

Solid principi

U okviru projekta su ispoštovani sljedeći principi:

- Liskov princip zamjene (podtipovi moraju biti zamjenjivi osnovnim tipovima)

Princip ispoštovan korištenjem apstraktnih i izvedenih klasa i polimorfizmom

- Otvoreno zatvoreni princip (entiteti bi trebali biti otvoreni za nadogradnju a zatvoreni za modifikaciju)

Softver tj. kod koji razvijamo će poštovati ovaj princip tj. sve klase, funkcije i ostalo korišteno će biti otvoreni za neku (eventualnu) nadogradnju i implementaciju novih stavki u budućnosti ali trenutno postojeći tipovi neće biti podložni modifikacijama u svrhu održavanja postojeće logike rada programa.

- Princip izoliranja interfejsa (korisnici ne smiju ovisiti o metodama koje im ne koriste)

Korisnik aplikacije će imati mogućnost korištenja samo onih metoda(opcija) koje su potrebne za njegov rad(u ovisnosti od uloge korisnika, njegovih prava pristupa i sl.), dok mu ostale metode neće biti vidljive. Na taj način, promjene nad funkcijama koje korisnik neće koristiti, neće ga ni zamarati (jer nema potrebe da zna bilo šta o njima).

- Princip pojedinačne odgovornosti (klasa treba imati samo jedan razlog promjene)

Sve klase koje će se koristiti će biti odgovorne samo za implementaciju onih metoda koje njoj pripadaju. Npr. klasa kreditna kartica brine o validaciji iste, dok je klasa rezervacija (iako usko vezana za kreditnu karticu i način plaćanja) rasterećena.

- Princip inverzije ovisnosti (moduli visokog nivoa ne trebaju ovisiti od modula niskog nivoa, moduli ne trebaju ovisiti od detalja)

Klase neće imati međusobnu ovisnost već će ovisiti samo od apstraktnih klasa i/ili interfejsa.