



Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za računarstvo i informatiku



SOLID Principi ecoRide

Objektno orijentisana analiza i dizajn

Članovi grupe:

Nejra Stršević
Vildana Tabaković
Berina Zejnilović

Single Responsibility Principle (Princip pojedinačne odgovornosti)

Ovaj princip glasi:

Klasa bi trebala imati samo jedan razlog za promjenu.

U našem dijagramu klasa, ovaj princip je ispunjen jer sve klase upravljaju isključivo nad svojim objektima.

Npr, klasa *Rezervacija* sadrži osnovne informacije o rezervaciji, poput vozila, vremena i lokacije rezervacije. Pored toga, imamo dodatnu klasu *Vozilo*, koja u sebi sadrži cijenu karte. Ona bi se nerijetko mogla mijenjati, pa bi svaka njena izmjena povlačila izmjenu implementacija klase *Rezervacija*, što nastojimo izbjeći jer to nije primarna odgovornost pomenute klase.

Open Closed Principle (Otvoreno-zatvoreni princip)

Ovaj princip glasi:

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

Ovaj princip je ispunjen jer u našem dijagramu klasa, klase su povezane vezama agregacije i kompozicije, jer u klasama kao attribute često imamo objekte drugih klasa tako da promjena u jednoj klasi neće značiti promjenu u ostalim klasama. Npr, ako bismo u klasi *Korisnik* dodali novu vrstu pogodnosti, ne bismo imali razloga za mijenjanje klase *Korisnik*.

Liskov Substitution Principle (Liskov princip zamjene)

Ovaj princip glasi:

Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.

Imamo samo jedno nasljeđivanje koje ispunjava ovaj princip. Klase *Uposlenik* i *Korisnik* mogu se bez ikakvih problema iskoristiti u bilo kojem kontekstu u kojem se koristi klasa *Osoba*.

Interface Segregation Principle (Princip izoliranja interfejsa)

Ovaj princip glasi:

Klijenti ne treba da ovise o metodama koje neće upotrebljavati.

Ovaj princip je zadovoljen iz razloga što u našem sistemu ne postoji nijedan interfejs. Pored toga, naš sistem u sebi ne sadrži nijednu "debelu" klasu.

Dependency Inversion Principle (Princip izoliranja interfejsa)

Ovaj princip glasi:

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

b) Moduli ne bi trebali ovisiti od detalja. Detalji bi trebali biti ovisni od apstrakcija.

Ovaj princip je ispunjen zato što imamo klasu Osoba koja je proglašena apstraktnom klasom, te sve ostale klase naslijeđene iz nje se ne mogu koristiti samo sa svojim atributima. Primjer toga vidimo u klasama *Korisnik*, *Uposlenik*.