Analiza ispunjenosti SOLID principa

**Princip S**: svaka klasa treba imati samo jednu odgovornost.

* Ovaj princip je u našem dijagramu klasa ispunjen, jer svaka klasa obavlja operacije vezane samo za nju i za njene atribute
* Kao primjer vidimo da klasa Agencija radi samo sa svojim atributima i upravlja njima, te implementira interfejs Rezervacije preko kojeg se vrši rezervacija putovanje od strane klijenata.
* Metode klase Putovanje također upravljaju i rade samo sa atributima te klase.
* Za sve ostale klase vrijedi isto, te zaključujemo da je princip S ispunjen.

**Princip O**: klasa treba biti otvorena za nadogradnje, ali zatvorena za modifikacije.

* Modifikacije bilo koje klase koja se koristi u skolu neke druge klase ne bi uzrokovala izmjene na klasi koja je koristi.
* Naše klase koriste druge klase u vidu atributa kojima je moguće jedino pristupiti i dalje pozivati metode te klase, ili ih koriste kao elemente liste, kojima je također moguće pristupiti ili brisati ih. Niti jedan slučaj od gore navedenih ne modifikuje klase koje koriti.

**Princip L**: svaka osnovna klasa treba biti zamjenjiva svim svojim podtipovima bez da to utječe na ispravnost rada programa

* Jedina apstraktna klasa jeste klasa Korisnik koja sadrži username i password korisnika. Korisnik može biti Administrator ili Klijent. U slušaju da je na primjer potrebno vidjeti sve username-e u sistemu ili ukoliko je potrebno prebrojati koliko ukupno korisnika sistem ima, tada obje podklase (Administrator i Klijent) mogu zamijeniti klasu Korisnik.
* Vidimo da je princip L također ispunjen.

**Princip I**: bolje je imati više specifičnih interfejsa, nego jedan generalizovani.

* Obzirom da imamo dva interface-a (Rezerviši i IzračunCijene) čija je uloga tačno definisana i nije moguće te interface-e razdvojiti na više specifičnih, možemo zaključiti da je princip I ispunjen

**Princip D**: sistem klasa i njegovo funkcionisanje treba ovisiti o apstrakcijama, a ne o konkretnim implementacijama

* U dijelu gdje se javlja zavisnost klase Agencija od klasa izvedenih od apstraktne klase Korisnik, veza je implementirana preko apstraktne klase (lista korisnika), te se kroz metode vrši selekcija korisnika (u zavisnoti šta metoda radi) i vraćaju se liste čiji su atributi nasljeđene klase.