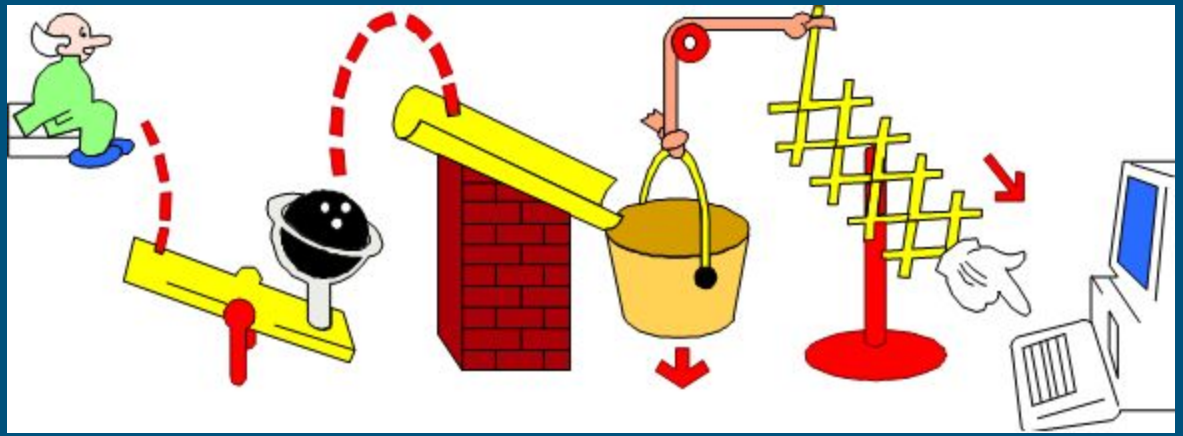
3. Objektno-orijentisani pristup razvoju softvera

Sadržaj

- Objektno-orijentisani (OO) koncepti
- OO programiranje
- Pojam klase i pojam objekta
- Deklaracija klase u Javi
- Slanje poruke objektu
- Referenca `this`

OO paradigma

- Posmatra realan svet kao organizovan skup **objekata**
- Objekti imaju utvrđeno **stanje** i **ponašanje**
- Objekti se nalaze u međusobnoj **interakciji**
- Cilj interakcije je ostvarenje unapred zadatih ciljeva



Pojam objekta

- Objekat je fizička ili konceptualna jedinica posmatranja koja poseduje
  - identitet
  - osobine (obeležja) i
  - operacije
- Objekat može biti
  - kreiran i
  - uništen

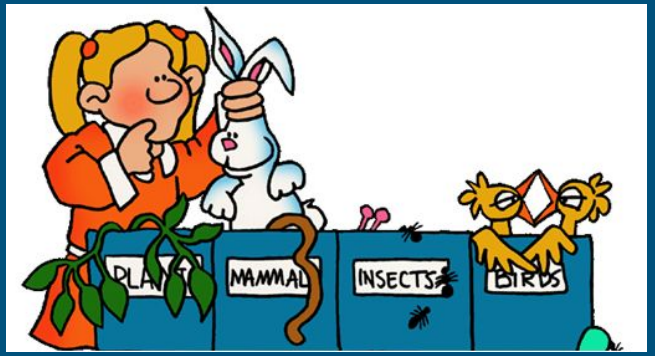
Stanje objekta

- Skup obeležja određuje strukturu objekta
  - svaki automobil ima osobine:
    - godina proizvodnje
    - trenutna brzina
    - količinu goriva u rezervoaru
    - zaključan
- Vrednosti obeležja definišu **stanje** objekta
  - konkretan automobil može imati sledeće vrednosti navedenih osobina:
    - 2013
    - 50 km/h
    - 11.5 litara
    - false

Ponašanje objekta

- Operacije definišu **ponašanje** objekta
  - svaki automobil može da:
    - menja trenutnu brzinu
    - prikazuje trenutnu brzinu
    - menja količinu goriva (troši ili dodaje)
    - prikazuje količinu goriva u rezervoaru
    - zaključava se i otključava
    - prikazuje godinu proizvodnje
- Izvršenje operacije
  - menja stanje objekta i/ili
  - daje informacije o stanju objekta

