Dijagram klasa i SOLID principi – Cinestvar

|  |
| --- |
|  |
| *Dijagram klasa* |

Implementacija SOLID principa

Unutar našeg dijagrama klasa trudili smo se da u što većoj mjeri prepoznamo potencijalna mjesta za implementaciju SOLID principa, te mislimo da se dosta naših rješenja može objasniti tom željom. U nastavku ćemo proći kroz svjesne SOLIDne odluke, i nadamo se da će ih pri analizi biti i nesvjesnih, te da će ih sveukupno biti više od neSOLIDnih.

**S:**  Single Responsibility

Princip jedinstvene odgovornosti je na dosta mjesta ispoštovan, što se može vidjeti u konceptu profila, iz Usera se nastavljaju specifični profili za korisnike (Neregistrovani, Pravno i Fizičko) koji služe za pristup informacijama profila: neregistrirani ima samo login i signup, jer je to jedino što on može raditi sa profilom, dok ostali koji su već ušli u svoje profile imaju opciju dajOpis za detaljne informacije o profilu. Od njih se razlikuje Radnik koji ima dodatnu funkciju za ažuriranje ponude, za koju mislimo da može ostati kako jeste jer je specifična za Radnik klasu, no može se premjestiti ukoliko se kosi sa principom. Rezerviši interfejs se sastoji samo od opcija za rezervaciju sale/karte, tako da se promjene u načinu rezervacije mogu brzo implementirati. Interface za Pogled daje funkcije prikaza za sve ponude koje imamo, međutim sve ponude se ponašaju kao kolekcija Stavki, koje imaju jednu metodu: dajOpis. Kako bi se što kompaktnije i sličnije odradile različite vrste ponuda, vrsta stavke se određuje običnim enumom, tako da je stavka zadužena samo za prikaz elementa menija.

**O:** Open-closed

Princip otvorenosti se ogleda ponovno u profilima, gdje je funkcionalnost rezervisanja odvojena u poseban interfejs, kako se ne bi sama klasa morala mijenjati ukoliko se način rezervacije promijeni (npr. ukoliko odjednom postane obavezno unijeti neku dodatnu informaciju). Iako dajOpisRezervacije može da bude unutar Rezervacije, istaknut je zbog jednostavnijeg baratanja, kako Rezervacija kao klasa ne radi sa licima, nego sa stringom koji ukazuje na rezervanta, tako da je dajOpis podložna modifikaciji zavisno koliko nam detaljne informacije trebaju.

**L:** Liskov

Ispunjenje Liskovinog principa možemo vidjeti na primjeru klasa Rezervacija i njenih derivata.

**I:** Interface

Interfejsi ovdje razdvajaju i rezervacije karata/sala i pregled ponuda od samih korisničkih profila, jer bi opcija pregleda ponude bila preobimna funkcija za profilne klase, a i ne bi imala pretežito smisla. Ovako, u zaseban interfejs smještamo rezervacije i za pravno i za fizičko lice, a što se tiče pregleda, on je povezan upotrebom sa neregistrovanim korisnikom, fizičkim i pravnim licem kao jedini način pristupa filmskoj ponudi.

**D:** Dependency inversion

I ovaj princip se može dobro vidjeti prateći strelice, većina zaista završava na interfejsima. Pošto se Ponuda neće mijenjati (u smislu da će uvijek biti kolekcija neke vrste stavki, makar se stavke mijenjale), na nju se smije nastavljati strelica gdje je potrebno, još jedna tako konkretna klasa je neregistrirani korisnik, jer je aplikacija koncipirana tako da ta klasa što prije izgubi mjesto u izvršavanju tj. da se korisnik prijavi ili registruje, međutim, tu možemo napraviti i alternativnu apstrakciju na neku klasu mušterija ukoliko u budućnosti bude bitno za razvoj.