SOLID PRINCIPI

1. S PRINCIP – SINGLE RESPONSIBILITY PRINCIPLE

S- princip zatijeva da svaka klasa ima samo jednu odgovornost. Naše osnovne klase imaju većinom samo konstruktor, gettere i settere, a naše repositroy klase obavljaju samo jedan tip akcija, većinom samo nad jednom osnovnom klasom. Ovaj princip je, stoga, zadovoljen.

2. O-OPEN-CLOSE PRINCIPLE

O – princip zahtijeva da klasa koja koristi neku drugu klasu ne treba biti modificirana pri uvođenju novi funkcionalnosti, ili pri potrebi za mijenjenjem druge klase. Trudili smo se da za one klase u kojima bi moglo doći do izmjena u budućnosti zadovoljimo ovaj princip. Na primjer, možemo promijeniti implementaciju klase Adresa bez da brinemo o klasama koje je koriste, a to su Bankomat, Filijala i Klijent.

3. L-LISKOV SUBSTITUTION PRINCIPLE

Ovaj princip zahtijeva da je na svim mjestima na kojima se koristi osnovni objekat moguće iskoristiti i izvedeni objekat. 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

I - princip zahtijeva da svi interfejsi zadovoljavaju princip S. Svaki naš interfejs obavlja tačno jednu vrstu akcija, i zadovoljava S princip. Na primjer INovosti radi samo sa novostima, dodaje ih, uređuje i briše. I princip sistema je zadovoljen.

5. D-DEPENDENCY INVERSION PRINCIPLE

D - princip zahtijeva da pri nasljeđivanju od strane više klasa bazna klasa uvijek bude apstraktna. Sve klase u našem sistemu iz kojih se nasljeđuje su apstraktne. To su klase Korisnik i KreditBaza. Uslov je zadovoljen.