SOLID principi

Single Responsibility Principle - Princip pojedinačne odgovornosti
Ovaj princip glasi: *Klasa bi trebala imati samo jedan razlog za promjenu*.
U našem dijagramu klasa, ovaj princip je ispunjen jer sve klase upravljaju isključivo nad svojim atributima. Npr. klasa Korisnik, koja ima atribute poput *ime, prezime, korisničko ime...* ima samo jednu odgovornost - predstavljanje podataka o korisniku.

Open Closed Principle - Otvoreno zatvoreni princip
Ovaj princip glasi: *Entiteti softvera (klase, moduli, funkcije) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije*.
Ovaj princip je ispunjen jer u našem dijagramu klasa kao atribute često imamo objekte drugih klasa tako da promjena u jednoj klasi ne znači i promjenu u ostalim klasama. Na primjer, ako bismo u klasi Osoblje dodali novu ulogu, ne bismo imali razloga za mijenjanje klase Osoblje.

Liskov Substitution Principle - Liskov princip zamjene
Ovaj princip glasi: *Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima*.
Naš dijagram klasa posjeduje apstraktnu klasu Proizvod iz koje su izvedene Lijek i MedicinskaPomagala. Klasa Narudžba kao atribut ima listu apstraktne klase Proizvod, koja može biti ispunjena i objektima tipa Lijek i objektima tipa MedicinskaPomagala.

Interface Segregation Principle - Princip izoliranja interfejsa
Ovaj princip glasi: *Klijenti ne treba da ovise o metodama koje neće upotrebljavati*.
Ovaj princip je zadovoljen iz razloga što u našem dijagramu klasa ne postoji niti jedan interfejs.

Dependency Inversion Principle - Princip inverzije ovisnosti
Ovaj princip glasi:

a) Moduli visokog nivoa ne bi trebali ovisiti od modula niskog nivoa, oba bi trebalo da ovise od apstrakcija,

b) Moduli ne bi trebali ovisiti od detalja. Detalji bi trebali biti ovisni od apstrakcija.

Na osnovu samih zahtjeva ovog principa, odnosno da ne treba biti ovisnosti od konkretnih klasa, te prilikom naslijeđivanja treba razmatrati slučajeve da je osnovna klasa apstraktna, mi smo ovo primijenili u klasi Komentar gdje kao atribut koristimo apstraktnu klasu.