**1.Princip pojedinačne odgovornosti**

Princip pojedinačne odgovornosti ( eng. Single Responsibility Principle-SRP) glasi:

*Klasa bi trebala imati samo jedan razlog za promjenu.*

S obzirom da se u dosta klasa javlja metoda generisiId(), mudro bi bilo napraviti odvojen interfejs sa ovom metodom, da bi se ispoštovao ovaj princip.

Trebalo bi dodati i klasu sa metodom za obračunavanje cijene karte, tj. računa na kasi, gdje će se uračunavati i popusti.

**2.Otvoreno zatvoren princip**

Otvoreno zatvoren princip ( eng. Open Closed Principle-OCP) glasi:

*Entiteti softvera (klase, moduli, funkcije) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije.*

Potrebno je napraviti jednu klasu BazaPodataka, u kojoj ćemo čuvati liste svih transakcija, uposlenih, projekcija, članova itd. Nakon toga će biti potrebno dodati odgovarajuće interfejse koji će omogućiti čitanje iz baze podataka i upisivanje u istu, bez potrebe za dodatnim modifikacijama u dijagramu klasa.

Potrebno je napraviti interfejs za obračunavanje popusta, u slučajevima kada je kupac na kasi ili e-Kupac već član.

Potrebno je restruktuirati postojeći dijagram klasa, tako da se klasa Projekcija nasljeđuje iz klase Film, potrebno je zatim korigovati klasu projekcija. U klasu Projekcija treba ubaciti i listu zauzetih sjedišta.

**3. Liskov princip zamjene**

Liskov princip zamjene (eng. Liskov Substitution Principle-LSP) glasi:

*Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.*

I ovaj princip se krši kod nasljeđivanja e-Kupca iz klase Kupac, ali će se riješiti nakon primjene principa 1.

Princip je ispoštovan kod Filma i Projekcije, a drugih nasljeđivanja nema.

**4. Princip inverzije ovisnosti**

Princip inverzije ovisnosti (eng. Dependency Inversion Principle-DIP) glasi:

A) *Moduli visokog nivoa ne bi trebali ovisiti od modula niskog nivoa. Oba bi trebalo da ovise od apstrakcija.*

B*) Moduli ne bi trebali ovisiti od detalja. Detalji bi trebali biti ovisni od apstrakcija.*

U našem dijagramu klasa, ovaj princip je prekršen za klase Kupac (kupuje karte direkno na kasi kina) i e-Kupac. Klasa e-Kupac je nasljeđena iz klase Kupac, koja nije apstraktna, već posjeduje atribut liste karata. Da bismo ovo popravili, trebali bismo napraviti apstraktnu klasu Kupac, koja neće sadržavati nijedan atribut, a iz nje bi se naslijedile klase KupacKasa i e-Kupac sa odgovarajućim atributima.

**5.Princip izoliranja interfejsa**

Princip izoliranja interfejsa ( eng. Interface Segregation Principle-ISP) glasi:

*Klijenti ne treba da ovise o metodama koje neće upotrebljavati.*

Ovaj princip je ispoštovan, s obzirom da nemamo „debelih“ klasa.

Potrebno je još dodati:

* atribut Datum u klasu Karta
* atribut Vrijeme u klasu Karta
* Formu LogoKina, koja će sadržavati ikonu
* Logo ubaciti u klasu Karta
* Const atribut popust = 0.1 u klasi Clan