

Univerzitet u Sarajevu

Elektrotehnički fakultet Sarajevo

Odsjek za računarstvo i informatiku

Objektno orijentirana analiza i dizajn

Laboratorijska vježba 7

Objašnjenje SOLID principa u Class Diagram-u

Studenti:

Elma Bejtović

Sara Makešoska-Džebo

Jusuf Delalić

SOLID je akronim koji se sastoji od početnih slova principa koji se koriste u objektno orijentisanom dizajnu. Mnogi stručnjaci smatraju da predstavljaju neizbježan dio razvoja kvalitetnog softvera, te da predstavljaju nešto što svako ko želi da se bavi razvojem softverskih projekata treba poštovati.

S – princip pojedinačne odgovornosti, Single Responsibility Principle - SRP, tvrdi da svaka klasa treba biti zadužena samo za jednu cjelinu unutar projekta, tj. klase trebaju imati pojedinačnu odgovornost.

Neki primjeri pojedinačnih odgovornosti u Class Diagram-u:

Klasa Administrator je zadužen za sve promjene koje se odnose na ponudu hotela, što uključuje ažuriranje, dodavanje i brisanje ponude.

Klasa Gost je zadužen za rezerviranje sobe/soba u hotelu, te potvrdu i eventualno otkazivanje. Također Gost može davati komentare i ocjene koje se odnose na ponudu.

O – otvoreno zatvoren princip, Open Closed Principle – OCP, tvrdi da trebamo biti sposobni mijenjati okruženje oko modula bez promjene samog modula.

Uzmimo za primjer apstraktnu klasu Osoba. Iz te klase postoje dvije izvedene klase - Administrator i Gost. Promjenom klase Administrator i/ili Gost, tj. dodavanjem atributa ili metoda koje dodatno definišu te klase mi ne moramo mijenjati klasu Osoba.

L – Liskov princip zamjene, Lisk Substitution Principle – LSP koji tvrdi da podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima. Liskov princip zamjene se krši svaki put kada pokušamo napraviti nelegalan poziv funkcije na izvedenoj klasi.

Iz klase Osoba postoje dvije izvedene klase - Administrator i Gost. Metode klase Osoba su takve da imaju smisla da se pozivaju i nad klasama Administrator i Gost jer predstavljaju samo get i set metode. Npr. getIme, getPrezime, getEmail, setBrojTelefona, setDatumRodjenja i sl.

I – princip izoliranja interfejsa (eng. Interface Segregation Principle-ISP) tvrdi da klijenti ne treba da ovise o metodama koje neće upotrebljavati. Korisnici se mogu zaštititi od metoda koje im ne trebaju, dajući im interfejse prema onim metodama koje su im potrebne. Ovo čuva korisnika od promjena metoda koje ga se ne tiču.

U Class Diagram-u postoji interfejs ponuda koji u sebi sadrži osnovne metode koje daju informacije o ponudi, tj. ne sadrže one metode koje korisniku ne trebaju.

D - princip inverzije ovisnosti (eng. Dependency Inversion Principle-DIP) se može jednostavno tumačiti da projekat ne treba ovisiti od konkretnih klasa. Apstraktne klase i interfejsi mnogo se manje mijenjaju nego njihove konkretne izvedenice. Stoga je bolje ovisiti o apstrakcijama nego o stvarnim klasama. Slijedeći ovaj princip smanjuje se utjecaj koji promjena može imati na sistem.

Interfejs Ponuda predstavlja jednu od osnovnih klasa i projekat u velikom dijelu ovisi o njemu.