S princip

Klase treba da znaju samo o jednoj stvari. One trebaju imati pojedinačnu odgovornost. Treba postojati samo jedan razlog za promjenu klase.

- Svaka klasa je napravljena samo zbog jednog razloga, kako bi obavljala samo jedan zadatak, odnosno služila jednom cilju. Klasa Zaposlenik je klasa iz koje su izvedene klase Doktor, Direktor i MedSestra i ona se brine o osnovnim podacima koji se tiču zaposlenika oridinacije i njihovim izmjenama. Izvedene klase se brinu o dodavanju i modifikaciji objekata od kojih se klase sastoje. Klasa Pregled se brine o svojim atributima, njihovim modificiranjem i dodavanjem. Klasa SpecDoktor je klasa koja služi da detaljnije specificira doktora. Karton je klasa koja čuva sve informacije o pregledima jednog pacijenta, ona povezuje klase Pregled i Pacijent. Zubex je klasa koja drži sistem, ona pruža mogućnost pregleda i ažuriranja korisnika sistema. Za sve funkcionalnosti za koje smo smatrali da su nespojive smo pravili posebno klase. Mislimo da smo sve funkcionalnosti pokrili odgovarajućim klasama.

O princip

Trebamo biti sposobni mijenjati okruženje oko modula bez promjene samog modula.

- Dodavanje novih metoda neće zahtijevati uređivanje već postojeće klase i njenih atributa. Klase koriste druge klase, ali u većini metoda koriste se postupci za ubacivanje objekata u Array i ArrayList, što je generička operacija. Dešavat će se samo nadogradnja i ponuda novih mogućnosti.

L princip

Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.

- Na dijagramu vidimo jednu apstraktnu klasu Zaposlenik i jednu baznu, ali ne apstraktnu klasu Doktor. Na svim mjestima, gdje se koristi osnovni objekat se može koristiti i izvedeni i pritom će to imati smisla.

I princip

Klijenti ne treba da ovise o metodama koje neće upotrebljavati.

- Korisnik nema veliki izbor metoda na raspolaganju i one su prilično dozirane njegovim potrebama. Sistem ne posjeduje “debele” klase. Tako da nije potrebno korisnika štiti od metoda koje im ne trebaju i od poznavanja implementacije objekta kojeg koristi. Kako nemamo niti jedan interfejs u sistemu (za sada, no ne znači da u daljnjem razvoju neće doći do potrebe za nekim) ovaj princip je po našem mišljenju zadovoljen.

D princip

Princip inverzije ovisnosti. Ne treba ovisiti od konkretnih klasa. Prilikom nasljeđivanja treba razmotriti slučaj da je osnovna klasa apstraktna.

- Ovaj princip zahtijeva da pri naslijeđivanju od strane više klasa bazna klasa bude uvijek apstraktna,što je kod nas i slučaj klasa Zaposlenik je apstraktna klasa. Naš dijagram klasa posjeduje i klasu Doktor koja je bazna klasa, ali nije apstraktna, klasi SpecDoktor, tu bi moglo doći do narušavanja ovoga principa, ali pošto je naslijeđivanje od strane samo jedne klase, mislim da možemo reći i da je ovaj princip zadovoljen.