

SOLID PRINCIPI

Single Responsibility Principle – Princip pojedinačne odgovornosti

Klase trebaju imati pojedinačnu odgovornost. Treba postojati samo jedan razlog za promjenu klase.

Ovaj princip je ispunjen, sve klase imaju pojedinačnu odgovornost. Razmotrimo klasu Korisnik. Ova klasa sadrži lične podatke o korisnicima. Također, možemo vidjeti da je ovaj princip ispunjen i na osnovu klase Recept koja sadrži osnovne informacije potrebne za objavu recepta .

Open Closed Principle – Otvoreno zatvoren princip

Svaka klasa treba biti otvorena za nadogradnju ali ne i za modifikacije.

Na dijagramu klasa vidimo da je ovaj princip zadovoljen. Ukoliko se odlučimo za promjenu jedne klase to neće uticati na druge klase.

Liskov Substitution Principle – Liskov princip zamjene

Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.

Dijagram klasa nema apstraktnih klasa.

Interface Segregation Principle – Princip izoliranja interfejsa

Klijenti ne treba da ovise o metodama koje neće upotrebljavati.

Ovaj princip je također zadovoljen jer dijagram klasa ne sadrži niti jednu klasu koja se može nazvati „debela“ klasa (klasa koja ima veliki broj metoda) i upotrebljavaju se sve metode klase.

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 možemo i jednostavnije interpretirati:

Ne treba ovisiti od konkretnih klasa. Prilikom nasljeđivanja treba razmotriti slučaj da je osnovna klasa apstraktna.

Dijagram klasa nema nasljeđivanja.