

SOLID PRINCIPI

Single responsibility Principle – Svaka klasa treba imati samo jednu odgovornost

- Ovaj princip je ispoštovan jer svaka klasa radi ono za šta je i namijenjena, te ima samo gettere i settere ili eventualno radi sa listama koje ima kao attribute. Jedina klasa koja bi potencijalno mogla kršiti ovaj princip je CineMagic.

Open/Closed Principle – Klasa treba biti otvorena za nadogradnje, ali zatvorena za modifikacije

- Ovaj princip je ispoštovan jer sve klase koje za attribute imaju druge klase, ne vrše modifikaciju istih već samo koriste njihove metode ili ih koriste kao attribute u nekoj listi.

Liskov Substitution Principle – Svaka osnovna klasa treba biti zamjenjiva svim svojim podtipovima bez da to utječe na ispravnost rada programa

- Ovaj princip je ispoštovan jer gdje god koristimo klasu Korisnik sistema možemo koristiti i klasu Administrator ili Korisnik sa nalogom, koje su izvedene iz bazne klase Korisnik sistema.

Interface Segregation Principle – Bolje je imati više specifičnih interfejsa, nego jedan generalizovani

- S obzirom da smo koristili interfejse *IHashirajPodatke*, *IPrototip*, *Iterator* i *IKreatorIterator*, ovaj princip je ispoštovan. Ovi interfejsi su iskorišteni u patternima, te nam olakšavaju implemntaciju određenih funkcionalnosti.

Dependency Inversion Principle – Sistem klasa i njegovo funkcionisanje treba ovisiti o apstrakcijama, a ne o konkretnim implementacijama

- Ovaj princip je ispoštovan jer je bazna klasa Korisnik ujedno i apstraktna klasa iz koje su naslijeđene klase Administrator i KorisnikSaNalogom.