

Paterni ponašanja

Stragery patern

Stragery patern izdvaja algoritam iz matične klase i uključuje ga u posebne klase. Pomoću ovog paterna se omogućava klijentu izbor jednog algoritma iz familije algoritama.

U našem slučaju, može se koristiti za fleksibilno sortiranje proizvoda iz ponude. Korisnik može da izabere način sortiranja – po cijeni (rastuće/opadajuće), dostupnosti, ili nutritivnoj vrijednosti (npr. kalorije). Na taj način omogućava kupcu jednostavan odabir najpogodnijeg načina sortiranja usluga. Lako je proširivo - može se dodati nova strategija bez izmjene postojećeg koda.

State patern

State patern mijenja način ponašanja objekata na osnovu trenutnog stanja. Korisnik može naručiti proizvode, pratiti status narudžbe, i dobiti obavještenja o promjenama. Narudžba može da ima više stanja, od kreiranja pa sve do dostave. Dodavanje novog stanja ne utiče na postojeći kod. Svako stanje zna samo svoje naredno stanje i ponašanje - što vodi ka čistom i čitljivom kodu.

Template Method patern

Omogućava izdvajanje određenih koraka nekog kompleksnog algoritma u određene podklase. Izdvajaju se mali dijelovi operacija koji se mogu implementirati na različite načine.

Za proces narudžbe koji uvijek ima iste osnovne korake (verifikacija, plaćanje, potvrda), ali koji može varirati u detaljima zavisno od vrste proizvoda (npr. torta, kolač, krofna).

Observer patern

Observer uspostavlja relaciju između objekata tako da promjene stanja automatski obavještavaju sve zainteresovane strane. Primjena u slastičarnoj može biti obavještavanje korisnika o promjenama. Registrovani korisnici automatski dobiju notifikaciju (e-mail, poruka) kada se pojavi nova ponuda (npr. popust na tortu iz praznične ponude).

Iterator patern

Iterator patern služi za efikasno i jednostavno pretraživanje kolekcije podataka, bez da otkrivamo unutarnju strukturu kolekcije. Za pretragu i pregled proizvoda u ponudi bez potrebe da znamo kako su proizvodi implementirani u pozadini (lista, skup, mapa...). Iterator omogućava prolazak kroz kolekciju kolača, torti, itd.

Chain of Responsibility patern

Omogućava razdvajanje jednog kompleksnog procesa obrade na više objekata koji na različite načine procesiraju primljene podatke. Koristan je za obradu narudžbi gdje se zahtjev proslijeđuje kroz lanac odgovornosti: validacija narudžbe → provjera zaliha → plaćanje → dostava. Svaki handler procesira dio zahtjeva i proslijeđuje ga dalje.

Mediator patern

Mediator služi kao međuobjekat da bi se izbjeglo direktno povezivanje velikog broja objekata. Administrator (na kasi) kreira narudžbu i preko mediatora obavještava zaposlenika koji je priprema, bez da komuniciraju direktno. Na taj način se omogućava lakše održavanje sistema i fleksibilno dodavanje novih učesnika poput dostavljača ili sistema za notifikacije.

Interpreter patern

Interpreter patern podržava interpretaciju specifičnih korisničkih zahtjeva, omogućavajući fleksibilniju pretragu. Pogodan je za naprednu pretragu proizvoda gdje korisnik unosi kompleksnije izraze. Patern interpretira korisnički upit prema definisanoj gramatici pretrage.