



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



# **SOLID principi**

Objektno orijentisana analiza i dizajn

Članovi tima:

Anida Nezović

Hana Biševac

Nada Alijagić

# 1. Single Responsibility Principle

*Klasa bi trebala imati samo jedan razlog za promjenu.*

U našem primjeru, ovaj princip je zadovoljen. Svaka klasa ima jedan tip zaduženja. Klasa *Atrakcija* čuva informacije poput naziva, lokacije, vodiča itd. Ona ne daje dodatni opis vodiča i lokacije, jer postoje klase *Vodic* i *Lokacija*, koje će detaljnije opisati ta dva objekta. Također, klasa *Tura* ne daje nikakav opis samog vodiča ili atrakcija koje pripadaju planu ture, jer za to postoje klase *Vodic* i *Atrakcija*.

# 2. Open Closed Principle

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

Ovaj princip je zadovoljen na našem dijagramu klasa, jer promjena u implementaciji bilo koje klase, ne forsira modifikaciju neke druge klase. Ako bismo izmijenili nešto u klasi *Atrakcija*, onda ne bismo morali mijenjati ništa u klasama *Tura* i *RegistrovaniKorisnik*, jer klase sadrže listu objekata tipa *Atrakcija*, te se promjena odrazi na svaki od njih.

# 3. Liskov Substitution Principle

*Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.*

Ovaj princip je zadovoljen na našem dijagramu, jer klase *RegistrovaniKorisnik*, *Vodič* i *Zaposlenik* mogu zamijeniti klasu *Korisnik* u svakom kontekstu. Atributi koje klasa *Korisnik* ima, čine sastavni dio samih klasa *RegistrovaniKorisnik*, *Vodič* i *Zaposlenik*, pa ne nailazimo na problem kršenja LSP.

## 4. Interface Segregation Principle

*Klijenti ne treba da ovise o metodama koje neće upotrebljavati.*

Kako smo dizajnirali model sistema, nismo koristili interfejs, te znamo da ovaj princip nije narušen.

## 5. Dependency Inversion Principle

*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, jer je osnovna klasa *Korisnik* apstraktna, a konkretne klase *RegistrovaniKorisnik*, *Vodič* i *Zaposlenik* su izvedenice.