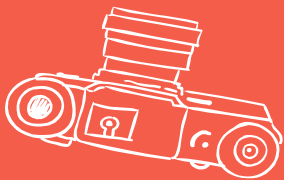


# OBJEKTNO ORIJENTIRANO PROGRAMIRANJE



# ČETIRI STUPA OBJEKTNO ORIJENTIRANOG PROGRAMIRANJA

ENKAPSULACIJA

NASLJEĐIVANJE

APSTRAKCIJA

POLIMORFIZAM

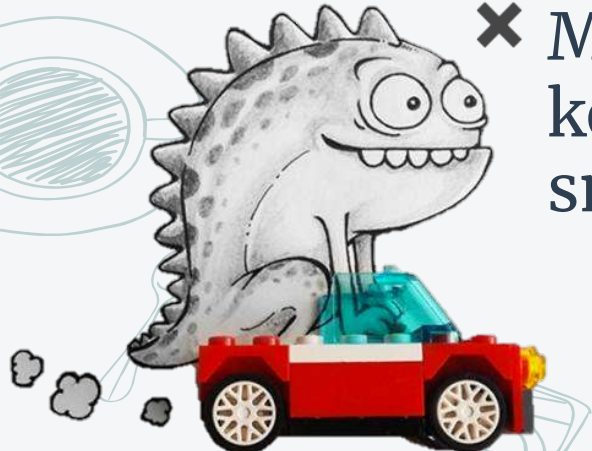
# ŠTO JE ENKAPSULACIJA ?

skrivanje detalja klasa...



