

SOLID PRINCIPI

1. S PRINCIP – SINGLE RESPONSIBILITY PRINCIPLE

Nijedna od klasa u našem sistemu ne obavlja više od jednog zadatka. Većina klasa imaju samo konstruktor, gettere i settere. Ovaj princip je zadovoljen.

2. O-OPEN-CLOSE PRINCIPLE

Dodavanje novih metoda neće zahtijevati uređivanje već postojeće klase i njenih atributa. Na primjer, možemo promijeniti implementaciju klase Adresa bez da brinemo o klasama koje je koriste, a to su Bankomat, Filijala i Klijent. To vrijedi za sve klase u našem sistemu, pa je i ovaj uslov zadovoljen.

3. L-LISKOV SUBSTITUTION PRINCIPLE

Na svim mjestima gdje se koristi osnovni objekat se može koristiti i izvedeni, i pritom će to imati smisla. Naše dvije klase iz kojih se nasljeđuje su Korisnik i KreditBaza, i one se kao takve trenutno ne koriste nigdje, i samim tim nismo prekršili L princip.

4. I-INTERFACE AGREGATE PRINCIPLE

Svaki interfejs obavlja tačno jednu vrstu akcija, i zadovoljava S princip. Na primjer IAdministratorUpravljačNovostima radi samo sa novostima, dodaje ih, uređuje i briše. I princip sistema je zadovoljen.

5. D-DEPENDENCY INVERSION PRINCIPLE

Sve klase iz kojih se nasljeđuje su apstraktne. To su klase Korisnik i KreditBaza. Uslov je zadovoljen.