

Paterni ponašanja

1. Strategy patern

Izdvađa algoritam iz matične klase i uključuje ga u posebne klase. Različiti algoritmi za obračun cijene rezervacije (standardna, promotivna, VIP) mogu se implementirati kao strategije. Rezervacija koristi odgovarajuću strategiju u zavisnosti od tipa korisnika ili akcije.

2. State patern

Mijenja način ponašanja objekata na osnovu trenutnog stanja. Ponašanje objekta Rezervacija zavisi od njenog trenutnog stanja (KREIRANA, U_TOKU, OTKAZANA). Promjena stanja mijenja dozvoljene akcije (npr. nije moguće otkazati već završenu rezervaciju).

3. Template method patern

Omogućava algoritmima da izdvoje pojedine korake u podklase. Procesiranje plaćanja može slijediti fiksni korak-po-korak algoritam (validacija kartice, slanje transakcije, ažuriranje statusa). Konkretno implementacije mogu redefinisati pojedine korake (npr. različite platne metode).

4. Observer patern

Uspostavlja relaciju između objekata takvu da kad se stanje jednog objekta promijeni svi vezani objekti dobiju informaciju. Kada se status rezervacije ili plaćanja promijeni, sistem notifikacija (email, SMS) može biti automatski obaviješten. Rezervacija i Plaćanje objekti mogu imati posmatrače koji reaguju na ove promjene.

5. Iterator patern

Omogućava pristup elementima kolekcije sekvencijalno bez poznavanja interne strukture kolekcije. Osoba može iterirati kroz svoju listu rezervacija ili plaćanja. Iterator omogućava jednostavan i ujednačen pristup tim kolekcijama

6. Command patern

Zahtjev pretvara u samostalni objekat. Na taj način se mogu jednostavno proslijediti, staviti u red čekanja ili zabilježiti zahtjevi i podržati operacije poput undo/redo. Akcije poput rezervisiVozilo, procesirajPlaćanje i odgovoriNaZahtjev mogu se enkapsulirati kao komandni objekti. Time se omogućava fleksibilno izvršavanje, zapisivanje u log ili undo/redo mehanizam.

7. Mediator patern

Enkapsulira protokol za komunikaciju među objektima dozvoljavajući da objekti komuniciraju bez međusobnog poznavanja interne strukture objekta. Umjesto da Osoba, Rezervacija i Plaćanje direktno komuniciraju, posrednik može koordinirati njihovu interakciju. Na primjer, kada se rezervacija kreira, mediator može automatski pokrenuti proces plaćanja.

8. Chain of responsibility patern

Predstavlja listu objekata, ukoliko objekat ne može da odgovori prosljeđuje zahtjev narednom u nizu. Korisnički zahtjev u Podrška može proći kroz lanac agenata (bot, agent prvog nivoa, supervizor). Svaki agent odlučuje da li će obraditi zahtjev ili ga proslijediti dalje.

9. Visitor patern

Definira i izvršava nove operacije nad elementima postojeće strukture ne mijenjajući samu strukturu. Iskoristio bi se za dodavanje novih operacija nad rezervacijama, plaćanjima ili vozilima (npr. generisanje izvještaja). Time se mogu obilaziti ti objekti bez izmjene njihovih klasa i fleksibilno proširivati funkcionalnost sistema.

10. Interpreter patern

Podržava interpretaciju instrukcija napisanih za određenu upotrebu. Ako korisnici žele pretraživati vozila ili rezervacije pomoću složenih uslova (npr. "vozila novija od 2020. i cijena ispod 50€"), može se primijeniti Interpreter. Na taj način se omogućava definisanje i tumačenje upitnog jezika za dinamičko filtriranje podataka.

11.Memento patern

Omogućava spašavanje internog stanja nekog objekta van sistema i njegovo ponovno vraćanje. Prije izmjene statusa rezervacije ili podataka o vozilu, može se sačuvati prethodno stanje. Time se omogućava vraćanje na prethodnu verziju u slučaju greške.