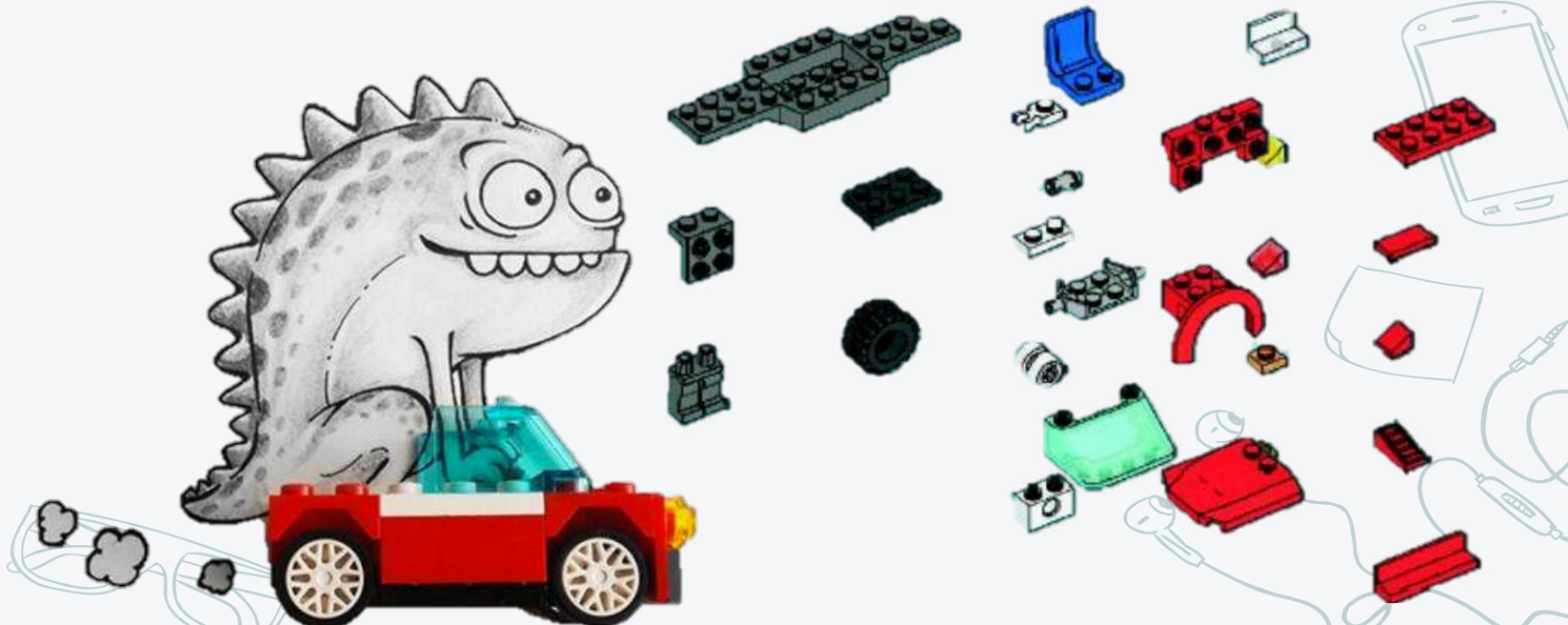
# POPULARAN PRIMJER ENKAPSULACIJE

- ✗ Često se kao primjer enkapsulacije uzima vožnja automobila.
- ✗ Trebate li znati kako funkcionira svaki dio (motor, karburator, alternator itd.)?
- ✗ Morate znati kako koristiti volan, kočnice, pedalu za gas, pokazivače smjera i ostalo važno za vožnju !



# ŠTO JE ENKAPSULACIJA ?

skrivanje nepotrebnih detalja klasa  
radi jasnijeg i jednostavnijeg korištenja





# POTREBA ZA ENKAPSULACIJOM

## JOŠ JEDAN PRIMJER

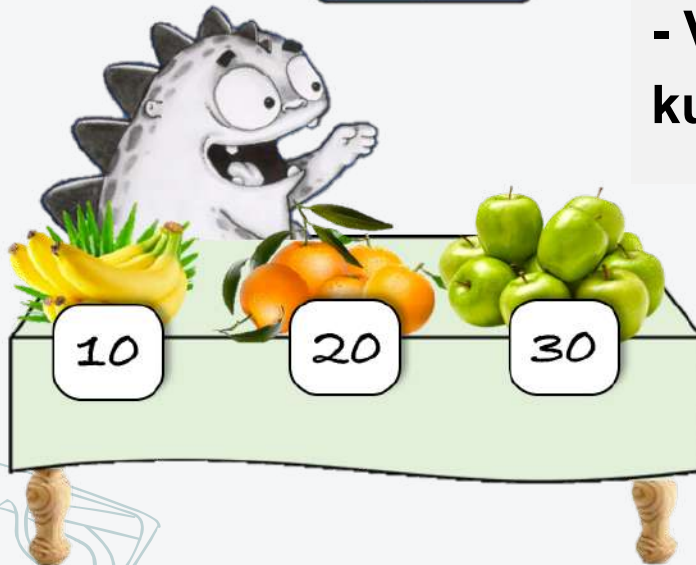
OBLIKOVANJE ŠTANDA ZA PRODAJU VOĆA





# PRODAJA VOĆA

Pitajte za  
pomoć

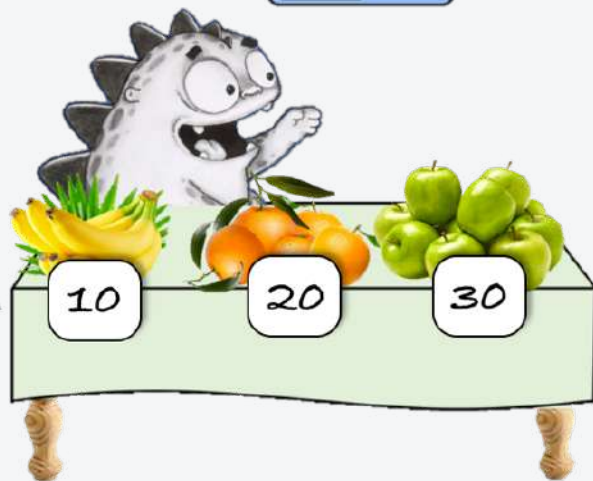


- Štand za voće oblikovan je kako bi poslužio sve kupce.
- Dizajn je takav da ne sprječava samoposluživanje.
- Voće je dostupno baš svim kupcima.





