Tips & tricks

Object oriented programming

POWERED BY:

nsoft sportradar

Programske paradigme (OOP) ?

Tri najvažnije programske paradigme danas:

- Proceduralna programska paradigma
- Objektno-orijentirana programska paradigma (OOP)
- Deklarativna programska paradigma

OOP

- objektno orijentirano programiranje
- označava vrstu pristupa programiranju
- interakcija objekata nasuprot akcijama koje se vrše nad podacima kod dosadašnjeg pristupa
- objekti mogu primati i obrađivati podatke, komunicirati međusobno
- svaki objekt je svijet za sebe

OOP - Glavna svojstva

➔ Enkapsulacija

Pošto ne postoje globalne varijable, iz drugih dijelova koda nije moguć pristup varijablama klase nikako osim ugrađenim metodama za njihovo čitanje i pisanje (ako smo ih deklarirali kao privatne, što je preporučeno). Na taj način se osigurava da objekt ne može doći u neko nepredviđeno stanje iz bilo kojeg razloga, npr. kad mu se pristupa istovremeno iz više dijelova koda (ako imamo višenitno programiranje) jer nužno je upotrebljavati funkcijske članove objekta u koje se mogu ugraditi sigurnosni mehanizmi poput sinkronizacije.

OPP - Glavna svojstva

➔ Naslijeđivanje

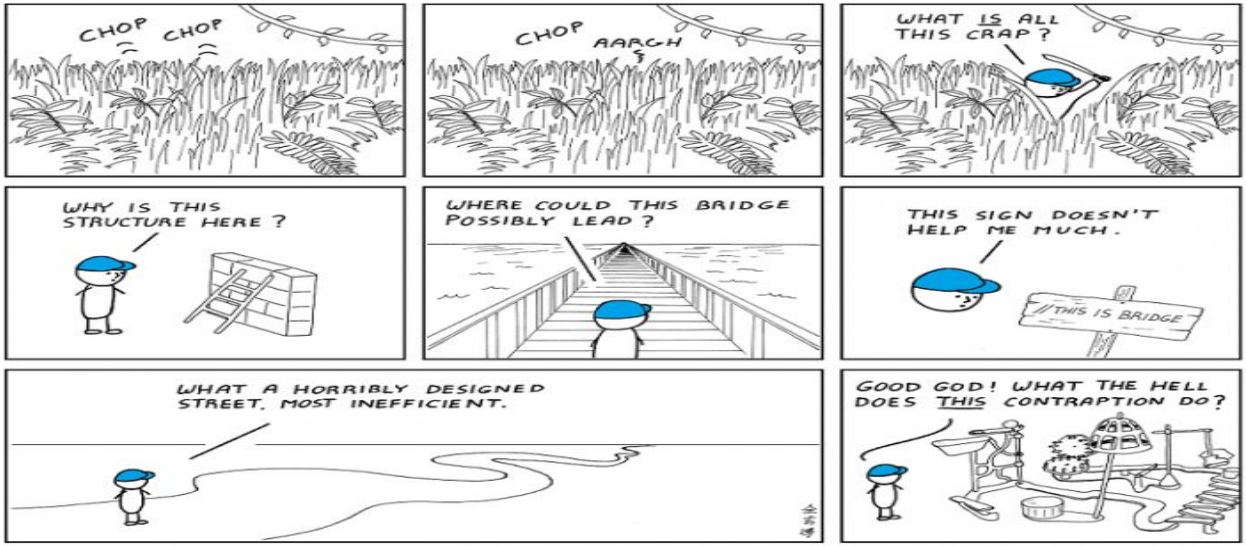
Kad već definiramo neki objekt, a zatreba nam neki sličan objekt koji je zapravo podskup početnog objekta, moguće je *naslijediti* početni objekt, čime štedimo vrijeme potrošeno za pisanje koda kao i diskovni prostor.

Zbog čega nam treba OOP ?

