

SOLID PRINCIPI

S princip: Napravili smo klase koje obavljaju samo jednu funkciju. Klasa *Narudžba* se bavi isključivo čuvanjem podataka o narudžbama kao što su ocjena, iznos, datum itd. Klasa *Osoba* sadrži samo osnovne informacije o korisniku, poput imena, prezimena, šifre i email adrese. Klasa *Popust* sadrži podatke koji se odnose samo na popuste, kao što su kod, datumi početka i isteka, te tip popusta.

O princip: Ovaj princip smo ispoštovali korištenjem apstrakcije. Imamo apstraktnu klasu *Osoba*, koju nasljeđuju konkretne klase *Korisnik* i *Zaposlenik*. Ove izvedene klase proširuju funkcionalnosti bez izmjene osnovne logike apstraktne klase *Osoba*, što omogućava lako dodavanje novih funkcionalnosti bez narušavanja postojećeg koda.

L princip: Kod nasljeđivanja smo se pridržavali ovog principa. Iz klase *Osoba* su izvedene klase *Korisnik* i *Zaposlenik*, koje u potpunosti funkcionišu kao zamjene za baznu klasu. Gdje god se očekuje instanca *Osobe*, moguće je bez problema koristiti i *Korisnika* ili *Zaposlenika*, bez narušavanja logike sistema.

I princip: Naše klase su koncipirane tako da sadrže samo neophodne podatke, bez nepotrebnih metoda. Nismo izdvajali posebne interfejse jer su ove klase isključivo podatkovne, dok se poslovna logika i sve funkcionalnosti realizuju unutar kontrolera.

D princip: Kao što smo već spomenuli, *Korisnik* i *Zaposlenik* nasljeđuju klasu *Osoba*. Pošto je *Osoba* apstraktna i služi samo za čuvanje osnovnih podataka, vjerujemo da se u budućnosti neće mijenjati. Zbog toga izmjene u izvedenim klasama neće utjecati na nju, jer je sama klasa dovoljno stabilna i neće zahtijevati dodatne prilagodbe