

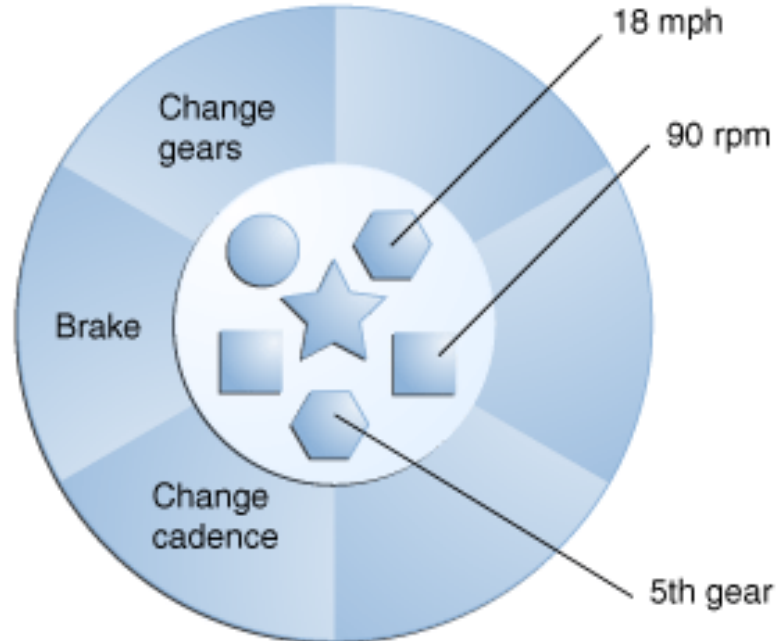
# Koncepti objektno orijentisanog programiranja

# Koncepti

- Objekti
- Klase
- Nasleđivanje
- Interfejsi
- Paketi
- ...

# Objekat

- Primeri iz realnog sveta: pas, sto, televizor
- Svi imaju **stanje** i **ponašanje**
- Objekti mogu imati podobjekte



# Klasa

- Šablon po kojem se kreira objekat
- Objekat je instanca (realizacija) klase

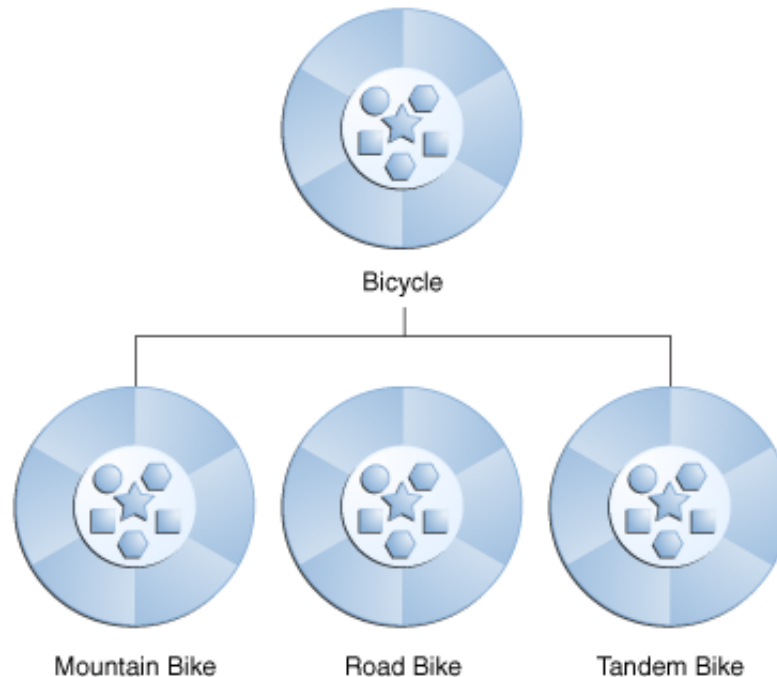
```
class Bicikl {  
    int brzina = 0, pogon = 1; //polja klase  
  
    void promeniPogon(int novaVrednost) { //metod klase  
        pogon = novaVrednost;  
    }  
    void ubrzaj(int ubrzanje) {  
        brzina = brzina + ubrzanje;  
    }  
    void koci(int usporenje) {  
        brzina = brzina - usporenje;  
    }  
    void prikaziStanje() {  
        System.out.println("Brzina: " + brzina + " Pogon:" + pogon);  
    }  
}
```

