

# SOLID principi

**S** – *single responsibility* – ovaj princip je ispoštovan pošto svaka klasa pokriva specifičnu grupu funkcionalnosti i bit će zadužena za taj dio biznis logike. Klase su dizajnirane upravo s ovim ciljem, tako da npr. svaka modifikacija funkcionalnosti vezane za ispite će utjecati samo na klasu *Ispit*. Analogno i za ostale klase i funkcionalnosti koje pokrivaju.

**O** – *open closed* – Sadašnji dizajn klasa je zatvoren pošto ove klase mogu da iznesu planirane funkcionalnosti. Moguća proširenja neće zahtijevati modifikaciju čitavog sistema, npr. apstraktna klasa *Osoba* je spremna za slučaj dodavanja nekog novog tipa korisnika u sistem. Potencijalna proširenja funkcionalnosti bi značila dodatke u smislu novih klasa, ali ne bi trebalo mijenjati postojeće. Npr. ako želimo uvesti u igru projekt i/ili seminarski rad kao oblik studentske aktivnosti, to se može uraditi nasljeđivanjem ovih novih klasa iz klase *Zadaca*, i tada možemo studentske aktivnosti podijeliti na ispite i zadatke, pri čemu će ovo biti funkcionalno bez obzira što *Zadaca* nije apstraktna klasa.

**L** – *Liskov substitution* – Klase *Student*, *Nastavnik* i *StudentskaSluzba* su naslijeđene iz apstraktne klase *Osoba*, i ovo je urađeno ispravno pošto su i studenti i nastavnici i zaposlenici studentske službe zaista i osobe.

**I** – *interface segregation* - u inicijalnom dizajnu još uvijek nemamo interfejsa, što znači da sigurno nije prekršen ovaj princip. Ipak, očigledno će se javiti potreba za interfejsima u nastavku dizajniranja, ali će svi biti implementirani prateći design patterne, koji dokazano znače dobar dizajn i ispunjenje SOLID principa.

Primjer: Klase *Ispit* i *Zadaca* imaju očigledne sličnosti pošto su oboje aktivnosti koje se boduju i eventualno ulaze u zaključnu ocjenu, tako da će biti moguće izdvojiti neke njihove zajedničke metode u zaseban interfejs.

**D** – *dependency inversion* - uzeti za primjer klasu *Osoba* i njenu djecu. Očigledna je neminovnost svih ovih klasa i razlike među njima, pošto svaka predstavlja jednog od aktera Sistema, te su razlike među njima ogromne. Zato, bilo kakve akcije nad jednom od njih neće utjecati na klasu *Osoba* koja je zadužena za funkcionalnost notifikacije, a za sve ostale detalje potrebno je raditi sa jednim njenim djetetom.