

PATERNI PONAŠANJA

Strategy patern

U slučaju kada želimo izdvojiti algoritme u posebne klase, koristimo Strategy patern. Pogodno ga je koristiti ukoliko za isti problem postoje različiti algoritmi koji vode ka rješenju, a klijent je onaj koji bira koji od algoritama želi koristiti. U ovom paternu je, kao i u ostalim paternima, implementacija algoritma u potpunosti neovisna od klijenta.

U nekim budućim nadogradnjama sistema, ovaj patern bi mogao biti primijenjen prilikom rezervacije i kupovine karata. Konkretna implementacija bi se sastojala u tome da se zaposleniku sistema, ukoliko je na sistem prijavljen sa tom ulogom, omogući određen popust prilikom rezervacije i kupovine karata.

U trenutnoj postavci sistema je ovaj patern moguće primijeniti za odabir načina plaćanja, tj. da će plaćanje biti različito obrađeno u zavisnosti od toga da li korisnik nakon izvršene rezervacije odluči da li će kartu platiti online platnom karticom ili pak na licu mjesta pouzećem.

State patern [implementacija]

State patern je karakterističan po tome što omogućava promjenu ponašanja objekta, u zavisnosti od njegovog trenutnog stanja. Pogodan je ukoliko objekat može imati više stanja i ukoliko često dolazi do promjene trenutnog stanja. Patern zadovoljava Single Responsibility i Open / Closed princip.

U našem sistemu ovaj patern možemo iskoristiti za obrađivanje karata i sjedišta, te im postavljati određeni status. Na primjer, status karte može biti rezervirana, kupljena ili slobodna, dok status sjedišta može biti slobodno, zauzeto ili blokirano (ukoliko se koristi kao dio scenografije ili je u nekom slučaju pokvareno).

Template Method patern

Patern je pogodan za upotrebu kada postoji potreba za izdvajanjem određenih koraka algoritma u odvojene podklase, što ne utječe na strukturu samog algoritma, ali omogućava implementaciju pojedinačnih dijelova na različite načine.

Ovaj patern se može primijeniti prilikom prijave na sistem za različite korisnike. Ukoliko se na sistem prijavljuje zaposlenik, bit će zatraženi različiti podaci u odnosu na one kada se na sistem prijavljuje obični korisnik.

Observer patern

Uloga ovog patern je da uspostavi relaciju između objekata tako da prilikom promjene jednog objekta, drugi povezani objekti o tome budu obaviješteni.

U našem sistemu bi ovaj patern mogao biti implementiran sa Newsletter-om. Kako se korisnici prijavljuju za primanje Newslettera, ovaj patern bi mogao biti iskorišten za slanje podataka o ažuriranju repertoara svim zainteresiranim korisnicima. Također, mogao bi se implementirati i za obrade situacija kada za neku predstavu više nema slobodnih karata koje se mogu rezervirati, pa se korisnik može prijaviti da bude obaviješten ukoliko se neka od rezervacija otkaže.

Iterator patern [implementacija]

Iterator patern omogućava kretanje kroz kolekciju podataka bez da nužno poznaje detalje o strukturi kolekcije. Iterator može implementirati različite algoritme koji mu omogućavaju kretanje kroz kolekcije. Kretanje je omogućeno korištenjem zajedničkog interfejsa, a istovremeno se kroz kolekciju može kretati više objekata tipa Iterator.

U našem sistemu Iterator patern bi mogao biti implementiran za potrebe efikasnog prolaženja kroz postojeće predstave. Kako je pregled najbolje ocijenjenih predstava jedna od funkcionalnosti koja je omogućena na našoj stranici, potrebno je ažurirati ovu listu svaki put kada se dodaju nove ocjene u bazu. U tom slučaju, potrebno je iterirati kroz sve aktuelne predstave i generisati nove preporuke. Implementacijom Iterator patern, ovaj pregled bi se značajno unaprijedio i ubrzao.

Command patern

Command patern pretvara zahtjev u samostalni objekt koji se koristi za enkapsuliranje svih informacija potrebnih za izvršenje radnje ili pokretanje događaja u kasnijem trenutku. Ove informacije uključuju ime metode, objekt koji posjeduje metodu i vrijednosti za parametre metode. Ova transformacija omogućava prosljeđivanje zahtjeva kao argumenata metode, odlaganje ili stavljanje u red za izvršenje zahtjeva i podržavanje operacija koje se inače ne bi mogle izvršiti.

Kako u našem sistemu imamo dva načina plaćanja, online ili na licu mjesta, gdje opet može platiti karticom ili kešem, ovaj patern se može iskoristi za kreiranje jedinstvenog zahtjeva koji predstavlja plaćanje, a informacije o načinu plaćanja se popunjavaju u zavisnosti od odabranog načina.