

SOLID principi

Single Responsibility Princip

Klase treba da znaju samo o jednoj stvari. One trebaju imati pojedinačnu odgovornost.

U našem sistemu, svaka klasa ima samo jednu odgovornost.

Jedini potencijalni problem je klasa Beauty centar, ali ona upravlja cijelim sistemom .

Open-Close Principle

Entiteti softvera trebali biti otvoreni za nadogradnju, ali zatvoreni za modifikacije. Iz postojećih apstraktnih klasa mogu se izvoditi nove klase, bez da se išta mijenja u njima, tj. nasljeđivanjem iz klasa Forma, Korisnik i Zaposlenik, nove klase će koristiti već postojeće apstrakcije.

Liskov Substitution Principle

Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.

Lični podaci koji se nalaze u klasi Zaposlenik se odnose na sve izvedene klase. Isto vrijedi i za klasu Korisnik. Zaključujemo da je i ovaj princip ispunjen.

Interface Agregate Principle

U našem sistemu nema interfejsa tako da je ovaj princip ispunjen.

Dependency Inversion Principle

Ne treba postojati ovisnost od konkretnih klasa. Prilikom nasljeđivanje treba razmotriti slučaj da je osnovna klasa apstraktna.

Ovaj princip je također ispunjen. Iz apstraktne klase Zaposlenik su izvedene klase Recepcioner, Administrator i Kozmetičar, a iz apstraktne klase Korisnik su izvedene klase Registrovani i Neregistrovani korisnik.