

SRP

Dodan interfejs `ObracunPopusta` koji ce pomocu svojih metoda obracunavat popust nad instancom `KupovinaKarte`. Morali smo ovo uraditi jer je obracun popusta nesto sto bi se moglo nerijetko mijenjati, a svaka takva izmjena bi povlacila izmjenu implementacije klase `KupovinaKarte`, sto nastojimo izbjeći jer to nije primarna odgovornost pomenute klase.

Funkcionalnosti slanja karte na mail i slanja gift koda kao poklona prebacene u odgovornost kontrolera jer svakako ne bi trebale biti dio modela.

OCP

Direktno vezano za sve nabrojano iznad. Recimo, nesto za cim bi se cesto mogla ukazati potreba za promjenom je nacin obracuna popusta. Da nismo dodali navedeno interfejs, svaka izmjena bi vodila ka izmjeni i same klase koja sada samo koristi interfejs koji opisuje funkcionalnost za kojom se ukazala potreba za promjenu. Ovako, izmjene je potrebno samo napraviti u interfejsima. Slicno vrijedi i za druga dva dodana interfejsa.

LSP

Imamo samo jedno nasljeđivanje, koje ispunjava ovaj princip. Klase `KupovinaGiftKoda` i `KupovinaKarte` mogu se bez ikakvih problema iskoristiti u bilo kojem kontekstu u kojem se koristi klasa `Kupovina`, sve metode koje implementira klasa `Kupovina` primjenjive su na izvedene klase.

DIP

Vecinski je i ovaj princip ispostovan dodavanjem prethodno navedenih interfejsa. Klase koje „previse“ ovise od drugih klasa morale su biti dizajnirane na taj nacin zbog same prirode problema. Jedina mjesta na kojima je ovo primjenjivo vezana su za prethodno opisane funkcionalnosti slanja karte na mail, slanja gift koda kao poklona i obracun popusta, ali mislim da je tu sada sve u redu. Gdje god visi nivoi ovise o nizim, to je bilo potrebno zbog same prirode teme projekta.

Klasu `KupovinaKarte` smo proglasili apstraktnom jer ne bi smjela postojati mogucnost da je instanciramo. Ovime postizemo i to da konkretna izvedba `KupovinaKarte` ovisi o apstrakciji, sto ovaj princip nastoji zadovoljiti.

ISP

Za sada postoje dva razlicita tipa popusta, za registrovanog kupca koji je uclanjen u loyalty klub i za iskoristen gift kod. Sta ako bismo dodali jos neki oblik popusta u buducnosti? Određena metoda unutar `ObracunPopusta` ce primat listu karti i na osnovu toga koji oblik popusta je u pitanju (sto je direktno vezano za kupca) obracunavat popust. Posto smo `ObracunPopusta` realizovali kao interfejs, novi oblik popusta bismo mogli realizovati tako sto naslijedimo interfejs i implementiramo potrebne metode. Na taj nacin, klasa `ObracunPopusta` neće ovisiti o metodama koje joj nisu potrebne. Ovo je bila jedina prilika za ispostovati ISP, ostatak je korektan.