Paterni ponašanja (behavioral design patterns)

U ovom dokumentu ćemo detaljno obratiti paterne ponašanja (teoretsko objašnjenje i primjenu) koji su potrebni za pravilno funkcionisanje projekta – „Art Gallery Management&Bidding System“. Paterni kroz koje ćemo proći su: Observer, Command, State, Strategy i Template Method.

OBSERVER PATTERN

Koristi se kada želimo da više objekata automatski budu obaviješteni o promjenama stanja nekog objekta. Najčešće se koristi u implementaciji pretplate/obavještavanja (npr. UI komponente koje se ažuriraju kada se promijeni model).

U našem sistemu, Observer patern je naročito koristan za asinhrono obavještavanje korisnika o promjenama koje ih se tiču. Na primjer, kada je korisnik postavio ponudu na određenu aukciju, on može biti "pretplaćen" na događaje vezane za tu aukciju. Kada neko drugi postavi veću ponudu ili kada aukcija bude zatvorena, sistem automatski šalje obavijest svim korisnicima koji su u nekom trenutku bili aktivni učesnici.

Pored toga, Observer patern se koristi i za obavještavanje o novim aukcijama, promjenama termina, obavijesti o finalizaciji kupovine i slanje QR koda nakon završetka transakcije. Na ovaj način se izbjegava potreba za ručnim provjeravanjem promjena, jer se korisnici automatski informišu kada dođe do događaja koji se odnosi na njih.

COMMAND PATTERN

Enkapsulira zahtjev (komandu) kao objekat, čime omogućava parametrizaciju akcija, redoslijed izvršavanja, poništavanje (undo) i evidentiranje komandi. Koristi se za rad sa akcijama koje se mogu izvršiti, odgoditi ili poništiti.

Command patern omogućava enkapsulaciju zahtjeva (komandi) korisnika kao objekata, što omogućava fleksibilnu obradu i upravljanje tim zahtjevima. U ovom sistemu, komandni obrazac se može koristiti kod upravljanja ponudama korisnika, gdje svaka akcija korisnika — postavljanje ponude, ažuriranje ili povlačenje ponude — može biti predstavljena kao komanda.

Ova struktura omogućava jednostavno logovanje svih akcija korisnika, podršku za eventualno poništavanje posljednje akcije (undo), kao i obradu komandi u određenom redoslijedu, što je korisno ako sistem postane distribuiran ili treba da podrži obradu velikog broja zahtjeva. Takođe, command patern omogućava proširivanje — lako se može dodati nova vrsta komande bez mijenjanja postojećeg koda.

STATE PATTERN

Omogućava objektu da promijeni svoje ponašanje kada mu se promijeni unutrašnje stanje. Umjesto if-else grananja za stanje, koristi se djeljenje na različite objekte koji predstavljaju stanja.

State patern omogućava da objekat mijenja svoje ponašanje na osnovu trenutnog internog stanja. U našem sistemu, primarni kandidat za primjenu ovog paterna je aukcija. Tokom životnog ciklusa aukcije, ona prolazi kroz nekoliko stanja:

- Priprema - još nije aktivna
- Aktivna - korisnici mogu nuditi cijenu
- Zatvorena - aukcija je završena, više ne prihvata ponude
- Finalizirana - transakcija izvršena, kupljena umjetnina

Svako od ovih stanja može imati različita ponašanja u smislu šta sistem dozvoljava korisniku da uradi. Na primjer, u stanju "Aktivna", korisnici mogu nuditi više, dok je u stanju "Zatvorena" to onemogućeno. State patern omogućava dinamičku promjenu ponašanja objekta Aukcija bez potrebe za velikim uslovima grananja.

STRATEGY PATTERN

Definiše porodicu algoritama, enkapsulira ih i omogućava njihovu zamjenu u toku izvođenja. Koristi se kada želimo da dinamički biramo algoritam koji će se koristiti bez izmjene konteksta u kojem se koristi.

Strategy patern se može koristiti kod procjene cijene umjetnine, gdje postoji više mogućih algoritama za izračunavanje estimirane vrijednosti. Na primjer:

- Jedna strategija može procjenjivati cijenu na osnovu tehničkih karakteristika (tehnika, format, materijal)
- Druga na osnovu reputacije autora i tržišnih trendova
- Treća može koristiti istorijske podatke sa prethodnih aukcija.

Svaka od ovih metoda se može enkapsulirati kao zasebna strategija, što omogućava lako proširenje sistema bez potrebe za izmjenom osnovne logike. Administrator ili kritičar može odabrati odgovarajuću strategiju na osnovu tipa umjetnine, a sam sistem ostaje fleksibilan i prilagodljiv novim pravilima procjene.

TEMPLATE METHOD PATTERN

Definiše kostur algoritma u nadklasi, ali omogućava podklasama da redefinišu određene korake bez mijenjanja strukture algoritma. Koristi se za postavljanje osnovne logike uz fleksibilnost da se dio ponašanja prilagodi.

Template Method patern može se koristiti za kreiranje šablona ponašanja koji ima fiksni tok koraka, ali omogućava da neki od tih koraka budu redefinisani u podklasama. Konkretan primjer u ovom sistemu jeste proces registracije korisnika ili slanja email obavijesti.

Na primjer, prilikom registracije, svi korisnici prolaze kroz iste osnovne korake:

1. Unos podataka
2. Validacija
3. Snimanje u bazu
4. Slanje potvrde putem e-maila

Međutim, moguće je da se korisnici različitih tipova (kupac, kritičar, administrator) razlikuju po dodatnim koracima, što se može omogućiti definisanjem šablona u baznoj klasi, a specifičnih detalja u podklasama. Na taj način se izbjegava dupliciranje logike i obezbjeđuje organizovana struktura.