Pitajte za  
pomoć



# PRODAJA VOĆA

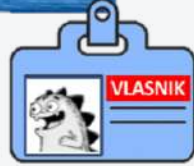
Kupac može obaviti kupnju,  
uzeti voće i platiti bez ikakve  
interakcije s vlasnikom.





# PRODAJA VOĆA

Pitajte za  
pomoć



Vlasnik je uveo mogućnost  
**REZERVACIJE** voća preko telefona.  
Voće koje ste rezervirali odvojit će  
sa strane.

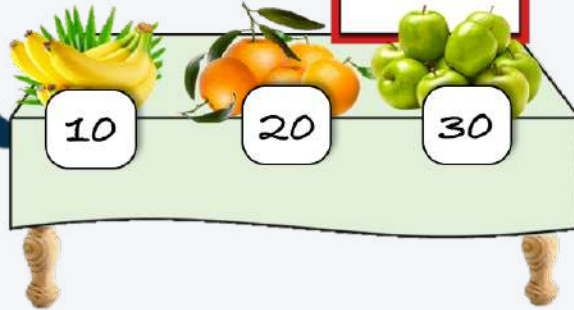
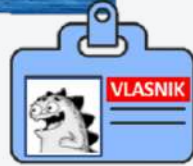


**REZERVIRANO!**



# PRODAJA VOĆA

Pitajte za  
pomoć



Temeljem zaprimljenog  
zahtjeva kupca, vlasnik je  
rezervirao voće.

Vlasnik smatra kako je zaštitio voće  
od kupaca koji nisu izvršili rezervaciju



# PRODAJA VOĆA

Pitajte za  
pomoć



Ipak zbog dizajna štanda, kupac je mogao odabrati i odnijeti voće koje je bilo rezervirano!



# PRODAJA VOĆA

**Vlasnik, nezadovoljan ponašanjem kupaca uvodi dodatne mjere zaštite svog štanda...**







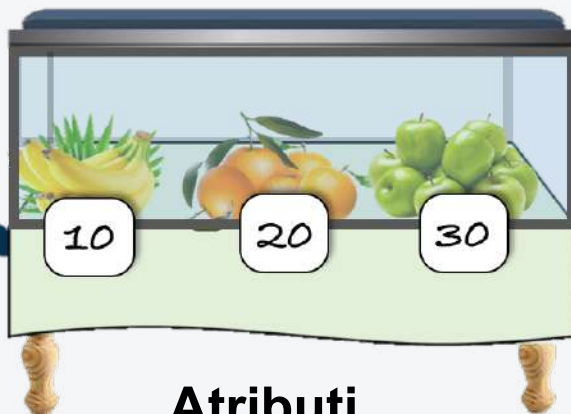
# PRODAJA VOĆA

Kako je zaštitio voće stavivši ga iza stakla, sada kontrolira svaki pristup kupaca, kao pravi vlasnik. Vlasnik je ENKAPSULIRAO PROCES kupovine voća.



# ENKAPSULACIJA

Pitajte za  
pomoć



Atributi



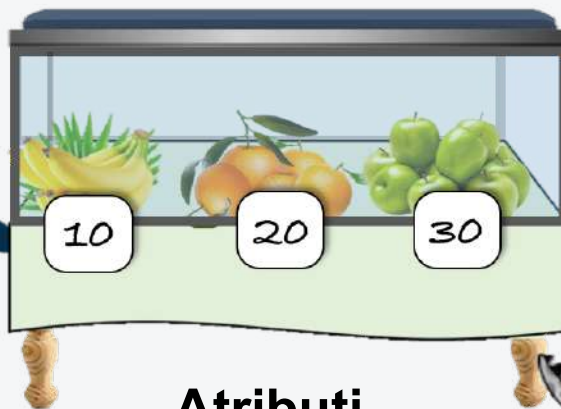
Vlasnika se pita za promjenu atributa.



