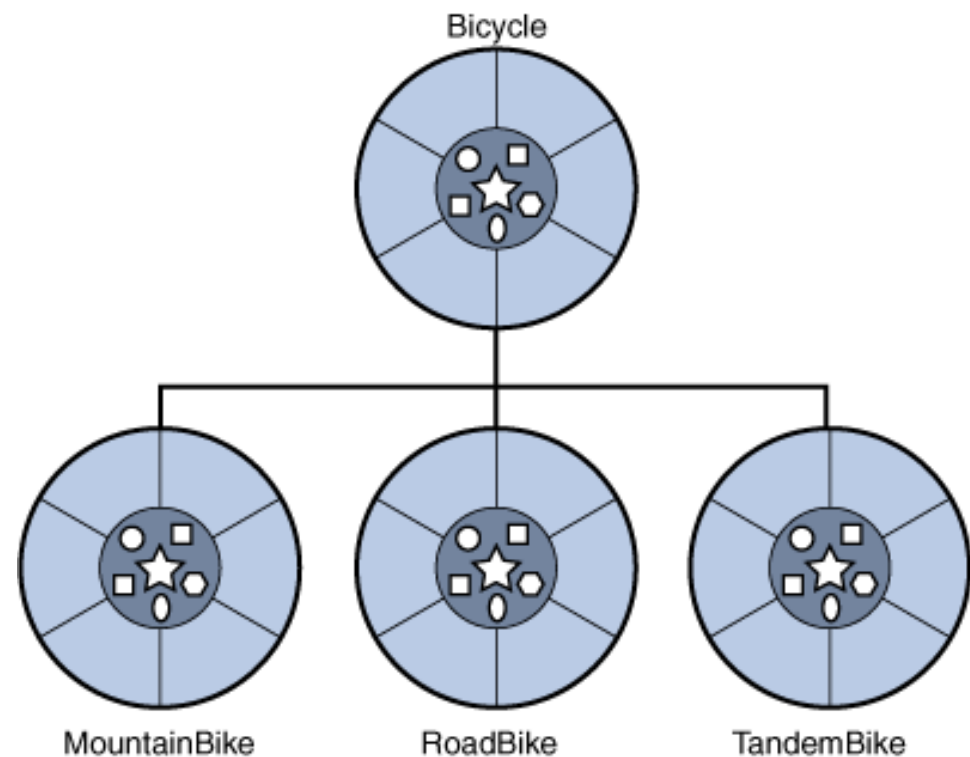
- Modularitāte
  - Objekta izejas kodu var izveidot un uzturēt neatkarīgi no citu objekta koda. Kad objekts ir gatavs, to ir viegli iekļaut kopējā sistēmā.
- Informācijas slēpšana
  - Sadarbojoties ar objektu izmantojot tikai tā metodes, implementācijas detaļas tiek paslēptas no ārpasaules
- Atkal izmantošana
  - Ja objekts ir jau aprakstīts (iespējams to izdarīja kāds cits) jūs varat to izmantot savā programmā. Jūs varat koncentrēties uz savu uzdevumu nevis sadalīt uzmanību uz apkārtējiem objektiem.
- Atklūdošanas vienkāršība
  - Ja objekts izrādās kļūdains, to var vienkārši izmest un paņemt citu (ja tāds eksistē) līdzīgi kā auto – ja saplīst viena skrūve, nemaina taču visu auto.

# Polimorfisms

- Polimorfisms ļauj mainīgajam, funkcijai vai objektam pieņemt dažādas reprezentācijas. Tas ļauj diviem vai vairākiem objektiem atbildēt uz vienu un to pašu ziņojumu dažādā veidā.
- Virsklašu objektu vietā var lietot to apakšklašu objektus.

# Kas ir mantošana?

- **Mantošana** (angl. inheritance) ir mehānisms, kas ļauj klasei A mantot klases B īpašības. Tādējādi klases A objektam ir piekļuves tiesības klases B atribūtiem un metodēm, bez nepieciešamības tos pārdefinēt.
- Ja klase A manto īpašības no klases B, tad saka, ka B ir A **superklase** jeb **virsklase**. Savukārt, A sauc par B **apakšklasi**.





# Kas ir pakotne?

- Pakotnes ir veids kā organizēt saistītas klases un interfeisus.
  - Līdzīgi kā direktorijas uz datoriem
- Java platforma piedāvā ļoti daudz klases, kuras ir sagrupētas pakotnēs  
<http://java.sun.com/j2se/1.4.2/docs/api/>

# Daži jautājumi

- Kas raksturo objektu?  
reālā pasaulē: stāvoklis un metodes  
programmatūrā: lauki un metodes
- Kas ir objekta metožu kopa?  
Interfeiss
- Vai un kāpēc iekapsulēšana ir lietderīga?  
Modularitāte  
Informācijas slēpšana  
Koda atkal izmantošana  
Atklūdošanas vienkāršība