**Princip pojedinačne odgovornosti:** „Klasa bi trebala imati samo jedan razlog za promjenu”.

Sve klase vode brigu samo o stvarima koje su za njih vezane. Ukoliko trebamo nesto promijeniti, mijenja se samo jedna klasa, i to ne utice na druge klase.

**Otvoreno zatvoren princip :**  “Entiteti softvera (klase, moduli, funkcije) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije.”

Klase su ovisne, ali do te mjere da, ukoliko dodje do potrebe za modifikacijom, nadogradnja jedne klase nece uzrokovati promjenu u ostalim klasama. Klase se mogu nadograđivati ali ne i mijenjati.

**Liskov princip zamjene** : „Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima“

Svaki od podtipova klase Klijent se može zamjeniti sa tom klasom. Klase Invalid, Ucenik, Student, Penzioner su sve zamjenjive klasom Klijent.

**Princip inverzije ovisnosti** : „Moduli visokog nivoa ne bi trebali ovisiti od modula niskog nivoa. Oba bi trebalo da ovise od apstrakcija.”

Moduli visokog nivoa nisu zavisni od onih nižeg, dok su sve niži moduli više zavisi od detalja. Klasa Klijent je jednostavna i nema razloga za njenu promjenu.

**Princip izoliranja interfejsa:** Koristi se malo metoda, pa klijent i ne mora brinuti o metodama koje nece koristiti. Kompleksne klase ne ovise o jednostavnijih. Tipovi ovise o apstrakcijama u vidu klasa koje ne pretstavljaju konkretne entitete (npr. klasa Osoba).