

Pet osnovnih **SOLID** principa:

### 1. Single Responsibility Principle

Ovaj princip je zadovoljen u našem modelu jer svaka klasa ima tačno jednu ulogu. Sve klase imaju samo settere i gettere, eventualno implementiraju interfejs koji obavlja traženu funkcionalnost.

### 2. Open-closed principle

Ovaj princip je zadovoljen u našem modelu jer bilo koja promjena neke od klasa ne uzrokuje promjenu neke druge klase. Klasa *Zahtjev* koristi attribute koji su objekti klasa *Klijent* i *Strucnjak*, međutim niti jedna metoda ne ovisi o konkretnoj implementaciji ovih klasa.

### 3. Liskov substitution principle

Ovaj princip je zadovoljen u našem modelu jer svaka osnovna klasa može biti zamjenjiva svim svojim podtipovima, a da to ne utiče na ispravnost rada programa. Ovo je ispoštovano jer klase *Klijent* i *Strucnjak* naslijeđuju sve attribute i metode bazne klase, a ovo je ujedno i jedino nasljeđivanje u modelu.

### 4. Interface segregation principle

Ovaj princip je zadovoljen jer ne postoji jedan generalni interfejs u modelu, nego više specifičnih. Svaki od njih obavlja samo jednu funkcionalnost i podržava samo osnovne potrebne operacije.

### 5. Dependency inversion principle

Ovaj princip je zadovoljen jer govori da sistem klasa i njegovo funkcionisanje treba ovisiti o apstrakcijama, a ne o konkretnim implementacijama, što je i urađeno. Na vrhu hijerarhije svih klasa nalazi se ili apstraktna klasa, ili interfejs što je pokazatelj da je ovaj princip najvjerojatnije ispoštovan.