

SOLID principi

1. Single Responsibility Principle - sa dijagrama klasa je vidljivo da je svaka klasa odgovorna za isključivo jednu stvar u programu (klasa RegistrovaniKorisnik sadrži listu treninga ali ne vrši njihovu modifikaciju nego se sve informacije i izmjene vrše putem klase Trening koja je odgovorna za svaku instancu treninga)
2. Open Closed Principle - svaka od klasa u programu je jako jednostavna i klase su u maloj međuovisnosti tako da je moguće dodavati nove funkcionalnosti bez velikih izmjena kako klase u kojoj dodajemo funkcionalnost tako i klase koje su u vezi s istom
Npr. ukoliko izvršimo izmjenu klase Vježba, to neće izmijeniti klasu Trening
3. Liskov Substitution Principle - buduci da nema izvedenih klasa nema slučaja gdje bi se bazna klasa trebala moći koristiti gdje i izvedena - iz ovog razloga, princip je ispostovan
4. Interface Segregation Principle - imamo 5 interfejsa koji korisniku omogućavaju pristup onim metodama koje su mu potrebne
5. Dependency Inversion Principle - aplikacija nije toliko složena da ima nešto što bi se moglo smatrati apstrakcijom a ne mogu se ni primjeniti high-level i low-level moduli pa je princip zadovoljen