

SOLID PRINCIPI -AMA Cosmetics-

1. Single Responsibility Principle - Princip pojedinačne odgovornosti :

Potrebno je da svaka klasa ima tačno jednu odgovornost koja je vezana samo za nju. U našem dijagramu je to ispunjeno. Ako uzmemo za primjer klasu *Proizvod* možemo primjetiti da sadrži atribute i metode koje su vezane samo za nju kao što su *ID*, *naziv*, *opis*, *cijena*, *količina*, *kategorija* i sl. Također, klasa *Korisnik* sadrži listu narudžbi ali ne vrši njenu promjenu, već se sve promjene vrše putem klase *Narudžba*.

2. Open Closed Principle - Otvoreno zatvoren princip:

Ovaj princip zahtjeva da ovaj sistem bude otvoren za proširenje/nadogradnju, a zatvoren za modifikacije. To znači da ako izvršimo promjenu u jednoj klasi to neće zahtijevati promjenu drugih klasa, već ćemo samo postojeći dijagram dodatno proširiti. Na primjer, ukoliko izvršimo izmjenu klase *Proizvod*, to neće izmijeniti klasu *Korisnik*.

3. Liskov Substitution Principle - Liskov princip zamjene :

Ovaj princip se odnosi na to da podtipovi moraju bit zamjenjivi njihovim osnovnim tipovima bez da izazivaju probleme u sistemu. Na našem dijagramu možemo vidjeti da su na primjer podtipovi *NeregistrovaniKorisnik*, *RegistrovaniKorisnik* i *PremiumKorisnik* izvedeni iz apstraktne klase Korisnik odnosno da su oni zamjenjivi njihovim osnovnim tipom *Korisnik*.

4. Interface Segregation Principle - Princip izoliranja interfejsa:

Ovaj princip zahtjeva da klijenti ne treba da ovise o metodama koje neće upotrebljavati.

Možemo primijetiti da na našem dijagramu klasa niti jedna klasa ne posjeduje metodu koja je se ne tiče.

5. Dependency Inversion Principle - Princip inverzije ovisnosti :

Ovaj princip zahtjeva da sistem treba ovisiti od konkretnih klasa, te prilikom nasljeđivanja treba razmotriti slučaj da je bazna klasa apstraktna. Na dijagramu je jasno da je bazna klasa *Korisnik* apstraktna i da se iz nje izvode klase *NeregistrovaniKorisnik*, *RegistrovaniKorisnik* i *PremiumKorisnik*.