

# SOLID principi

## 1. Single-Responsibility Principle

Cilj ovog principa je da se svaka klasa bavi samo onim što se tiče direktno nje. Klasa Korisnik je odgovorna za upravljanje informacijama o korisniku. Klasa Administrator nasljeđuje Korisnik i dodaje funkcionalnosti upravljanja proizvodima, refundacijama i otkazivanjima narudžbi. Klasa Proizvod je osnovna klasa za sve proizvode i sadrži zajedničke attribute i funkcionalnosti. Klase Odjeća i Obuća nasljeđuju Proizvod i dodaju specifične attribute za odjeću i obuću. Klasa Poslovnica sadrži informacije o poslovnicama. Klasa Recenzija upravlja informacijama o recenzijama proizvoda. Klasa Narudžba upravlja informacijama o narudžbama i procesu narudžbe. NaciniPlacanja, KategorijeObuce i KategorijeOdjece omogućuju odabir opcija za načine plaćanja, kategorije obuće i odjeće.

## 2. Open-Closed Principle

Kod ovog principa je glavno da su klase otvorene za proširenje, ali zatvorene za izmjene. Direktno u našem primjeru to znači, da klase poput Odjeća i Obuća mogu se proširiti bez potrebe za izmjenom osnovne klase Proizvod.

## 3. Liskov Substitution Principle

Kod ovog principa je glavno da su klase otvorene za proširenje, ali zatvorene za izmjene. Direktno u našem primjeru to znači, da klase poput Odjeća i Obuća mogu se proširiti bez potrebe za izmjenom osnovne klase Proizvod.

## 4. Interface Segregation Principle

Ovaj princip zahtijeva da klijenti ne trebaju ovisiti o metodama koje neće koristiti. U našem slučaju ovaj princip je ispunjen jer sve klase obavljaju samo one aktivnosti koje su za njih vezane. Na primjer, kod Administrator bi implementirali metodu za upravljanje proizvodima, dok kod Korisnika ne bi.

## 5. Dependency Inversion Principle

Ovaj princip zahtijeva da klase trebaju zavisiti o apstraktnim klasama. Klasa Narudžba ovisi o abstraktnoj klasi Proizvod, umjesto o konkretnim klasama Odjeća i Obuća. Time se osigurava da Narudžba može sadržavati različite vrste proizvoda bez potrebe za ovisnostima o konkretnim klasama.