1. Single responsibility principle (SRP)

Sve klase u modelu imaju pojedinačnu odgovornost i u svakoj postoji samo jedan razlog za promjenu. Svaka klasa čuva samo njoj neophodne informacije te ovo pravilo nije narušeno. U slučaju da postoji klasa Uposlenik na našem dijagramu klase sa metodama: dajBrojOdrađenihVožnji(), dajProsječnuOcjenu(), dajBrojVozača(), dajBrojTelefonaUposlenika(). SRP bi bio narušen jer ta klasa ne bi imala pojedinačnu odgovornost, već više njih, stoga bi bolje rješenje bilo napraviti više klasa (npr. StatistikaVožnje, StatistikaOcjene, itd.) koje će imati pojedinačnu odgovornost u kojima će postojati samo jedan razlog za promjenu.

2. Open – Closed principle (OCP)

OCP princip je ispunjen jer su klase otvorene za nadogradnju, ali zatvorene za modifikacije. Promjena u bilo kojoj klasi neće forsirati modifikaciju neke druge klase. Princip bi bio narušen ukoliko bi imali klasu za čitanje i pisanje informacija o vozačima u bazu podataka. Problem bi nastao kada bi željeli samo testirati rad klase Vozač a ne manipulisati nad stvarnom bazom podataka jer bi promjena nad klasom za rad sa bazom, odrazila se na klasu Vozač. Ostali bi dosljedni OCP principu tako što bi napravili interfejs za rad sa bazom, a zatim odgovarajuće izvedene klase za testni rad, ili rad sa stvarnom bazom podataka.

3. Liskov Substitution Principle (LSP)

Pravilo nije narušeno jer izvedene klase iz klase Osoba možemo upotrijebiti u bilo kojem kontekstu u kojem se očekuje klasa Osoba. Drugim riječima, sve izvedene klase iz klase Osoba su logički isto kao klasa Osoba, sa određenim specifičnostima, u vidu dodatnih atributa i metoda. Npr. ako očekujemo instancu klase Osoba kao parametar neke metode, možemo toj metodi proslijediti instancu klase Vozač ili instancu klase Klijent i program će raditi bez problema.

4. Interface – Segregation Principle (ISP)

Pravilo nije narušeno jer nemamo interfejs u dijagramu klasa. Ukoliko bi imali neki interfejs sa mnogo metoda (npr. metode koje se bave pribavljanjem statistike za različite aspekte naše aplikacije) poput dajProsječnuOcjenu() i dajBrojVozača(). Adminu će sigurno biti u interesu da koristi obje metode, dok će klijenta više zanimati prosječna ocjena svih vozača nego ukupan broj vozača (tj. neke metode će koristiti malo ili nikako). Da ne bi narušili ovo pravilo morali bi razdvojiti taj interfejs u više manjih interfejsa razdvojenih kao više bliske logičke cjeline.

5. Dependency – Inversion Principle (DIP)

DIP pravilo je ispunjeno jer nema ovisnosti o konkretnim klasama već o apstraktnim klasama i interfejsima. Bolje je ovisiti o apstrakcijama jer se oni generalno manje modifikuju nego stvarne klase. Npr. Vozač i Klijent su klase koje ovise o klasi Osoba koja je apstraktna.