SOLID principi

1. SINGLE RESPONSIBILITY PRINCIPLE

Cilj ovog principa je da se svaka klasa bavi samo onim što se tiče direktno nje. U našem slučaju to znači da se u klasi "User" nalaze stvari vezane samo za korisnika (ime, prezime, username, itd.). Ako pogledamo ostale klase možemo primjetiti da je to zadovoljeno, pošto svaka klasa opisuje samo ono što njen naziv znači. Neke od većih klasa kao "Courses" iako imaju puno atributa će se mijenjati samo ako nastane potreba za mijenjanjem kursa kao klase, a ne mijenjanjem neke od povezanih klasa. Npr. ako se klasa student promjeni, kursevi se ne mijenjanju. Isto važi za svaku klasu. Sve što bi se moglo opisati nekom posebnom klasom, a da ima smisla opisati, je tako i odrađeno. Time je i postignuto da princip bude ispunjen, odnosno da se klase mijenjanju samo u slučaju mijenjanja onoga što opisuju.

2. OPEN CLOSED PRINCIPLE

Dodavanje novih metoda ili atributa u klase ne smije dovesti do promjene bilo koje klase unutar sistema. Cilj ovog principa je postignut unutar svih klasa. U našem dijagramu klasa pojavljuju se veze i agregacije i kompozicije. Da bi princip bio ispunjen, mijenjanjem/nadogradnjom neke klase

ne smije dovesti do promjene klase koje je ona dio. Npr. klase “Course” i “CourseMaterial”. Ako bi se u “CourseMaterial” dodao neki atribut ili metoda, recimo autor, to neće dovesti do nikakve promjene u klasi “Course”. Takvo stanje se može primjetiti u svim klasama što ipmlicira da je princip zadovoljen.

3. LISKOV SUBSTITUTION PRINCIPLE

Ovaj princip govori da svaki podtip mora biti zamjenjiv njegovim osnovnim tipom. U našem dijagramu klasa imamo apstraktnu klasu User, koja sadrži osnovne informacije o korisniku, iz koje su naslijeđene klase Professor i Student. U ovom slučaju je zadovoljen Liskov princip zamjene jer svi podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima, što je u našem slučaju ispunjeno.

4. INTERFACE SEGREGATION PRINCIPLE

Ovaj princip zahtijeva da klijenti ne trebaju ovisiti o metodama koje neće koristiti. U našem slučaju ovaj princip je ispunjen jer sve klase obavljaju samo one aktivnosti koje su za njih vezane. Također da bi se koristila neka od klasa ne moraju se koristiti sve metode, tako da korisnik ima slobodu da ih koristi po želji bez njihovog uticanja na slobodu.

5. DEPENDENCY INVERSION PRINCIPLE

Ovaj princip zahtijeva da klase trebaju zavisiti o apstraktnim klasama. U našem slučaju ovaj princip je ispunjen tako što su klase Student i Professor izvedene iz osnovne klase User koja je apstraktna. Također klase ne bi treble zavisiti od detalja. To svojstvo je isto zadovoljeno, jer mijanjanjem bilo kojeg detalja unutar bilo koje klase koja je izvedena, ne smije mijenjati ništa unutar drugih naslijeđenih klasa, nego mijenjanjem abstraktne klase se dešava promjena i u naslijeđenim. Kako i je i ovo zadovoljeno, sam princip je zadovoljen.