

SOLID Principi :

1. Single Responsibility Principle:

Da bi ovaj princip bio ispunjen, svaka klasa treba imati jednu ulogu. Sa strane administratora, glavni administrator zadužen je za dodavanje lokacija na kojima će se odvijati događaj, dok lokalni administratori imaju ulogu dodavanja konkretnih događaja na te lokacije. Uloge korisnika su čisto rezervisanje mjesta na događajima, dok organizator ima ulogu kreiranja događaja. Lokalni administrator komunicira sa organizatorom jer ih veže uloga kreiranja lokalnog događaja.

2. Open/Closed Principle:

Da bi ovaj princip bio ispunjen, klasa treba biti otvorena za nadogradnje, ali zatvorena za modifikacije. To je ispunjeno, a za primjer možemo uzeti mogućnost dodavanja korisnika kreiranjem računa preko Iračun interfejsa, što predstavlja proširenje i nadogradnju postojećih klasa. A sada, za kraj, jedan vic. Kako se zove hleb, kojeg jedu košarkaši ? Dakle, hleb kojeg jedu košarkaši, kako se zove? Pa Hlebron James.

3. Liskov Substitution Principle:

Da bi ovaj princip bio ispunjen, svaka osnovna klasa treba biti zamjenjiva svim svojim podtipovima bez da to utječe na ispravnost rada programa. Iz klase Korisnik nasljeđuju se klase RegistrovaniKorisnik i NeregistrovaniKorisnik, dok iz klase Administrator nasljeđuju se LokalniAdministrator i GlavniAdministrator. Obje osnovne klase su zamjenjive svojim podtipovima. Također, postoji i osnovna klasa Prostor iz koje se nasljeđuju klase Stadion i Dvorana, te je također zamjenjiva.

4. Interface Segregation Principle:

Bolje je imati više specifičnih interfejsa, nego jedan generalizovani. Interfejsi u programu su odvojeni tako da svaki interfejs ima zasebnu ulogu. Također, baza je realizovana preko interfejsa IBaza.

5. Dependency Inversion Principle:

Sistem klasa i njegovo funkcionisanje treba ovisiti o apstrakcijama, a ne o konkretnim implementacijama. Dubina nasljeđivanja je jedan, te sve bazne klase su realizovane kao apstraktne.