## SOLID PRINCIPLES

## Single-Responsibility Principle

Dijagram klasa organizovan je na prinipu da je svakoj klasi dodjeljena tačno jedna uloga. Npr. klasa Korisnik čuva podatke isključivo vezane za korisnika sistema; klasa „Administrator“ sadrži sve podatke neophodne za specifikaciju jednog administratora, dok je sami proces brisanja korisnika dodjeljen klasi „BrisanjeNaloga“ koja je u direktnoj vezi sa klasom „Administrator“. Na taj način, svakoj od klasa je dodjeljena tačno jedna odgovornost. Stoga, princip vrijedi za cijeli dijagram.

## Open-Closed Principle

Ovaj princip je u našem sistemu ispunjen za dvije klase, a to su „Plaćanje“ i „Korisnik“. Dodavanjem još jedne alternativne forme plaćanja, kao i dodavanje nove vrste korisnika ne zahtjeva ni promjene ni brisanje kako bazne klase, tako ni drugih klasa.

**Liskov Substitution Principle**

U slučaju obje bazne klase u našem sistemu („Korisnik“ i „Plaćanje“) ne postoji potreba da naslijeđenu klasu šaljemo kao parametar umjesto bazne klase. Stoga je ispoštovan LISKOV princip.

## Interface Segregation Principle

U našem sistemu postoji interfejs „IPotvrdaUplate“. S obzirom na to da se interfejs i njegove metode isključivo odnose na potvrđivanje izvršene gotovinske uplate između korisnika sistema, princip izoliranja interfejsa je ispunjen. Drugi interfejs koji koristimo je „IMedijator“, koji se kao i prvi interfejs odnosi isključivo na kontrolu sadržaja poruka i sve klase ga u potpunosti realizuju.

## Dependency Inversion Principle

Pojednostavljenje definicije princip naglašava da klasa ne smije ovisiti od konkrentih klasa, već od apstrakcija, kao i da prilikom nasljeđivanja modulu višeg nivoa ne smiju ovisiti od modula nižeg nivoa. U okviru našeg sistema, primjer klase za koju važi dati princip je klasa „Klijent“ , ovisna o klasi „Plaćanje“, kao i klasa „Administrator“ koja je ovisna o klasi „Korisnik“.