

SOLID PRINCIPI

- **Single Responsibility Princip:** „Klasa bi trebala imati samo jedan razlog za promjenu.“
Klasa „Skola“ je jedina klasa čije metode bi eventulano pristupale bazi ili lokalnoj datoteci. Sve ostale imaju samo jednu odgovornost. Budući da je klasa „Skola“ svakako klasa koja je namijenjena da upravlja cijelim sistemom i ona zadovoljava ovaj princip.
- **Open/Closed Principle:** „Entiteti softvera (klase, moduli, funkcije) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije.“
Klasa „Skola“ sadrži druge klase kao attribute, ali metode ove klase će se koristiti samo za dodavanje objekata u listu što ne narušava princip.
Iz apstraktne klase „Osoba“ moguće je izvesti nove klase bez potrebe da se modificira već postojeća. Takođe ova klasa je otvorena za nadogradnju.
- **Liskov princip zamjene:** „Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.“
Klase „Nastavnik“, „Roditelj“, „Ucenik“ i „Administrator“ su nasljeđene iz klase „Osoba“. Klasu „Osoba“ možemo zamijeniti svim ovim klasama. Takođe, klasa „Razrednik“ je nasljeđena iz klase „Nastavnik“. Klasu „Nastavnik“ možemo zamijeniti klasom „Razrednik“. Ovaj princip je ispunjen.
- **Interface Segregation Principle:** „Klijenti ne treba da ovise o metodama koje neće upotrebljavati.“
Budući da u našem sistemu još uvijek nema interfejsa, ne dolazi do narušavanja ovog principa.
- **Dependency Inversion Principle:** „Moduli ne bi trebali ovisiti od detalja. Detalji bi trebali ovisiti od apstrakcija.“
Klasa „Razrednik“ nasljeđuje klasu „Nastavnik“, koja nije apstraktna. Kako bi zadovoljili ovaj uslov sve zajedničke attribute i metode smo izdvojili u novu klasu „Osoba“ koja nam je bazna klasa i koja je apstraktna.