

SOLID principi

Tim 19 – VetNet System

1. Single responsibility principle – princip pojedinačne odgovornosti

Ovaj princip nam govori da bi svaka klasa trebala imati samo jedan razlog za promjenu. Odnosno, svaka klasa treba da zna samo o jednoj stvari i da ima samo jednu odgovornost. U našem sistemu svaka klasa ima svoju odgovornost čime je ovaj zahtjev ispunjen.

2. Open closed principle – otvoreno zatvoren princip

U sistemu, klase su povezane na način da jedna klasa kao atribut sadrži objekat tipa druge klase. Samim tim, mijenjanje jedne klase neće uzrokovati mijenjanje druge klase. Npr. klasa *Pregled* kao svoje atribute sadrži objekte tipa *Ljubimac* i *Veterinar*. Mijenjanje ovih klasa neće narušiti funkcionalnost klase *Pregled*.

3. Liskov substitution principle – Liskov princip zamjene

Dijagram klasa sadrži apstraktnu klasu *Korisnik*. Iz nje su izvedene klase *Vlasnik*, *Apotekar* i *Veterinar*. Prilikom dodavanja novog korisnika u sistem, umjesto objekta tipa klase *Korisnik* mi dodajemo objekat tipa *Vlasnik*, *Apotekar* ili *Veterinar*. Kako svaki od ovih objekata smisleno zadržava sve atribute klase *Korisnik*, tako je ispunjen i ovaj princip.

4. Interface segregation principle – princip izoliranja interfejsa

Klijenti ne treba da ovise o metodama koje neće upotrebljavati. Ovaj zahtjev je ispunjen činjenicom da sve klase u sistemu obavljaju samo one aktivnosti koje su klijentu potrebne.

5. Dependency inversion principle – princip inverzije ovisnosti

Ovaj princip nam govori da ne treba ovisiti od konkretnih klasa, odnosno da prilikom nasljeđivanja bazna klasa bude apstraktna. U našem sistemu imamo jednu baznu klasu *Korisnik*, iz koje su izvedene klase *Apotekar*, *Veterinar* i *Vlasnik*. Kako je ova bazna klasa apstraktna tako je ispunjen i posljednji princip.