

SOLID PRINCIPI

1. Single Responsibility Principle - Princip pojedinačne odgovornosti

„KLASA BI TREBALA IMATI SAMO JEDAN RAZLOG ZA PROMJENU.“

U našem klasnom dijagramu, ovaj princip je već u startu ispoštovan, zato što svaka klasa sadrži samo 'svoje' attribute i brine se o jednoj stvari.

2. Open Closed Principle - Otvoreno zatvoren princip

„ENTITETI SOFTVERA (KLASE, MODULI, FUNKCIJE) TREBALI BI BITI OTVORENI ZA NADOGRADNJU, ALI ZATVORENI ZA MODIFIKACIJE.“

I ovaj princip je ispoštovan zato što ne dolazi do modifikacije klase već samo okruženja, tj. atributa, čak i ako klasa sadrži instance neke druge klase. U slučaju da se dodaju neke nove metode, ni tada neće doći do izmjene čitave klase.

3. Liskov Substitution Principle - Liskov princip zamjene

„PODTIPOVI MORAJU BITI ZAMJENJIVI NJIHOVIM OSNOVNIM TIPOVIMA.“

Mišljenja smo da je u našem dijagramu ispoštovan Liskov princip zamjene jer je moguće bez problema koristiti obje klase („Korisnik“ i „Administrator“) izvedene iz apstraktne klase („User“).

4. Interface Segregation Principle -Princip izoliranja interfejsa

*KLIJENTI NE TREBA DA OVISE O METODAMA KOJE NEĆE
UPOTREBLJAVATI*

Ovaj princip je ispoštovan jer klase sadrže samo osnovne metode koje će se vrlo vjerovatno sve koristiti.

5. Dependency Inversion Principle – Princip inverzije ovisnosti

***A. MODULI VISOKOG NIVOA NE BI TREBALI OVISITI OD MODULA
NISKOG NIVOA. OBA BI TREBALO DA OVISE OD APSTRAKCIJA.***

***B. MODULI NE BI TREBALI OVISITI OD DETALJA. DETALJI BI TREBALI
BITI OVISNI OD APSTRAKCIJA.***

Ovaj princip je ispoštovan jer je bazna klasa User, koju ne možemo instancirati, ujedno i apstraktna klasa iz koje su naslijeđene klase Korisnik i Administrator, koje možemo instancirati.