SOLID Objašnjenje

1. Single Responsibility Principle (SRP)

Princip jedinstvene odgovornosti navodi da bi klasa trebala imati samo jedan razlog za promjenu, što znači da bi trebala imati samo jednu odgovornost.

Korisnik: Odgovoran je samo za upravljanje informacijama o korisnicima sistema. Ova klasa ne upravlja direktno rezervacijama ili recenzijama, već služi kao veza za profil korisnika i njegove aktivnosti unutar sistema.

Profil Korisnika: Čuva personalne informacije korisnika, odvojene od njegovog korisničkog naloga (Korisnik). To omogućava jednostavnije upravljanje korisničkim podacima bez miješanja u autentifikaciju ili druge korisničke funkcije.

2. Open/Closed Principle (OCP)

Princip otvorenosti/zatvorenosti kaže da bi softver trebao biti otvoren za proširenje, ali zatvoren za modifikaciju.

Resurs: Dizajniran je kao bazna klasa koja omogućava proširenje na specifične tipove resursa kao što su Nekretnina i Vozilo. Nove vrste resursa mogu se lako dodati bez izmjena postojećeg koda.

Nekretnina i Vozilo: Nasljeđuju opšte osobine i funkcije klase Resurs, ali dodaju specifične karakteristike relevantne za svaki tip resursa.

3. Liskov Substitution Principle (LSP)

Princip supstitucije Liskov navodi da bi objekti u programu trebali biti zamjenjivi sa instancama njihovih podtipova bez uticaja na tačnost programa.

Nekretnina i Vozilo: Kao podklase klase Resurs, obezbeđuju da sve operacije definisane za Resurs mogu biti jednako validno izvršene sa Nekretninom ili Vozilom. To omogućava generalizaciju funkcija kao što su prikaz ili rezervacija, koristeći referencu na Resurs.

4. Interface Segregation Principle (ISP)

Princip segregacije interfejsa sugeriše da klijenti ne bi trebali biti primorani da zavise od interfejsa koje ne koriste.

Svaka klasa u sistemu definisana je sa specifičnim zadacima i funkcijama koje su relevantne samo za te klase. Na primer, funkcije upravljanja rezervacijama su odvojene od korisničkog profila i recenzija, čime se izbjegava nepotrebna zavisnost među klasama.

5. Dependency Inversion Principle (DIP)

Princip inverzije zavisnosti navodi da bi module visokog nivoa trebalo da zavise od apstrakcija, a ne od konkretnih implementacija nižeg nivoa.

Rezervacija: Ova klasa interaguje sa Resursom preko njegovog interfejsa, ne zavisno od specifičnog tipa Resursa. To omogućava da Rezervacija ostane nezavisna od konkretnih klasa kao što su Nekretnina ili Vozilo.

Recenzija: Slično, interagira sa interfejsom Resursa, što omogućava dodavanje recenzija za bilo koji tip resursa bez direktnog pozivanja funkcija specifičnih za Nekretninu ili Vozilo.