

SOLID principi

1. Single responsibility principle

Svaka klasa treba da ima jednu jedinu odgovornost.

Klasa Korisnik sadrži samo podatke o korisniku (ime, prezime, email, šifra, itd.) i metode koje se tiču korisnika.

Klasa Obavijest je zadužena samo za obavijesti (sadržaj, vrijeme slanja), a ne za korisničke podatke ili oglase.

2. Open closed principle

Klasa treba da bude otvorena za proširenje, ali zatvorena za izmjene.

Enumeracije Uloge, Status, Obavjestenje omogućavaju lahko dodavanje novih vrijednosti bez izmjene postojećeg koda.

3. Lisk substitution principle

Objekti izvedenih klasa treba da mogu da zamijene objekte bazne klase bez narušavanja funkcionalnosti.

Naš dizajn ne koristi nasljeđivanje tako da je ovaj princip ispunjen.

4. Interface segregation principle

Klijenti ne trebaju da zavise od interfejsa koje ne koriste.

U UML dijagramu to nije eksplicitno prikazano, ali se kroz postojanje klasa poput OglasKorisnik i ObavijestKorisniku može zaključiti da su funkcionalnosti pravilno razdvojene i da nijedna klasa nije opterećena metodama koje joj nisu potrebne.

5. Dependency inversion principle

Zavisnosti treba da budu prema apstrakcijama, ne konkretnim klasama.

Dizajn koristi posredne klase koje služe kao sloj između glavnih entiteta, čime se smanjuje direktna zavisnost i povećava fleksibilnost sistema.

Univerzitet u Sarajevu
Elektrotehnički Fakultet
Objektno Orijentisana Analiza i
Dizajn