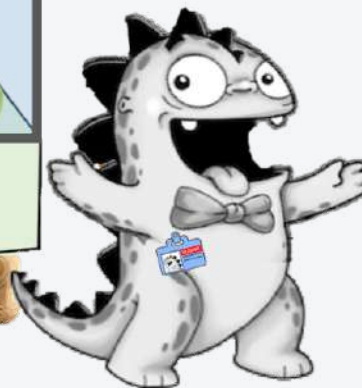
Ostale metode  
i programi

# ENKAPSULACIJA

Pitajte za  
pomoć



Atributi



Enkapsulacija predstavlja zaštitu atributa od arbitrarnog pristupa ostalih metoda i programa. Vlasnik kontrolira način pristupa atributima i sretan svojim štandom za voće.



Ostale metode  
i programi



## POGLEDAJMO KOD

*// Klasa, konstruktor i metoda*

```
class User {  
  constructor(name, email, age) {  
    this.name = name;  
    this.email = email;  
    this.age = age;  
  }  
  getage() {  
    return `Korisnik je star ${this.age} godina`;  
  }  
}  
  
const user1 = new User("Cody", "cody@mail.com", 2);  
const user2 = new User("Lia", "lia@gmail.com", 10);  
  
console.log(user1.name); // Cody  
console.log(user2.name); // Lia  
  
console.log(user1.getage()); // Korisnik je star 2 godina
```



```
// Klasa, konstruktor i metoda  
// private getteri
```

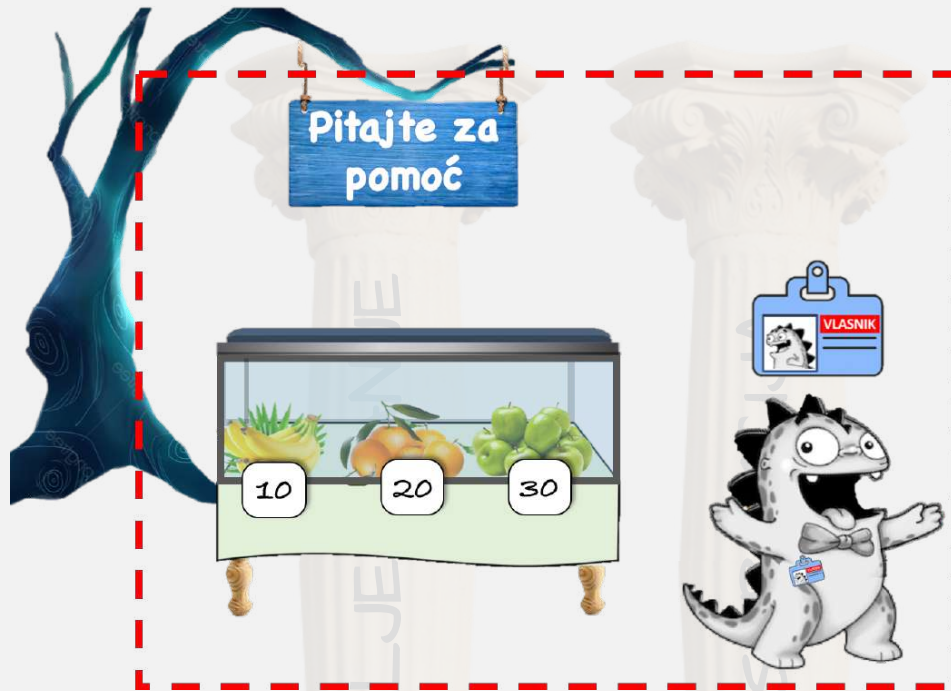
## POGLEDAJMO KOD

```
class Natjecatelj {  
  constructor(name, email, age) {  
    this._name = name;  
    this._email = email;  
    this._age = age;  
    this._skills = [];  
  }  
  
  learnSkill(skill) {  
    this._skills.push(skill);  
  }  
  
  isEligibleForCompetition() {  
    return this._age >= 2;  
  }  
  
  getAge() {  
    return `Natjecatelj je star ${this._age} godina`;  
  }  
}
```

```
const natjecatelj1 = new Natjecatelj("Cody",  
  "cody@mail.com", 2);  
const natjecatelj2 = new Natjecatelj("Lia", "lia@gmail.com",  
  10);  
  
natjecatelj1.learnSkill("agility");  
natjecatelj1.learnSkill("search and rescue");  
  
console.log(natjecatelj1._skills); // ['agility', 'search and  
  rescue']  
console.log(natjecatelj1.isEligibleForCompetition()); // true  
console.log(natjecatelj2._name); // odgovor je tu, ali znate  
  da ne bi smjeli  
console.log(natjecatelj2.getAge());
```

