

1.PRINCIP POJEDINAČNE ODGOVORNOSTI

Klasa bi trebala imati samo jedan razlog za promjenu.

U našem dijagramu klasa, svaka klasa se bavi stvarima koje se tiču isključivo nje. Tako klasa Korisnik sadrži kao attribute osnovne informacije o korisniku, a klasa Pas sadrži lične podatke ljubimca. SOLID princip. Slično, da je ovaj princip ispunjen, možemo vidjeti i na osnovu klasa Dokument I Komentar. Da smo klasi Komentar pripisali i odgovornost za Dokumente, to bi činilo klasu Komentar kompleksnom.

2.OPEN CLOSED PRINCIPLE -OTVORENO ZATVOREN PRINCIP

Entiteti softvera (klase, moduli, funkcije) bi trebali biti otvoreni za nadogradnju, ali zatvoreni za modifikacije.

Da bi bio ispoštovan princip klase su otvorene za proširenje, stvaranje podklase, dodavanje novih atributa i metoda, ali zatvorena za modifikacije. Ne bi se smjelo desiti da nadogradnja klase zahtijeva modifikaciju neke druge klase. Upravo je to ispunjeno u našem dijagramu klasa. U našem dijagramu klasa postoje veze agregacije, što znači da postoje klase koje sadrže objekte ili kolekcije drugih klasa, tako da izmjena u nekoj klasi neće implicirati promjenu druge klase. Na primjer, ukoliko se u sistemu odlučimo omogućiti još jedan način održavanja instrukcije (na primjer: reakcije na komentare), to ni na koji način neće zahtijevati modifikaciju klasa, nego ćemo samo postojeći sistem dodatno nadograditi.

3. LISKOV SUBSTITUTION PRINCIPLE

Liskov princip zamjene glasi:

Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima

Naš dijagram klase ne posjeduje apstraktne klase pa ne postoje podtipovi i samim tim je ispunjen Liskov princip

4. INTERFACE SEGREGATION PRINCIPLE -PRINCIP IZOLIRANJA INTERFEJSA

Princip izoliranja interfejsa:

Klijenti ne treba da ovise o metodama koje neće upotrebljavati

U našem sistemu ne postoje interfejs ne posjedujemo klase sa prevelikim brojem metoda, tako da je ovaj princip zadovoljen.

5. DEPENDENCY INVERSION PRINCIPLE -PRINCIP INVERZIJE OVISNOSTI

Princip inverzije ovisnosti:

Moduli visokog nivoa ne bi trebali ovisiti od modula niskog nivoa.oba bi trebalo da ovise od apstrakcija

Moduli ne bi trebali ovisiti od detalja. detalji bi trebali biti ovisni od apstrakcija.

Kako i u 3. Principu nemamo apstraktizaciju tj. apstraktne klase smatramo princip inverzije ovisnosti ispunjenim.