

Manifestovanje SOLID principa u projektu FitnessFusion

U ovom dokumentu se vrši kratak pregled i analiza manifestovanja SOLID principa na modele koji su prikazani na dijagramu klase. Analiza se vrši pojedinačno za svaku klasu, gdje se u kratkim crtama navodi na koji način je zadovoljen svaki od principa.

Klasa Osoba:

Klasa *Osoba* je apstraktna i ona predstavlja istoimeni generalizirani model koji je zajednički za sve korisnike sistema. Način na koji ona zadovoljava SOLID principe su sljedeći:

- **SRP** je zadovoljen jer klasa može imati samo jedan razlog za izmjenu, tj. ona ne posjeduje metode koje obavljaju više neovisnih funkcionalnosti, a za koje postoji mogućnost modificiranja. Metode koje ona posjeduje su takozvani setteri i getteri, te su to metode koje koegzistiraju sa odgovarajućim atributima. Dakle, potencijalni razlog za izmjenu ove klase je da se doda ili izbaci atribut, što podrazumjeva prateće set i get metode. Također, ova klasa posjeduje metodu za prijavu u sistem koja je zajednička za sve aktere.
- **OCP** je također zadovoljen jer se klasa može dodatno nadograditi. Moguće je vršiti nasljeđivanje ove klase kao i implementiranje interface-a što neće uticati na samu klasu tj. neće biti potrebno vršiti modifikaciju nad njom. Jedini oblik modifikacije koji je moguć, ali ne narušava ovaj princip, jer je u skladu sa SRP principom, je dodavanje ili uklanjanje atributa.
- **LSP** se manifestira u nasljeđivanju i ova klasa potpuno zadovoljava ovaj princip jer ne posjeduje metode za koje se može desiti nelegalan poziv nakon nasljeđivanja. U slučaju nadograđivanja (implementiranja interface-a) potrebno je ponovo razmotriti ovaj princip i pravilno izvršiti nadogradnju.
- **ISP** je također zadovoljen jer ne narušava SRP princip. Sve metode koje ova klasa treba da koristi, a koje nisu navedene u njoj se mogu implementirati pomoću interfacea.
- **DIP** se zadovoljava povećanom ovisnosti o interface-ima i apstraktnim klasama, što ova klasa omogućava obzirom da je ona apstraktna što je primjer poželjne ovisnosti u sistemu. Uzevši u obzir i metriku NOC (eng. *Number Of Children*) koja je u ovom slučaju jednaka 2, može se zaključiti da je ovaj princip ispoštovan.

Klasa Trener:

Ova klasa se nasljeđuje iz klase *Osoba* i ona zadovoljava SOLID principe na sljedeći način:

- **SRP** je zadovoljen jer klasa ima samo jednu odgovornost, a to je da unosi rezultate korisnika. Sve potencijalne izmjene se odnose na tu metodu. Metode koje se odnose na pregled i izmjenu korisnikovog rasporeda su involvirane u sistem na taj način da su to dodatne odgovornosti ove klase.
- **OCP** je ispoštovan jer sve potencijalne promjene kao što su dodavanje mogućnosti uvida u rezultate ili sličnih funkcionalnosti se može postići implementiranjem interface-a pri čemu neće biti potreba za modificiranjem ove klase jer je nadogradnja neovisna.
- **LSP** je ostvaren nasljeđivanjem i ova klasa može biti zamjenjena svojom baznom klasom bez pojave neregularnog nasljeđivanja. Također, kako je zadovoljen ovaj princip za baznu klasu, tako je on iz istih razloga zadovoljen i za ovu.
- **ISP** je u indirektnoj vezi sa SRP i on je zadovoljen jer ova klasa ne ovisi o onim funkcionalnostima koje neće koristiti. Tako je zadovoljen i OCP jer svakom nadogradnjom novih metoda se ne narušavaju ovi principi.
- **DIP** se manifestira sa brojem asocijacija koje su uspostavljene sa ovom klasom. Prema metriki CF (eng. *Coupling factor*) se može zaključiti da je ovisnost od ove klase relativno mala. Uzevši u obzir i DIT (eng. *Depth of Inheritance Tree*) metriku koja je za ovu klasu jednaka vrijednosti 1, može se zaključiti da je ovaj princip zadovoljen.

