**Tim:** SMS

**Članovi:** Purišić Muamer, Pašalić Tarik, Rovčanin Dželaluddin

# Principi dobrog dizajna primjenjeni na dijagramu klasa:

1. **Princip pojedinačne odgovornosti** : KLASA BI TREBALA IMATI SAMO JEDAN RAZLOG ZA PROMJENU

Prvi princip SOLID-a smo zadovoljili na način da smo težili ka fokusiranju poslova pojedinačnih klasa, tako da npr. postoje klase za štampanje prijemnica (**PrijemnicaPrint**) i radnih naloga (**RadniNalogPrint**) čiji je osnovni i jedini posao vođenje računa o štampanju ovih dokumenata, tako da ukoliko je potrebno promijeniti formu za štampanje dokumenata, dovoljno je modificirati samo ove klase, a ne i one koje ih koriste, u konkretnom slučaju **Prijemnica** i **RadniNalog.**

Brigu o načinu dodavanja novih klijenata, uposlenika, kao i partnerskih ovlaštenih servisa vode respektivno klase: **Klijenti**, **Uposlenici** i **ovlasteniServisi.**

2. **Otvoreno zatvoren princip :** ENTITETI SOFTVERA TREBALI BI BITI OTVORENI ZA NADOGRADNJU, ALI ZATVORENI ZA MODIFIKACIJE.

Ovaj princip smo nastojali ispoštovati činjenicom da nismo pravili „prelaznih“ veza između klasa, tako da promjena određene klase neće forsirati promjenu neke koja je vezana za nju.

3. **Liskov princip zamjene :** PODTIPOVI MORAJU BITI ZAMJENJIVI NJIHOVIM OSNOVNIM TIPOVIMA

Primjer ispunjenosti ovog principa je apstraktna klasa **Klijent** sa svojim izvedenim klasama **FizickoLice** i **PravnoLice**. Iako bi se fizičko lice kao klijent i bilo koji uposlenik mogli tretirati kao osobe, ipak je realizacija logičkim razdvajanjem u zasebne klase bolji i potpuniji način. Svaki klijent se tretira jednoznačno, bilo da je fizičko ili pravno lice, naravno uvažavajući pri tome razlike u atributima odgovarajućih klasa.

4**. Princip inverzije ovisnosti**

Ovaj princip smo pokušali uvažiti na onim mjestima gdje je npr. generalizacija imala smisla, odnosno gdje je bila moguća, konkretan primjer za to bi opet bila klasa **Klijent**.

5. **Princip izoliranja interfejsa**

Korisnik neće biti forsiran da koristi metode koje mu neće trebati, a to će biti konačno vidljivo kroz impementaciju korisničkog interfejsa, u kome će korisnik koristiti samo one metode koje se njega tiču.