

# 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:** Sistem je projektovan tako da je otvoren za proširenje, a zatvoren za izmjene. Na primjer, korištenje enumeracija kao što su `'ReportType'`, `'PaymentType'` i `'DiscountType'` omogućava lako dodavanje novih opcija bez izmjene postojeće strukture. Također, metode u klasama omogućavaju pristup i manipulaciju podacima bez direktnog mijenjanja osnovne strukture klasa.

**L princip:** U dijagramu nije eksplicitno prikazano nasljeđivanje, ali ako bi se, na primjer, iz klase `'Person'` izvodile klase kao što su `'Customer'` i `'Employee'`, one bi mogle bez problema zamijeniti osnovnu klasu `'Person'` gdje god je to potrebno. Tako bi se zadržala konzistentnost sistema i poštovao princip supstitucije.

**I princip:** Pošto su prikazane klase uglavnom podatkovne (npr. `'Product'`, `'Order'`, `'Cart'`), one nisu opterećene nepotrebnim metodama. Funkcionalnosti su raspoređene tako da svaka klasa ima samo metode koje su direktno povezane s njenim podacima, čime se izbjegava preopterećenje nepotrebnim interfejsima.

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