

SOLID principi

S princip: Napravili smo klase koje obavljaju samo jednu funkciju. Klasa Narudžba sadrži samo podatke o narudžbi (ocjena, iznos, datum itd.). Klasa Osoba ima samo informacije o imenu, prezimenu, šifri i emailu osobe. Popust ima samo kod, datum isteka, vrstu i datum početka i sl.

O princip: Ovaj princip je zadovoljen upotrebom apstrakcije. Mi u našem sistemu imamo apstraktnu klasu Osoba, koju nasljeđuju: Korisnik i Zaposlenik. 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 kontrolira ispravnost nasljeđivanja. Mi smo iz klase Osoba izveli klase: Korisnik i Zaposlenik. Korisnik i 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, Korisnik i 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. Dakle, promjene nasljeđenih klasa neće utjecati na klasu koju nasljeđuju, dok imamo nasljeđivanje apstraktne klase (koja se neće nikada mijenjati).