

Paterni ponašanja

1.STRATEGY PATTERN

Strategy patern izdvaja algoritam iz matične klase i uključuje ga u posebne klase. Pogodan je kada postoje različiti primjenjivi algoritmi za neki problem. Strategy patern omogućava klijentu izbor jednog od algoritama iz familije algoritama za korištenje. Strategy patern je svoje mjesto u našem projektu našao u dijelu u kojem korisnik pretražuje ponudu automobila. Nakon što korisnik unese željene specifikacije, bira način sortiranja (rastući ili opadajući) po cijeni automobila. Pomoću ovog paterna neće biti teško dodati nove metode za sortiranje ukoliko se za to ukaže potreba.

2.STATE PATTERN

State pattern je dinamička verzija Strategy paterna. Objekat mijenja način ponašanja na osnovu trenutnog stanja. Ovaj patern nismo implementirale u našem sistemu, ali ukoliko bismo to htjele uraditi mogle bismo dodati intefejs IStanjeZahtjeva te klase ZahtjevNaCekanju, OdobrenZahtjev, OtkazanZahtjev i KompletiranZahtjev koje bi implementirale taj intefejs, i metodu postaviStatusZahtjeva kojom bi status prelazio u navedeno stanje.

3.TEMPLATE METHOD

Patern Template method patern služi da omogući izmjene ponašanja u jednom ili više dijelova. Najčešće se primjenjuje kada se za neki kompleksni algoritam uvijek trebaju izvršiti isti koraci, no pojedinačne korake moguće je izvršiti na različite načine. Nismo implementirale ovaj patern u našem sistemu, ali to bi mogle uraditi kada bi uvele klasu SortiranjeKorisnika, kao i klase AbecednoSortiranje, BrojRezervacijaSortiranje koje bi predstavljale sortiranje korisnika po imenima i po dosadašnjem broju rezervacija ili klasu SortiranjeAuta i klase CijenaSort, BrojPređenihKilometaraSort, BrojZakupaSort koje bi predstavljale sortiranje automobila po cijeni, po broju pređenih kilometara i po dosadašnjem broju rentanja auta od strane svih korisnika.

4.OBSERVE

Uloga Observer paterna je da uspostavi relaciju između objekata tako kada jedan objekat promijeni stanje drugi zainteresirani objekti se obavještavaju. Zainteresirani objekti dobivaju obavještenja, a za slanje obavještenja zadužena je nadležna klasa. Mi smo ovaj patern odlučili implementirati kada administrator promijeni stanje rezervacije u odobrena ili odbijena. Dodat ćemo interfejs IrezervacijObserver, na poziv metode OdobriRezervaciju rezervacija se dodaje u bazu.

5.ITERATOR

Iterator patern omogućava sekvencijalni pristup elementima kolekcije bez poznavanja kako je kolekcija strukturirana. Kolekcija može biti struktuirana u obliku stabla, niza itd. U našem projektu nismo koristili niti jednu kolekciju samim time nije uopšte moguće implementirati ovaj paterni ili navesti način na koji bi smo ga mogli iskoristiti u našem projektu.