

Princip S – svaka klasa treba imati samo jednu odgovornost

- Jedina klasa koja bi mogla narušiti ovaj princip je klasa Administrator, sve ostale klase imaju metode koje obavljaju samo jednu vrstu funkcionalnosti. Međutim, dodavanjem interfejsa *IDataBase* odvojili smo funkcionalnost interakcije sa vanjskim objektom od klase Administrator i ona sad ima samo jedan tip funkcionalnosti, dodavanje i modifikaciju objekata. S toga možemo zaključiti da je ovaj princip zadovoljen.

Princip O – klasa treba biti otvorena za nadogradnje, ali ne i za modifikacije

- Modifikacija bilo koje od klasa, koje se koriste u sklopu nekih drugih, ne bi uzrokovala izmjene na klasi koja je koristi. Uglavnom klase sadrže liste drugih klasa, te vrše manipulacije dodavanja ili izbacivanja elemenata liste, što ni u kojem slučaju ne bi narušavale izmjene bilo koje klase.

Princip L – sva nasljeđivanja moraju biti ispravno implementirana

- U našem dijagramu imamo dvije apstrakte klase MigrantskiCentar i Zahtjev iz kojih su nasljeđene podklase. S obzirom da je na svim mjestima gdje se koristi osnovni objekat moguće iskoristiti i izvedeni objekat i da to ima smisla. Zaključujemo da je ovaj princip zadovoljen.

Princip I – svaki interfejs mora obavljati samo jednu vrstu akcija

- S obzirom da imamo samo jedan interfejs i da je on zadužen za rad sa bazom, zaključujemo da je princip zadovoljen.

Princip D – pri nasljeđivanju od više klasa bazna klasa mora biti apstraktna

- U našem dijagramu imamo dva nasljeđivanja i oba su implementirana iz apstraktnih klasa, te zaključujemo da je ovaj princip zadovoljen.