

## **SRP**

Dodan je interfejs `SlanjeKartiNaMail` koji koristi klasu `KupovinaKarte` jer funkcionalnost koju opisuje ovaj interfejs ne bi trebala biti odgovornost klase `KupovinaKarte`. Ako bismo željeli promijeniti način slanja karte na mail morali bismo mijenjati i implementaciju klase `KupovinaKarte`, a to ne bi trebao biti njen razlog za promjenu. Metode interfejsa `SlanjeKartiNaMail` nisu navedene jer su usko vezane za sam naziv.

Dodan interfejs `ObracunPopusta` koji će pomoću svojih metoda obračunavati popust nad instancom `KupovinaKarte`. Morali smo ovo uraditi jer je obračun popusta nešto što bi se moglo nerijetko mijenjati, a svaka takva izmjena bi povlačila izmjenu implementacije klase `KupovinaKarte`, što nastojimo izbjeći jer to nije primarna odgovornost pomenute klase.

Dodan interfejs `SlanjeGiftKoda` jer to apsolutno ne bi trebalo biti odgovornost klase `Kupac`, pogotovo što se način na koji se implementira mehanizam slanja poklona (u vidu gift koda) može često mijenjati.

## **OCP**

Direktno vezano za sve nabrojano iznad. Recimo, nešto za čim bi se često mogla ukazati potreba za promjenom je način obračuna 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 ukazuje potreba za promjenom. Ovako, izmjene je potrebno samo napraviti u interfejsima. Slično 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 način 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 obračun popusta, ali mislim da je tu sada sve u redu. Gdje god visi nivoi ovise o nizu, to je bilo potrebno zbog same prirode teme projekta.

## **ISP**

Za sada postoje dva različita tipa popusta, za registrovanog kupca koji je uključen u loyalty klub i za iskoristeni gift kod. Sta ako bismo dodali još neki oblik popusta u budućnosti? Određena metoda unutar `ObracunPopusta` će imati listu karti i na osnovu toga koji oblik popusta je u pitanju (što je direktno vezano za kupca) obračunavati popust. Komplikovano bi bilo usloznavati dijagram klase na osnovu toga da li je kupac član loyalty kluba i da li je unio gift kod jer to moramo uraditi sa dijelom koji se tiče klase `Kupac`, pa onda povezivati sa dva razdvojena interfejsa slična interfejsu `ObracunPopusta`. Postoji je obračun popusta realizovan u interfejsu, pomirit ćemo se s tim da ćemo morati vršiti korekcije ili dodavanje metoda unutar interfejsa. Ovo je bila jedina prilika za ispostovati ISP, ostatak je korektan.