- Jedan od glavnih razloga:

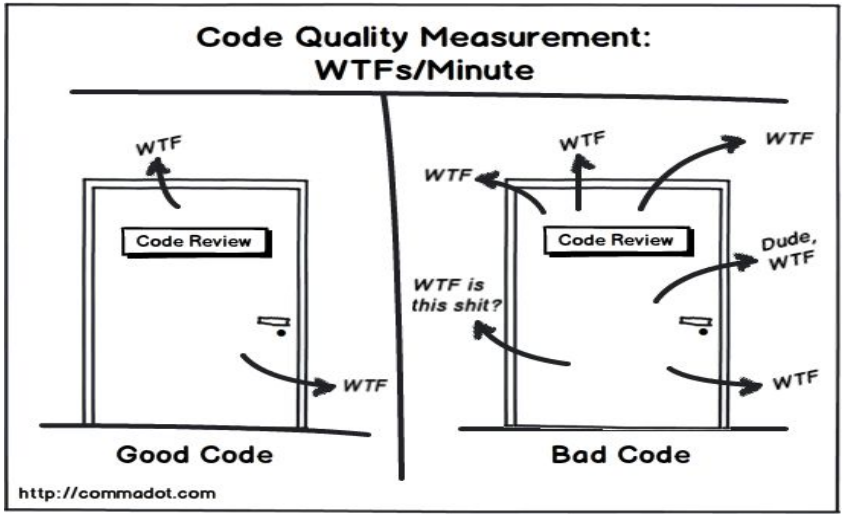
Smanjenje kompleksnosti razvoja i održavanja programskog koda

Zbog čega nam treba OOP ?



Zbog čega nam treba OOP ?

- Sustavnom primjenom objektno-orijentirane paradigme podiže se **kvaliteta programskog koda** kojeg programeri pišu



OOP

- dosad smo koristili objekte kako bismo povezali propertye u jednu cjelinu i mijenjali ih po potrebi
- u oop, objekti su cjeline kojemu se pristupa kroz definirano sučelje
- izbjegava se korištenje globalnih varijabli i funkcija izvan objekata

Constructor

- pomoću constructora kreiramo nove objekte
- pozivamo ga s *new* operatorom
- postavlja propertye u novom objektu

Constructor

- kad se funkcije pozivaju s *new*, vraćaju objekt s postavljenim parametrima *this*
- imaju property *constructor* koji je funkcija koja je kreirala objekt
- *constructor* je dio *prototypea* objekta
- svaki objekt ima prototype prema kojemu se kreira i koji mu daje propertye

Constructor

- definirajte constructor Cd tako da ima propertye title, artist, te metode *setDuration* i *getDuration* koja će postaviti odnosno vratiti dužinu trajanja zapisa. Title i artist potrebno proslijediti prilikom instanciranja.
- instancirajte dva cd-a i postavite duration

Prototype

- {} => new Object()
- kad dohvaćamo property objekta, prvo se gleda postoji li u samom objektu, zatim u prototypeu pa u prototypeu prototypea sve dok ga ne nađe
- kad postavljamo, postavlja se uvijek samo u objekt
- možemo bilo kad dodavati objekte i metode svim objektima baziranim na nekom prototypeu, nešto što im je zajedničko

Prototype

```
var cd= new Cd('Empire Burlesque', 'Bob Dylan');

Cd.prototype.price = 10;
console.log(cd.price); // 10

cd.price = 15;
console.log(cd.price); // 15
```

Zadatak

- definirajte constructor Auto tako da ima propertye naziv, boja, kilometraža, prosječna potrošnja i metodu *vozi* koja će povećati broj kilometara za 5
- instancirajte dva automobila i provozajte ih
- dodajte trenutnu količinu goriva property automobilima i postavite ju kod oba automobila
- dodajte metodu koja će smanjivati količinu goriva za prosječnu potrošnju i smanjite količinu goriva u automobilima

Zadatak

- napravite klasu O*soba* tako da ima ime, godine
- proširite mu metodu u kojoj ispisuje uobičajeno vrijeme buđenja - 8:00
- proširite mu metodu u kojoj ispisuje svoje ime
- instancirajte jednu osobu i ispišite joj vrijeme buđenja
- napravite klasu student tako da ima ime, godine i da može ispisati vrijeme buđenja
- neka vrijeme buđenja kod studenta bude 10:00
- instancirajte studenta i ispišite mu vrijeme buđenja i ime

Zadatak

- napravite klasu Auto tako da ima naziv, boju i prosječnu potrošnju proširite mu metodu u kojoj povećava prosječnu potrošnju za 5
- instancirajte novi auto i povećajte mu potrošnju
- napravite klasu TrkaciAuto tako da ima naziv, boju i prosječnu potrošnju
- povećajte prosječnu potrošnju za 30
- instancirajte novi trkaći auto i ispišite mu prosječnu potrošnju

POWERED BY: nsoft sportradar

SPARK. SCHOOL

THANK YOU

for attention