Klasa Korisnik:

Klasa Korisnik je drugo dijete klase *Osoba* i ona zadovoljava principe na sljedeći način:

- **SRP** se manifestira tako što ova klasa ima samo jednu odgovornost, a to je upravljanje podacima koji se odnose na korisnika u funkcionalnostima sistema kao i pokretanje ostalih funkcionalnosti.
- **OCP** se ostvaruje time što je klasa *Korisnik* otvorena za nadogradnju novim metodama (putem nasljeđivanja ili interface-a), a zatvorena za modifikaciju.
- **LSP** je zadovoljen jer ova klasa se može zamijeniti sa osnovnim tipom, tj. ovim nasljeđivanjem nije postignuto neregularno nasljeđivanje.
- **ISP** je ostvaren pomoću metoda koje su relevantne za ovu klasu.

Objektno Orijentisana Analiza i Dizajn

- **DIP** je ispoštovan jer ova klasa zavisi od apstraktne klase *Osoba* što je u skladu sa principima dobrog dizajna.

Klasa Kartica:

Klasa *Kartica* poštuje ove principe kao što je to prikazano u nastavku:

- **SRP** je ispoštovan jer ova klasa ima zadatak samo da čuva podatke koji se odnose na karticu sa kojom se vrši plaćanje. Metode sa kojima ona raspolaže su setteri i getteri, te metode za upravljanje radom sa karticom i kao takva ima odgovornost samo za dobavljanje i obradu njenih informacija.
- **OCP** je zadovoljen. Mogućnosti sa nadogradnjom novih metoda su moguće bez modifikacije. To bi potencijalno bile neke specijalizirane metode koje bi vršile posebnu obradu nad karticom, što je moguće postići sa interface-om ili nasljeđivanjem neke klase. U slučaj da se javi potreba za dodavanjem dodatnih atributa koji se odnose na karticu, tada je potrebno izvršiti dodatne modifikacije nad atributima, što ne predstavlja veliku prijetnju da se naruši ovaj princip.
- **LSP** se ne manifestuje za ovu klasu obzirom da se ona ne nasljeđuje, niti ona nasljeđuje druge klase. Mogućnost koja može da se iskoristi, jeste da se kreira apstraktna klasa *kartica* koja bi se naslijedila za različite vrste *kartica*. Time bi se dodatno ispoštovao OCP princip.
- **ISP** je zadovoljen jer ova klasa posjeduje samo one metode koje su joj neophodne.
- **DIP** je zadovoljen jer ova klasa se ne nasljeđuje tako da zavisnosti od nje nisu prisutne.

Klasa Aktivnost:

Ova klasa zadovoljava principe na sljedeći način:

- **SRP** je zadovoljen jer klasa ima jedino zadatak da čuva osnovne informacije za aktivnosti te sve eventualne izmjene se odnose na dodavanje ili brisanje nekih od atributa sa pratećim set i get metodama.
- **OCP** ova klasa zadovoljava jer ona može da naslijedi druge klase i da implementira interface bez da se vrše modifikacije nad njom.

Objektno Orijentisana Analiza i Dizajn

- **LSP** se ne manifestira za ovu klasu jer ona ne vrši nasljeđivanje, ali moguće je izvršiti nadogradnju sistema kojom bi se ovaj princip manifestovao i ispoštovao.
- **ISP** je ispoštovan jer ova metoda ne koristi metode koje nisu relevantne za nju.
- **DIP** je zadovoljen jer ova klasa predstavlja konkretnu klasu od koje ne ovisi niti jedna druga klasa jer ona nije iznasljeđena.

Klasa Termin:

Način na koji ova klasa zadovoljava SOLID principe su:

- **SRP** je ispoštovan jer je ova klasa apstraktna. Samim time postoji i samo jedan razlog da se ona modificira. Slično kao klasa Osoba.
- **OCP** se manifestuje na način da je klasa otvorena za nadogradnju metoda ili nasljeđivanje, ali ne i za modifikacije.
- **LSP** je postignut time što je ova klasa apstraktna i predstavlja odgovarajuću baznu klasu iz koje se regularno nasljeđuju izvedene klase.
- **ISP** je ispoštovan jer klasa ne koristi metode koje nisu relevantne za nju tj. nije *fat tip* klase.
- **DIP** je ispunjen jer je ispunjen i SRP te je ovom klasom postignuta ovisnost koja je primjer dobrog dizajna.

Klasa Trening:

SOLID principi su ispoštovani na sljedeći način:

- **SRP** je zadovoljen jer ova klasa ima samo jednu odgovornost, tj. razlog da se izmjeni, a to je dodavanja novih atributa sa pratećim set i get metodama.
- **OCP** je ispunjen time što se na ovu klasu mogu nadograđivati nove funkcionalnosti bez da se ona modificira.
- **LSP** je zadovoljen jer se za ovu klasu može izvršiti suspdstitucija sa njenim baznim tipom pri čemu se zadržava svojstva regularnog nasljeđivanja.
- **ISP** je ispoštovan jer klasa ne koristi metode koje nisu relevantne za nju tj. nije *fat tip* klase.

