

## **SOLID PRINCIPI:**

### **Single Responsibility Princip:**

Po ovom principu svaka klasa treba imati samo jednu ulogu.

Većina naših klasa je veoma jednostavna jer imaju samo nekoliko metoda tako da nije narušen ovaj princip.

### **Open/Closed Princip:**

Po ovom principu svaka klasa mora biti otvorena za nadogradnju, a zatvorena za modifikaciju.

Pri uvođenju novih funkcionalnosti nema potrebe za modifikacijom klasa.

### **Liskov Substitution Princip:**

Po ovom principu svaka osnovna klasa treba biti zamjenjiva svim svojim podtipovima bez da to utječe na ispravnost rada programa.

Iz klase Osoba se nasljeđuju klase: Administrator i Korisnik. Klasa Osoba je zamjenjiva njima pa je prema tome ovaj princip zadovoljen. Isto vrijedi i za klase Korisnik i SuperKorisnik.

### **Interface Segregation Princip:**

Po ovom principu interfejs treba obavljati jednu vrstu akcije.

Kako u našem sistemu nema nijedan interfejs samim tim ovaj princip je ispunjen.

### **Dependancy Inversion Princip:**

Po ovom principu pri nasljeđivanju od strane više klasa bazna klasa uvijek treba biti apstraktna.

U našem projektu klasa Osoba je apstraktna. Zbog toga smo sve zajedničke atribute i metode izdvojili u ovu baznu klasu.

Iako klasa SuperKorisnik nasljeđuje klasu Korisnik, koja nije apstraktna, ovaj princip je zadovoljen zbog apstraktne klase Osoba.