# ČETIRI STUPA OBJEKTNO ORIJENTIRANOG PROGRAMIRANJA

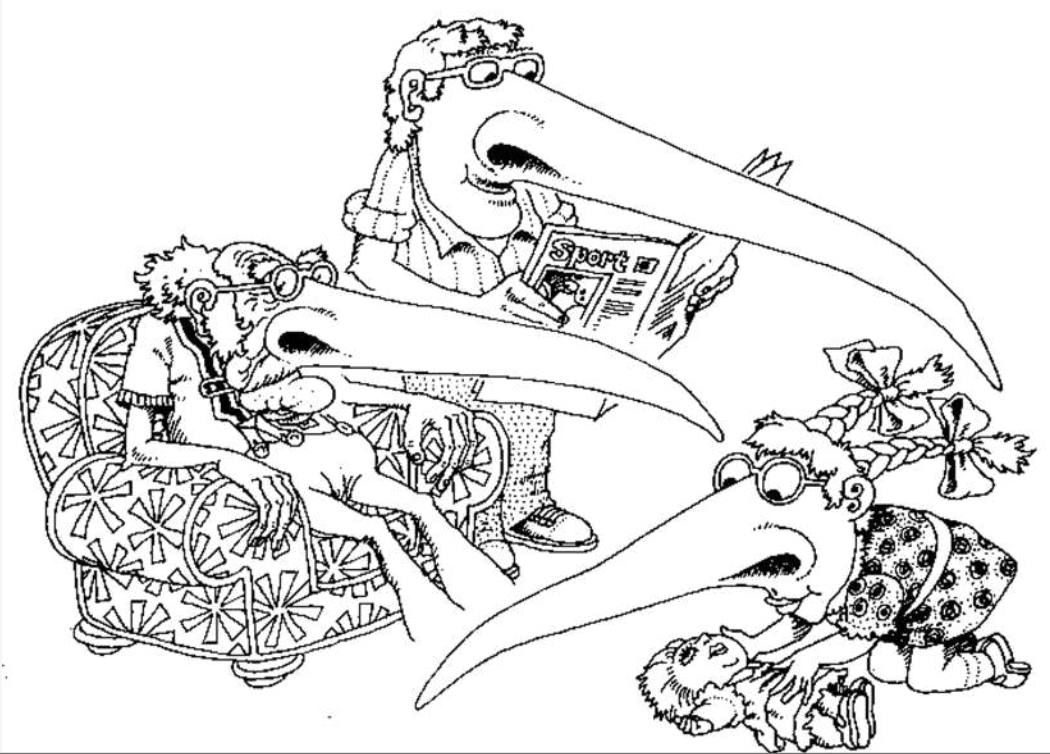
ENKAPSULACIJA



unutarnji ustroj nije vidljiv vanjskom svijetu  
vi imate kontrolu nad pristupom atributima