# Kreiranje objekata (primer)

```
class BiciklDemo {  
  
    public static void main(String[] args) {  
  
        // Kreira dva Bicikl objekta  
        Bicikl bajk1 = new Bicikl();  
        Bicikl bajk2 = new Bicikl();  
  
        // Pozivanje metoda nad objektima  
        bajk1.ubrzaj(10);  
        bajk1.promeniPogon(2);  
        bajk1.prikaziStanje();  
  
        bajk2.ubrzaj(10);  
        bajk2.promeniPogon(2);  
        bajk2.ubrzaj(10);  
        bajk2.promeniPogon(3);  
        bajk2.prikaziStanje();  
    }  
}
```

# Nasleđivanje

- Različiti objekti mogu deliti zajednička svojstva
- Nasleđivanje omogućava da se određena svojstva objekata zadrže i prošire



# Nasleđivanje (primer)

```
class BrdskiBicikl extends Bicikl {  
    int nagib;  
  
    void izmeriNagib(){  
        ...  
    }  
  
    @Override  
    void prikaziStanje(){  
        System.out.println("Brzina: " + brzina + " Pogon:" + pogon+ " Nagib:"  
            + nagib);  
    }  
}
```

# Interfejs

- Skup metoda bez implementacije
- Definišu šta klasa koja ih implementira mora da zadovolji

```
interface Bicikl{  
  
    public void promeniPogon(int novaVrednost);  
    public void ubrzaj(int ubrzanje);  
    public void uspori(int usporenje);  
}  
  
class NekiBicikl implements Bicikl{  
    ...  
}
```



# Paket

- Imenski prostor za skup klasa
- Slično direktorijumima u fajl sistemu
- Služi samo za logičniju organizaciju projekta, nema drugu upotrebu

# Zadatak 1

- Automobil ima pogon i maksimalnu i trenutnu brzinu kretanja
- Reč je o automatik modelu tako vozač može da ubrzava i usporava ( koči ), ali ne i da eksplicitno menja pogon
- Pogon se sam menja na veći stepen pri svakoj dodatnoj promeni od  $1/\text{pogon} * [\text{maksimalna brzina}]$
- Automobil ima 4 točka gde je svaki točak opisan tipom gume, tipom felni i informacijom o ispravnosti
- Automobil pripada vozilima
- Za sva vozila je karakteristično da mogu biti ispravna ili neispravna i da mogu da ubrzaju i usporavaju

# Zadatak 1 - nastavak

- Sva vozila mogu da ispišu svoje trenutno stanje
- Automobil je ispravan (pojednostavljenje) ako su i svi točkovi ispravni
- Postoji 3 tipa automobila: luksuzni, mali gradski i sportski
  - Mali gradski ima mogućnost uključivanja ekonomičnog pogona koji troši manje goriva
  - Sportski može da prebaci u sportski režim rada
  - Luksuzni može da aktivira sistem za eliminaciju proklizavanja
- Pored automobila, vozilo je i kamion...
- Napraviti test primer upotrebe 4 vozila, gde su dva mali gradski automobil, jedan sportski i jedan luksuzni
- Sve pomenute klase i interfejsne smestiti u paket vozila

# Zadatak 2

- Računari se opisuju radnim taktom, količinom RAM i disk memorije i operativnim sistemom
- Svaki računar može da se upali, ugasi, restartuje i koristi
- Postoje varijante računara: stoni (desktop), pokretni (laptop) i tablični (tablet)
- Desktop računar se opisuje ekranom, tastaturom i mišem
- Ekran se opisuje rezolucijom i brzinom osvežavanja
- Laptop ima mogućnost preklapanja
- Tablet ima ekran osetljiv na dodir, tj. može se po njemu i pisati

# Zadatak 2 - nastavak

- Laptop i Tablet pripadaju grupi prenosivih računara za koju važi da se mogu prenositi
- Računar pripada sirem skupu objekata koji se nazivaju elektro-uređaji
- Za svaki uređaj se može proveriti da li je ispravan, i može se ispisati njegovo unutrašnje stanje polja
- Računar je neispravan ako je bilo koja od njegovih komponenti neispravna
- Pegla i televizor su takođe elektro-uređaji
- Sve klase i interfejse ubaciti u paket elektro-uređaji