Osnovni koncepti OO paradigme



- Klasifikacija
  - grupisanje objekata koji imaju isti skup osobina i operacija
  - definiše **klasu** objekata kao skup osobina i operacija
  - klasa se nikada ne definiše u odnosu na **vrednosti** osobina
- Specijalizacija
  - identifikacija podskupa nekog skupa objekata
  - definiše **podklase** neke klase objekata
  - objekti **podklase** imaju sve osobine i operacije **nadklase**, kao i dodatne osobine i operacije
- Polimorfizam
  - sposobnost različitih klasa da realizuju **iste** operacije na **različite** načine
  - brodovi, bicikli i automobili istu operaciju *ubrzaj* realizuju na različite načine

Mehanizmi OO programiranja

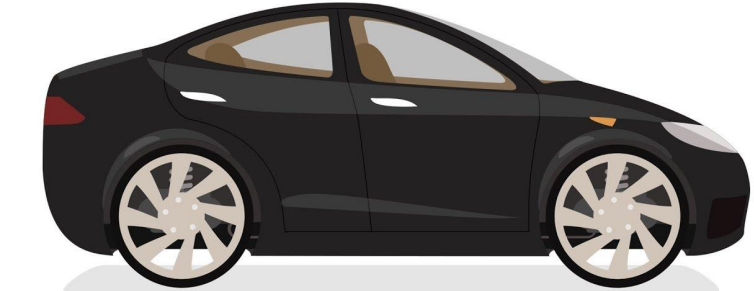
- OO programiranje ima svoje mehanizme pomoću kojih realizuje univerzalne OO koncepte
- Enkapsulacija
  - kontrolisano menjanje stanja objekta
  - ograničeno prikazivanje informacija o stanja objekta
- Proširivanje
  - višestruko korišćenje koda proširivanjem postojeće klase
- Dinamičko povezivanje
  - implementacija polimorfizma

Klasa i objekat u OO programiranju

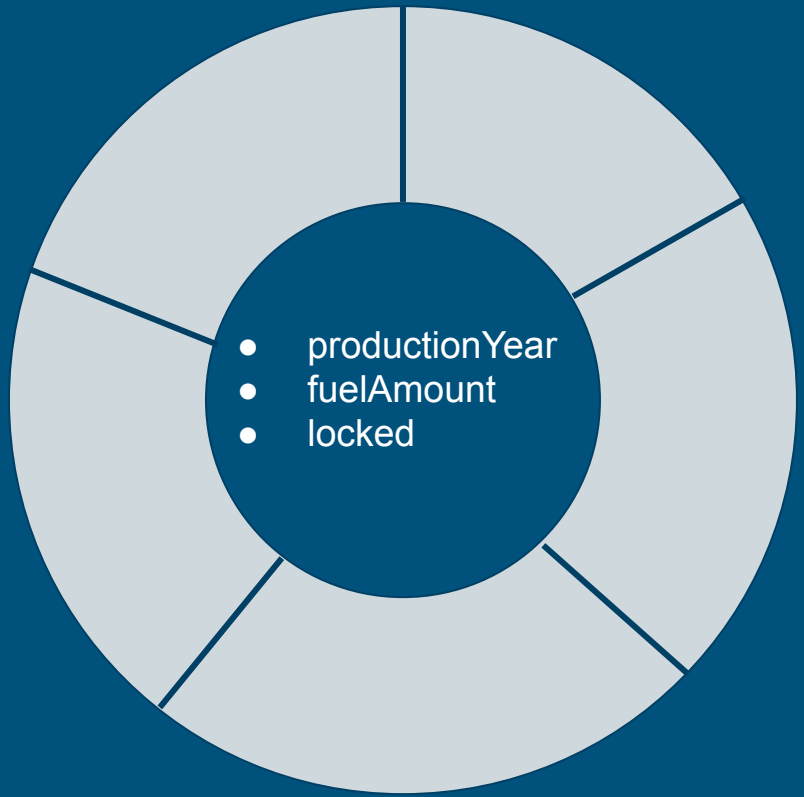
- Klasa
  - definiše obeležja i operacije objekta
  - je “kalup” koji definiše strukturu objekta
- Objekat je
  - je “odlivak” kreiran iz “kalupa”
  - je jedna instanca (pojava) klase
  - ima strukturu i ponašanje definisano u klasi

Stanje objekta

- Objekat sadrži informacije o svom tretnutnom stanju
- Informacije o stanju su u stvari vrednosti obeležja objekta
- Skup obeležja objekta definisan je klasom objekta



Donut dijagram (stanje)



