

## Task 6

### Određivanje SOLID principa na primjeru class dijagrama

Bakir Bajrović

Stefani Kecman

Merjem Bećirović

#### **S princip:**

Napravili smo klase koje obavljaju samo jednu funkciju. Klasa BankovniRačun skladišti samo podatke o broju računa i iznosu na njemu. Klasa Osoba ima samo informacije o imenu i prezimenu osobe. Ekipa ima samo svoj naziv i bodove i sl.

#### **O princip:**

Ovaj princip je zadovoljen upotrebom apstrakcije. Mi u našem sistemu imamo apstraktnu klasu Osoba, koju nasljeđuju: RegistrovaniKorisnik, Administrator, Zaposlenik, PremiumKorisnik. Nadgradnja ovih klasa ne mijenjaju logiku apstraktne klase Osoba. Samim tim, omogućeno je dodavanje novih funkcionalnosti u izvedenim klasama, bez da se mijenja klasa Osoba koja ne bi trebala biti podložna tim promjenama.

#### **L princip:**

L princip kontroliše ispravnost nasljeđivanja. Mi smo iz klase Osoba izveli klase: RegistrovaniKorisnik, Administrator, Zaposlenik, a iz klase RegistrovaniKorisnik izveli klasu PremiumKorisnik. PremiumKorisnik jeste RegistrovaniKorisnik. RegistrovaniKorisnik, Administrator, Zaposlenik jesu Osoba, po svojim funkcionalnostima. U bilo kojem dijelu koda izvedene klase mogu zamijeniti klasu koju nasljeđuju.

#### **I princip:**

Napravili smo klase tako da one nemaju metode koje neće biti iskorištene i nismo imali potrebe za izdvajanjem interfejsa. Razlog je taj što su ovo samo podatkovne klase, a sve radnje će se odvijati u kontroleru.

**D princip:**

Kao što smo ranije naveli, RegistrovaniKorisnik, Administrator, Zaposlenik nasljeđuju klasu Osoba. Osoba je apstraktna podatkovna klasa i gotovo smo sigurni da u budućnosti neće trpiti promjene, pa samim tim neće utjecati ni na klase koje je nasljeđuju. S druge strane, PremiumKorisnik nasljeđuje klasu RegistrovaniKorisnik, koja nije apstraktna, ali zbog logike koja stoji iza ovog nasljeđivanja, sasvim je opravdano da se promjene u RegistrovaniKorisnik primjenjuju i na PremiumKorisnik klasu. Dakle, promjene nasljeđenih klasa neće utjecati na klasu koju nasljeđuju, dok imamo nasljeđivanje apstraktne klase (koja se neće nikada mijenjati) i nasljeđivanje konkretne klase, koje je argumentovano opravdano.