

SOLID principi

Single responsibility principle

Ovaj princip je ispoštovan na više mjesta uvođenjem pomoćnih klasa čija je uloga da rasterete “glavne” klase u sistemu. Primjer toga je korištenje klase *Popust* za izračun, tako da ta klasa radi samo razne izračune vezane za popust a narudžbe ne moraju više se baviti tim već samo koriste dobivene rezultate i vrše proces naručivanja.

Open-closed principle

Kreiranjem apstrakne klase *User* smo omogućili da u budućnosti po potrebi možemo dodavati nove vrste korisnika (premium korisnici i sl.) bez da mijenjamo dosadašnju implementaciju, već samo dodajući novu.

Liskov-substitution principle

Klase *Customer* i *Guest* ispunjavaju Liskov princip zamjene tako što su naslijeđene iz zajedničke klase, te se obe klase mogu proizvoljno zamjenjivati u zajedničkim aktivnostima (ubacivanje u korpu).

Interface-segregation principle

Ovaj princip ispunjavamo razbijanjem klasičnog *Person* interfejsa na 2, *User* i *Person*. Ovim postizemo da naša *Customer* klasa ima sve potrebne metode i attribute tako što implementira 2 interfejsa, dok klasa *Administrator* ne mora implementirati *Person* klasu, te se oslobađa od povezanosti za korpu koju ta klasa ne treba da ima.

Dependency-inversion principle

U okviru shopping cart-a je ovaj princip ispoštovan na način da shopping cart zavisi samo od toga da je u pitanju neki korisnik shopa (apstraktna klasa *User*), a ne i od same interpretacije (da li je ta klasa *guest*, korisnik ili u budućnosti potencijalno nešto treće).