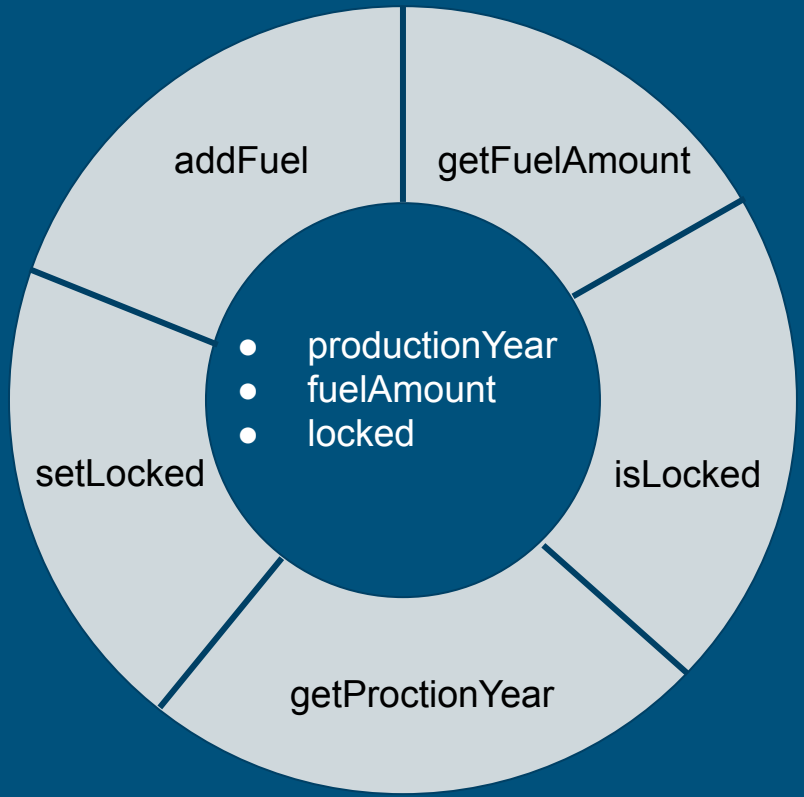
Deklaracija obeležja (osobina)

```
public class Car {

  private int productionYear;
  private double fuelAmount = 5.0;
  private boolean locked;

  ...
}
```

Ponašanje objekta

- Ponašanje objekta je određeno operacijama koje su definisane u njegovoj klasi
- Svi objekti iste klase imaju isto ponašanje

Donut dijagram (ponašanje)



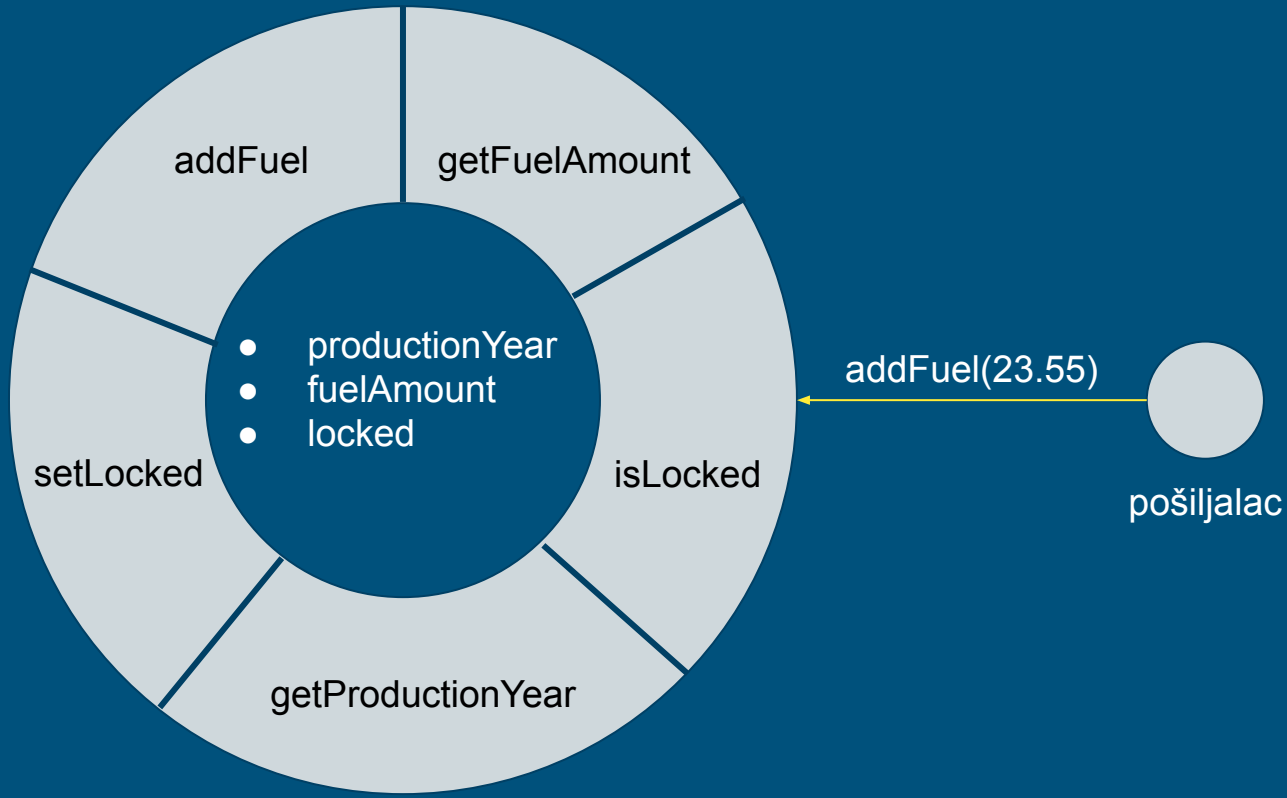
Deklaracija metoda (operacija)

```
public class Car {
   ...
   public int getProductionYear() {...}
   public double getFuelAmount() {...}
   public void addFuel(double additionalFuelAmount) {...}
   public boolean isLocked() {...}
   public void setLocked(boolean locked) {...}
}
```

