

SOLID PRINCIPI

Single Responsibility Principle

- Ovaj princip je zadovoljen jer svaka klasa ima samo jednu odgovornost tj. obavlja jednu ulogu u sistemu. Klase se većinom sastoje od gettera i settera i jedne dodatne metode što je dokaz da obavljaju samo jednu ulogu. Izuzetak je nešto složenija klasa Sistem koja upravlja cijelim sistemom, ali i ona ispunjava ovaj princip jer je jedina odgovornost da upravlja sistemom.

Open/Close Principle

- Entiteti softvera (klase, moduli, funkcije) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije. Ovaj princip je ispoštovan jer sve klase koje za attribute imaju druge klase, ne vrše modifikaciju tih klasa nego ih koriste kao attribute u nekoj listi.

Liskov Substitution Principle

- Ovaj princip kaže da podtipovi moraju biti zamjenjivi osnovnim tipovima. U našem sistemu nemamo nasljeđivanja, ali i pored toga uloga administratora, upravnika parkinga i običnog korisnika je zamjenjiva klasom korisnik te je ovaj princip ispunjen.

Interface Segregation Principle

- Interfejs bi trebao obavljati samo jednu vrstu akcije. S obzirom da mi nemamo interfejsa ovaj princip je zadovoljen.

Dependency Inversion Principle

- Prilikom nasljeđivanja treba razmotriti slučaj da je osnovna klasa apstraktna. Apstraktne klase i interfejsi mnogo se manje mijenjaju nego njihove konkretne izvedenice. Stoga je bolje ovisiti o apstrakcijama nego o stvarnim klasama. S obzirom da nemamo nasljeđivanja u našem sistemu i ovaj princip je zadovoljen.