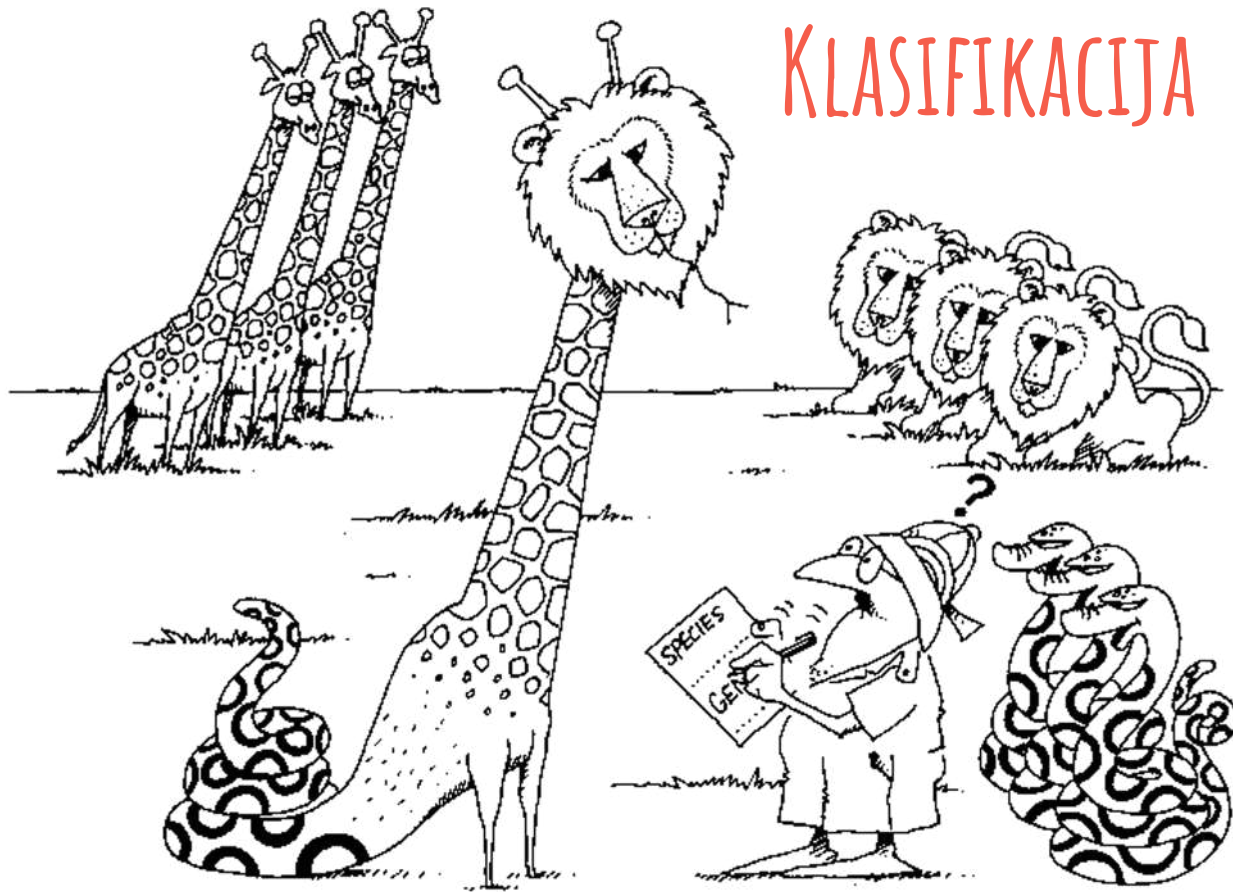


# NASLJEĐIVANJE



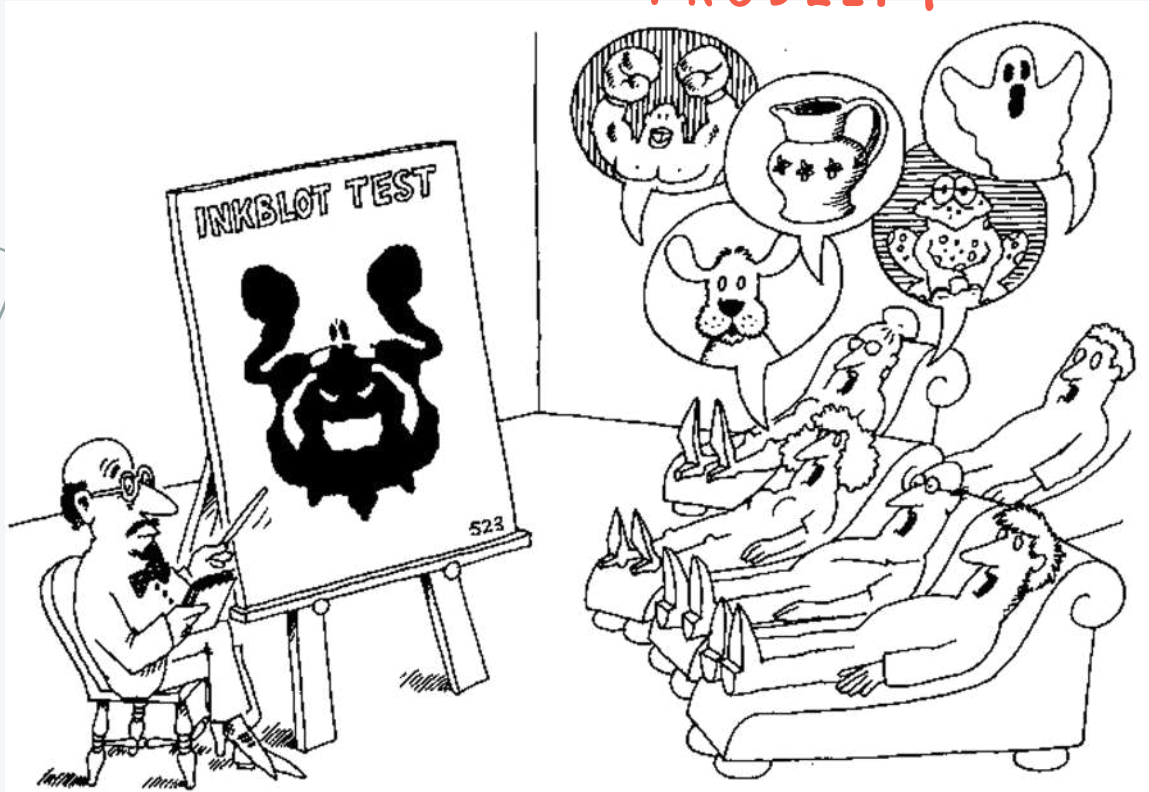
Izvedena klasa obično nasljeđuje strukturu  
i ponašanje bazne

# KLASIFIKACIJA



Klasifikacija je proces kojim uređujemo znanje

# PROBLEM



Veliki problem: različiti promatrači će klasificirati isti objekt na različite načine

# PONOVIMO!

Objektno orijentirano programiranje koristi objekte i njihove interakcije za modeliranje i izgradnju računalnih programa.

Tehnike objektno orijentiranog programiranja uključuju pojmove kao:

**ENKAPSULACIJA** – skrivanje nepotrebnih detalja klasa radi jasnijeg i jednostavnijeg rada

**NASLJEĐIVANJE** – omogućuje klasama nasljeđivanje polja i metoda od nadređenih klasa

**APSTRAKCIJA** – izdvajanje bitnih svojstava objekta, omogućuje nam rad s objektom bez da znamo njegov unutarnji izgled (npr. da bi koristili TV ne moramo znati što se točno unutra nalazi)

**POLIMORFIZAM** – različito ponašanje



# DO IDUĆI PUTA!

modelirajte i kodirajte

