

Single-responsibility princip zahtijeva da svaka klasa ima svoju odgovornost.

Naše klase sadržavaju konstruktore, gettere, settere i atributi se mogu samo mijenjati unutar te date klase tako da princip je zadovoljen.

Open closed principle kaže da klase trebaju biti otvorene za nadogradnju bez modifikovanja klase.

U našem slučaju klase su jednostavne. Ako promijenimo jednu klasu, ne može doći do promjene druge klase, tako da princip je zadovoljen.

Liskov princip kaže da izvedene klase možemo zamijeniti baznim, a da se rezultat ne promijeni.

Ovaj princip nije zadovoljen. Imamo baznu klasu Osoba i tri izvedene, i korisnik ima svoje metode koje klasa "osoba" nema, tako da objekat tipa 'Osoba' ne može zamijeniti ni jednu izvedenu.

Princip izoliranja interfejsa

Ovaj princip se koristi kada imamo ogromne klase, pa dosta metoda nisu upotrebljive i u tom slučaju pravi se interfejs.

Naša aplikacija je jednostavna i nema ni jednog interfejsa i klase nisu kompleksne.

Nema nepotrebnih metoda ni u jednoj klasi.

Princip inverzije ovisnosti

Ovaj princip nalaze da ne treba ovisiti o izvedenim tj. konkretnim klasama, već o apstrakcijama jer se one rijetko mijenjaju. Princip je zadovoljen, jer sve izvedene klase zavise od bazne, a nema ni jedan slučaj gdje postoji zavisnost od izvedenih, tj. konkretnih klasa.