SOLID PRINCIPI

SINGLE RESPONSIBILITY PRINCIPLE - PRINCIP POJEDINAČNE ODGOVORNOSTI

Princip pojedinačne odgovornosti glasi: KLASA BI TREBALA IMATI SAMO JEDAN RAZLOG ZA PROMJENU.

Glavni cilj ovog principa jeste smanjenje kompleksnosti. Klase trebaju imati pojedinačnu odgovornost.U našem dijagramu klasa možemo primijetiti da se svaka klasa bavi isključivo stvarima koje se tiču direktno nje. Tako klasa Korisnik sadrži kao atribute lične podatke koje bi svaki korisnik trebao imati.

Također, klase Trener, Grupa, PlanTreninga, ProgramVjezbanja, PrirucnikZaSuplementaciju i Administrator sadrže osnovne informacije koje se tiču njih.

OPEN CLOSED PRINCIPLE - OTVORENO ZATVOREN PRINCIP

Otvoreno zatvoren princip glasi: ENTITETI SOFTVERA (KLASE, MODULI, FUNKCIJE) BI TREBALI BITI OTVORENI ZA NADOGRADNJU, ALI ZATVORENI ZA MODIFIKACIJE.

U našem dijagramu klasa dodavanje metoda i novih atributa postojećim klasama neće uzrokovati promjenu na trenutnim stanjem klasa, što se naravno ne bi smjelo desiti. Dakle, klase su otvorene u smislu da ih možemo proširiti, dodati nove atribute i metode, stvoriti podklase. Svaka klasa treba istovremeno biti otvorena za nadogradnju, ali zatvorena za modifikacije. U našem dijagramu klasa najzastupljenija je veza agregacije, što znači da većina klasa sadrži objekte ili kolekcije drugih klasa, tako da izmjena u nekoj klasi neće prouzrokovati promjenu druge klase. Na primjer, ako želimo dodati jos neku informaciju koju treba sadržavati Grupa, to neće modificirati sistem već će samo nadograditi tu klasu.

LISKOV SUBSTITUTION PRINCIPLE - LISKOV PRINCIP ZAMJENE

Liskov princip zamjene glasi: PODTIPOVI MORAJU BITI ZAMJENJIVI NJIHOVIM OSNOVNIM TIPOVIMA.

Naš dijagram klasa ne sadrži apstraktne klase pa tako ni podtipove, a samim tim je ispunjen i ovaj princip.

INTERFACE SEGREGATION PRINCIPLE - PRINCIP IZOLIRANJA INTERFEJSA

Princip izoliranja interfejsa glasi: KLIJENTI NE TREBA DA OVISE O METODAMA KOJE NEĆE UPOTREBLJAVATI.

U našem dijagramu klasa ne postoje interfejsi jer većina klasa sadrži samo gettere i settere, tj. nema klasa sa prevelikim brojem metoda, čime se može smatrati da je ovaj princip zadovoljen.

DEPENDENCY INVERSION PRINCIPLE - PRINCIP INVERZIJE OVISNOSTI

Princip inverzije ovisnosti glasi: 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.

Ovaj princip može se interpretirati da ne treba ovisiti od konkretnih klasa pri nasljeđivanju, ali budući da u našem sistemu nemamo nasljeđivanja, i ovaj princip se može smatrati zadovoljenim. Većina klasa u sistemu ovisi od klasa koje su tipovi podataka, kao što je prvenstveno List, koje se neće zasigurno mijenjati tako da i tu nije narušen ovaj princip.