

# SOLID PRINCIPI

## MOVIEHUB

### **S PRINCIP (Single Responsibility Princip):**

Vidimo da svaka klasa ima svoju odgovornost, što nam se zahtjeva u ovom principu. Jedino nam problematična može biti klasa MovieHub, međutim ona u našem slučaju služi kao kontejnerska klasa, te je njen zadatak da upravlja sistemom, tako da u konačnici možemo reći da je S princip zadovoljen.

### **O PRINCIP (Open/Closed Princip):**

Što se tiče Open/Closed principa, on nam govori da svaka klasa treba biti otvorena za nadogradnje, ali zatvorena za modifikacije. U našem slučaju, većina klasa posjeduje samo getere i setere, tako da ne bi trebalo predstavljati problem pravljenje bilo kakvih nadogradnji, a potrebe za modifikacijama ne bi trebalo biti. MovieHub koja je služi kao kontejnerska klasa ima kao attribute ostale klase, tako da ne dolazi do modifikacije tih klasa. Mogućnost za nadogradnje se ogleda u tome što ako dodamo neki novi tip korisnika u budućnosti(npr. Gost, VIP, itd.), nema potreba za ikakvim modifikacijama, jer MovieHub klasa sadrži listu Korisnika, koja je apstraktna klasa.

### **L PRINCIP (Liskov Substitution Princip):**

Ovaj princip tvrdi kako svaka osnovna klasa treba biti zamjenjiva svim svojim podtipovima bez da utječe na ispravnost rada programa. U našem slučaju imamo nasljeđivanje iz klase Korisnik klase RegistrovaniKorisnik i Administrator. Vidimo da je moguće zamjeniti klase RegistrovaniKorisnik i Adminstrator sa klasom Korisnik, pa je samim tim i ovaj princip zadovoljen.

### **I PRINCIP (Interface Segregation Princip):**

Nemamo trenutno interfejsa, tako da je ovaj princip ispunjen.

### **D PRINCIP (Dependany Inversion Princip):**

Sistem klasa i njegovo funkcionisanje treba ovisiti o apstrakcijama, a ne o konkretnim implementacijama, a u našem vidimo da nam je bazna klasa Korisnik apstraktna, tako da zaključujemo da je i ovaj princip zadovoljen.