

Paterni ponašanja

1. Observer Pattern (Posmatrač)

Koristi se kada više objekata treba biti obaviješteno o promjeni stanja drugog objekta.

U primjeru mogli bi obavijestiti sve korisnike koji su rezervisali određenu projekciju kada dođe do promjene termina ili otkazivanja.

Uloge:

Subject (ProjekcijaObservable): Projekcija koja može biti izmijenjena (npr. promjena termina).

Observers (KorisnikObserver): Korisnici koji su rezervisali karte za tu projekciju.

2. Strategy Pattern (Strategija)

Omogućava zamjenu različitih algoritama u toku izvršavanja programa.

Implementacija:

Interfejs CijenaStrategija sa metodom izracunajCijenu(Karta).

Klase StandardnaCijena, StudentskaCijena, DjecijaCijena implementiraju različite strategije.

Rezervacija koristi strategiju pri kreiranju konačne cijene.

3. State Pattern (Stanje)

Omogućava promjenu ponašanja objekta u zavisnosti od njegovog trenutnog stanja.

Uloga u primjeru: različita ponašanja rezervacije u zavisnosti od njenog trenutnog stanja (Kreirana, Plaćena, Otkazana, Završena).

4. Iterator Pattern (Iterator)

Omogućava prolazak kroz kolekciju bez poznavanja njene strukture.

Primjena: iteriranje kroz listu svih karata u jednoj rezervaciji ili kroz sve dostupne projekcije u danu.

5. Template Method Pattern

Definiše kostur algoritma, a podklase redefinišu određene korake.

Primjena: izračunavanje cijene rezervacije koja se sastoji iz više koraka (osnovna cijena, popust).

6. Chain of Responsibility Pattern (Lanac odgovornosti)

Svaki objekat u lancu odlučuje da li obrađuje zahtjev ili ga prosljeđuje dalje.

Primjena: validacija rezervacije: validacija dostupnosti sjedista, validacija QR koda, validacija termina.

7. Mediator Pattern (Posrednik)

Centralizuje komunikaciju između objekata kako ne bi bili direktno povezani.

Primjena: koordinacija između korisnika, rezervacije i karte.

Implementacija:

Klasa RezervacijaMediator upravlja interakcijama: korisnik pravi rezervaciju → sistem rezerviše mjesto → generiše QR kod

8. Visitor Pattern (Posjetilac)

Dodaje nove operacije postojećim klasama bez njihovog mijenjanja.

Primjena: statistike tj. broj prodatih karata, broj slobodnih mjesta po projekciji, broj rezervacija po korisniku.

9. Interpreter Pattern

Kreira jezik za specifičnu domenu i omogućava njegovo tumačenje.

Primjena: Korisnik unosi upite tipa: film: "Inception" datum < 15.06 da bi pretražio projekcije.

10. Memento Pattern (Memento)

Omogućava spremanje i vraćanje starog stanja objekta.

Primjena: undo funkcionalnost kada korisnik promijeni lozinku ili e-mail.

11. Command Pattern

Svaka akcija se pretvara u objekat, što omogućava redoslijed izvršavanja, poništavanje i spremanje.

Omogućiti redoslijedno izvršavanje, poništavanje i ponovno izvršavanje korisničkih akcija u procesu upravljanja rezervacijama, npr. kada korisnik: doda kartu, ukloni kartu, potvrdi rezervaciju, poništi prethodni korak