Objektno Orijentisana Analiza i Dizajn

- **DIP** je ispunjen jer je ispunjen i SRP. Osim toga, klasa ovisi samo o apstraktnoj klasi Termin, čime su zadovoljene metrike koje zastupaju ovaj princip.

Klasa DodatniTermin:

SOLID principi se za ovu klasu manifestuju na sljedeći način:

- **SRP** je zadovoljen na isti način kako je zadovoljen i za klasu Trening.
- **OCP** je ispunjen time što se na ovu klasu mogu nadograđivati nove funkcionalnosti bez da se ona modificira.
- **LSP** je identično zadovoljen kao i za klasu Trening.
- **ISP** je ispoštovan jer klasa ne koristi metode koje nisu relevantne za nju tj. nije *fat tip* klase.
- **DIP** je ispunjen jer je ispunjen i SRP. Osim toga, klasa ovisi samo o apstraktnoj klasi Termin, čime su zadovoljene metrike koje zastupaju ovaj princip.

Klasa Ocjena:

Manifestovanje SOLID principa za ovu klasu je sljedeće:

- **SRP** je ispunjen jer klasa ima samo jednu odgovornost, a to je čuvanje informacija. Slično kao i klasa Aktivnost.
- **OCP** je ispoštovan jer se klasa može nadograditi metodama bez da se vrši njena modifikacija.
- **LSP** se ne manifestira jer ona ne učestvuje u nasljeđivanju.
- **ISP** se manifestuje na način da klasa ne zavisi od metoda koje nisu relevantne za nju.
- **DIP** je ispunjen jer od ove klase ne ovise druge klase.

Klasa ProgramAktivnosti:

Manifestovanje SOLID principa za ovu klasu je sljedeće:

- **SRP** je zadovoljen jer klasa ima samo jednu odgovornost, a to je da prikazuje programe aktivnosti koje korisnik može da izabere. Na osnovu toga, postoji samo jedan razlog da se ona promjeni.
- **OCP** je ispunjen jer se ova klasa može nadograditi sa novim metodama koje neće utjecati na postojeće stanje klase.
- **LSP** se ne ogleda na ovu klasu jer ova klasa nije izvedena iz apstraktne klase niti učestvuje u nasljeđivanju koje bi bilo neregularno.
- **ISP** se ovom klasom postiže tako što se koristi ograničen broj metoda što čini ovu klasu funkcionalnom, ali ne i podržanom za nerelevantne metode.
- **DIP** je ispunjen za ovu klasu jer od nje ne ovisi niti jedna druga klasa.

Klasa Rezultat:

Manifestovanje SOLID principa za ovu klasu je sljedeće:

- **SRP** je ispunjen jer ova klasa ima odgovornost samo nad atributima koji se odnose na rezultate, te je to jedini razlog za njenu izmjenu.
- **OCP** se ogleda u mogućnosti nadogradnje novih metoda koje neće narušiti postojeću strukturu. Upotrebom interface-a se može postići proširenja.
- **LSP** se ne manifestuje jer ova klasa ne učestvuje u nasljeđivanju.
- **ISP** je zadovoljen jer klasa nije opterećena metodama koje neće koristiti.
- **DIP** je također zadovoljen jer od ove klase ne ovise druge klase kao što je to slučaj kod apstraktnih klasa, čime je ovaj princip ispunjen.

Klasa Raspored:

Način na koji ova klasa ispunjava SOLID principe su:

- **SRP** je ispoštovan jer ova klasa ima zadatak samo da čuva podatke koji se odnose na raspored. Metode sa kojima ona raspolaže su metode za upravljanje rasporedom, što je samo jedan razlog za njenu izmjenu, čime je uslov ovog principa ispunjen.
- **OCP** se manifestuje na način da je klasa otvorena za nadogranju metoda ili nasljeđivanje, ali ne i za modifikacije.
- **LSP** se ne manifestuje za ovu klasu obzirom da se ona ne nasljeđuje, niti ona nasljeđuje druge klase.
- **ISP** je zadovoljen jer ova klasa posjeduje samo one metode koje su joj neophodne.
- **DIP** je zadovoljen jer ova klasa se ne nasljeđuje tako da zavisnosti od nje nisu prisutne.