

SOLID PRINCIPI:

Single Responsibility Princip:

- Svaka klasa treba da ima samo jednu ulogu. Dodavanje i modifikacija objekata od kojih se neka klasa sastoji smatra se jednim tipom funkcionalnosti, jer je takvo nešto neophodno kako bi klasa ispravno radila. S druge strane, interakcija s vanjskim objektom predstavlja potpuno drukčiji tip akcije. Za svaku klasu posebno razmotrit ćemo zadovoljavanje ovog principa:
- **Korisnik**
 - Dodavanje objekata (DodajNaListuŽelja, KupiFilm IzbrišiSaListeŽelja);
 - Modifikaciju objekata (PromijeniKorisnickoIme, PromijeniSifru, OcijeniFilm, KomentarisFilm);
 - Interakciju s vanjskim objektom (-).
- ✓ Ova klasa nema interakciju sa vanjskim objektom, pa ćemo je zadovoljen Single Responsibility Princip.
- **Videoteka**
 - Dodavanje objekata (DodajKorisnika, DodajAdministradora, DodajFilm);
 - Modifikaciju objekata (PrepraviOcjenuFilma, IzbrisiKorisnika, IzbrisiAdministradora, IzbrisiFilm, SortirajPoOcjeni, SortirajPoCijeni);
 - Interakciju s vanjskim objektom (NadjiKorisnika, NadjiFilm, NadjiGlumca).
- ✓ Ova klasa ima Interakciju sa vanjskim objektom, pa ćemo ovdje morati realizovati interfejs, koji ćemo nazvati **Pretraga**, i u njega smjestiti ovo tri metode: NadjiKorisnika, NadjiFilm, NadjiGlumca.
- **Administrator**
 - Što se tiče klase “Administrator” ima samo jednu odgovornost pa je zadovoljen princip Single Responsibility.
- **Film**
 - Klasa “Film” zadovoljava ovaj princip jer ima samo jednu odgovornost pa je zadovoljen princip Single Responsibility.

Open/Closed Princip:

- Svaka klasa mora biti otvorena za nadogradnju, a zatvorena za modifikaciju. Što se tiče narušavanja ovog principa, većina klasa ima samo getere i setere te pri uvođenju novih funkcionalnosti nema potrebe za bilo kakvom modifikacijom, a s obzirom da klasa Biblioteka kao atribut ima većinu ovih klasa, pa samim tim i ona neće dovesti do modifikacije istih.

Liskov Substitution Princip:

- Svaka osnovna klasa treba biti zamjenjiva svim svojim podtipovima bez da to utječe na ispravnost rada programa.
- Iz klase Osoba se nasljeduju klase Administrator i Korisnik. Iz našeg primjera vidimo da je ovaj princip zadovoljen jer ni na jednom mjestu se ne koristi klasa Osoba, nego samo klase Administrator i Korisnik.

Interface Segregation Princip:

- Bolje je imati više specifičnih interfejsa, nego jedan generalizovani, što podrazumijeva da svaki interfejs zadovoljava princip S. U našem sistemu imamo interfejse “Izmijeni”, “Pretrazi”.
- Što se tiče interfejsa “Pretrazi” vidimo odmah da interfejs zadovoljava princip S, to jest vrši se pretraživanje filmova ili korisnika tako da naš primjer zadovoljava ovaj princip.

Dependancy Inversion Princip:

- *Sistem klasa i njegovo funkcionisanje treba ovisiti o apstrakcijama, a ne o konkretnim implementacijama.*
- *U sistemu imamo baznu klasu, klasu Osoba, i ona je apstraktna te ne narušava ovaj princip.*
- *Pored nadljeđivanja vidimo da postoji još klasa koje su osnovne. Primjer za to je recimo klasa "Videoteka" i "Film". Izmjena filmova neće direktno uticati na klasu "Videoteka". Nju "ne zanima" da li se npr. cijena, glumci filma mijenjali. Na osnovu iznijetih činjenica zaključujemo da je ovaj princip zadovoljen.*