

Paterni ponašanja

1. **Strategy patern**

Ovaj patern omogućava izdvajanje algoritama u zasebne klase, čime se olakšava njihova zamjena tokom izvršavanja programa.

U našoj aplikaciji primjenjujemo ga prilikom sortiranja proizvoda – različiti tipovi proizvoda se kategorizuju, kako bi korisnik mogao jednostavnije pronaći ono što traži.

2. **State patern**

State patern omogućava objektu da mijenja svoje ponašanje u zavisnosti od trenutnog stanja.

U našoj implementaciji koristi se za upravljanje stanjem narudžbe, koja može biti „obrađena“ ili „neobrađena“. Na osnovu tog stanja zaposlenici dobijaju uvid u narudžbe koje je potrebno obraditi.

3. **Template Method patern**

Ovaj patern definiše osnovni tok algoritma, ali omogućava podklasama da prilagode pojedine korake.

U aplikaciji ga možemo koristiti za obradu različitih tipova korisnika (administrator, zaposlenik, itd.), gdje svaka uloga ima specifičan pristup. Takođe, patern se može primijeniti i kod metoda plaćanja – na primjer, razlikovanjem koraka između kartičnog plaćanja i plaćanja prilikom preuzimanja.

4. **Iterator patern**

Iterator patern omogućava pristup elementima kolekcije bez otkrivanja njene unutrašnje strukture. U aplikaciji se koristi za pregled i prikaz liste proizvoda na jasan i jednostavan način.

5. **Observer patern**

Ovaj patern uspostavlja vezu između objekata tako da promjena u jednom objektu automatski obavještava sve zavisne objekte.

U našem slučaju, patern se koristi za notifikaciju korisnika kada proizvod koji su dodali u korpu više nije dostupan.

6. **Command patern**

Command patern enkapsulira zahtjeve kao objekte, omogućavajući fleksibilno izvršavanje operacija, njihovo odlaganje, logovanje ili podršku za poništavanje akcija. Primjenjen je kod opcije

„Otkazi narudžbu“, gdje korisnik može poništiti proces narudžbe i vratiti se na prethodno stanje korpe prije konačne potvrde.

7. **Memento patern**

Memento patern omogućava čuvanje i vraćanje prethodnog stanja objekta, bez narušavanja principa enkapsulacije.

U aplikaciji se koristi za funkcionalnost „Dodaj u korpu“, gdje korisnik ima mogućnost „undo“ opcije, vraćajući prethodno stanje korpe u slučaju greške ili promjene odluke.