

## ***SOLID PRINCIPI***

### 1. Princip pojedinačne odgovornosti

Ovaj princip je zadovoljen zbog činjenice da svaka klasa ima samo jednu odgovornost jer sadrži samo getere i setere i eventualno još jednu metodu dok su sve sporedne odgovornosti izdvojene u interefejse

### 2. Otvoreno-zatvoreni princip

Ovaj princip kaže da svaka klasa mora biti otvorena za nadogradnje i zatvorena za modifikacije. Većina klasa nema za atribute druge klase osim kontejnerske klase Sistem koja sadrži liste svih drugih klasa. Što se tiče nadogradnje, moguće je nadograditi klase dodatnim nasljeđivanjima i slično.

### 3. Liskov princip zamjene

Podklase bi trebale biti zamjenjive baznoj klasi. Ovaj princip kaže da bilo koja upotreba bazne klase omogućava upotrebu i izvedenih klasa, sa istim rezultatom. Klasu Korisnik nasljeđuje klasa RegistrovaniKorisnik i zamjenjiva je njome, također klasu RegistrovaniKorisnik nasljeđuje klasa VipKorisnik i zamjenjiva je njome tako da je ovaj princip ispunjen.

### 4. Princip inverzije ovisnosti

Moduli visokog nivoa ne bi trebali da zavise od modula niskog nivoa, oba bi trebalo da zavise od apstrakcija. VipKorisnik nasljeđuje klasu RegistrovaniKorisnik koja nije apstraktna ali ta klasa nasljeđuje klasu Korisnik koja jeste tako da je ispunjen i ovaj princip.

### 5. Princip izoliranja interfejsa

Svi interfejsi su minimalistički i podržavaju samo potrebne operacije odnosno klase ne implementiraju metode koje im nisu potrebne. I svaki od njih ima zasebnu ulogu tako da je ispunjen i ovaj princip.