Slanje poruke objektu

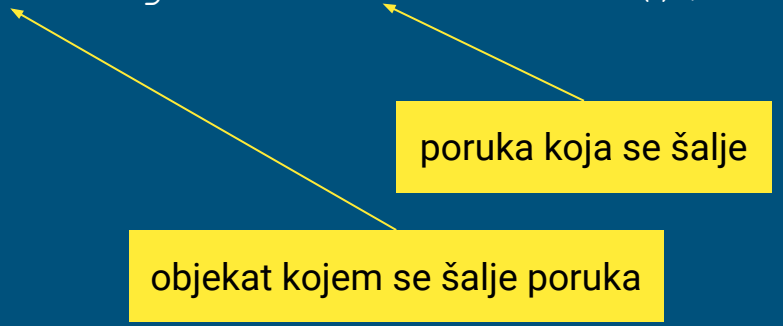
- Objekti u vreme izvršenja programa stupaju u interakciju
- Tokom interakcije, objekti jedni drugima šalju poruke
- Posledica slanja poruke je izvršenje operacije
  - navedene pri slanju poruke
  - definisane u klasi kojoj objekat (kome se šalje poruka) pripada

Donut dijagram (ponašanje)



Slanje poruke u programu

```
Car firstCar = ...      // kreiran jedan objekat
Car secondCar = ...     // kreiran drugi objekat
...
int firstCarsAge = 2020 - firstCar.getProductionYear();
...
secondCar.setLocked(true);
...
firstCar.addFuel(23.55);
```



Referenca `this`

- Rezervisana reč u Javi
- Referencira objekat kojem je poslata poruka
- Omogućava da paramatar metode ima isti naziv kao obeležje klase
- Može se izostaviti ukoliko se naziv parametra razlikuje od naziva obeležja klase

Implementacija metoda (1)

```
public class Car {
   ...
   private int productionYear;
   ...
   public int getProductionYear() {
      return this.productionYear;
   }
   ...
}
```

Implementacija metoda (2)

```
public class Car {
  private boolean locked;
  ...
  public boolean isLocked() {
    return this.locked;
  }
  public void setLocked(boolean locked) {
    this.locked = locked;
  }
}
```

Implementacija metoda (3)

```
public class Car {
  private double fuelAmount;
  ...
  public double getFuelAmount() {
    return this.fuelAmount;
  }
  public void addFuel(double additionalFuelAmout) {
    double loss = 0.002;
    fuelAmount = fuelAmount + (1 - loss) * additionalFuelAmount;
  }
}
```

Rezime

- OO paradigma nudi nove koncepte u razvoju softvera
- OO programski jezici obezbeđuju mehanizme za implementaciju OO koncepata
- Java kao OO jezik obezbeđuje razvoj softvera u skladu sa OO pristupom