

# 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 koju metodu dok su sve sporedne odgovornosti izdvojene u interefejs.

## 2. Otvoreno-zatvoreni princip

Ovaj princip kaže da svaka klasa mora biti otvorena za nadogradnje i zatvorena za modifikacije. Ukoliko promijenimo jednu klasu neće doći do promjene druge klase. Ni u jednoj klasi nemamo omogućeno mijenjanje druge klase vec isključivo samo getere i setere uz pokoju metodu.

Metode u svim klasama mijenjaju attribute samo tih klasa.

## 3. Liskov princip zamjene

Ovaj princip kaže da bilo koja upotreba bazne klase omogućava upotrebu i izvedenih klasa, sa istim rezultatom. Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.

Klasu Macka nasljeđuje klasa Premium i klasa Premium je zamjenjiva klasom Macka sto se slaze sa Liskovim principom zamjene.

Nad instancama klase Premium možemo koristiti sve metode koje mozemo koristiti nad instancama klase Macka.

## 4. Princip izoliranja interfejsa

Imamo samo jedan interfejs IProfil i on ograničava klasu da podržava samo potrebne metode.

Nema potrebe za grananjem istog jer sve klase koje njega implementiraju koriste sve njegove metode.

## 5. Princip inverzije ovisnosti

Moduli visokog nivoa ne bi trebali da zavise od modula niskog nivoa, oba bi trebalo da zavise od apstrakcija.

Premium nasljeđuje klasu Macka koja nije apstraktna ali ta klasa implemetira interfejs IProfil tako da je ispunjen i ovaj princip.