

# SOLID PRINCIPI

## 1. Single Responsibility Principle - Princip pojedinačne odgovornosti

Ovaj zahtjev nam govori da bi klase trebale imati samo pojedinačne odgovornosti odnosno da treba da postoji samo jedan razlog za promjenu klase. U našem sistemu svaka klasa ima jednu odgovornost čime smo ispoštovali ovaj princip.

## 2. Open Closed Principle - Otvoreno zatvoren princip

U dijagramu klasa, klase smo povezivali agregacijom i kompozicijom tako da u klasama kao atribut često imamo objekte nekih drugih klasa, pa samim tim promjena u jednoj klasi neće predstavljati promjenu u drugoj klasi. To se također može vidjeti i na primjeru da ukoliko bismo u klasi Uposlenik dodali neki novi atribut, ne bismo morali mijenjati klasu Osoba ili Pacijent.

## 3. Liskov Substitution Principle - Liskov princip zamjene

Dijagram klasa koji smo napravili sadrži apstraktnu klasu osoba. Iz nje su izvedene klase uposlenik i pacijent. Kada npr. dodajemo neko novo lice u sistem umjesto da dodajemo osobu dodajemo uposlenika ili pacijenta sa svim osobinama klase osoba. Kada se sjetimo kako glasi princip ("podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima") dolazimo do zaključka da je princip ispunjen.

## 4. Interface Segregation Principle - Princip izoliranja interfejsa

Ovaj zahtjev nam govori da klijenti ne treba da ovise o metodama koje neće upotrebljavati. Ovo je ispoštovano na način da su sve klase mogu obavljati samo one aktivnosti koje su potrebne.

## 5. Dependency Inversion Principle - Princip inverzije ovisnosti

Da bi ovaj zahtjev bio ispunjen ne treba ovisiti od konkretnih klasa i prilikom nasljeđivanja treba razmotriti slučaj da je osnovna klasa apstraktna.

Imamo da ovaj princip zahtijeva da pri nasljeđivanju od strane više klasa bazna klasa bude uvijek apstraktna, što je upravo kod nas i slučaj, klasa Osoba je apstraktna klasa.