

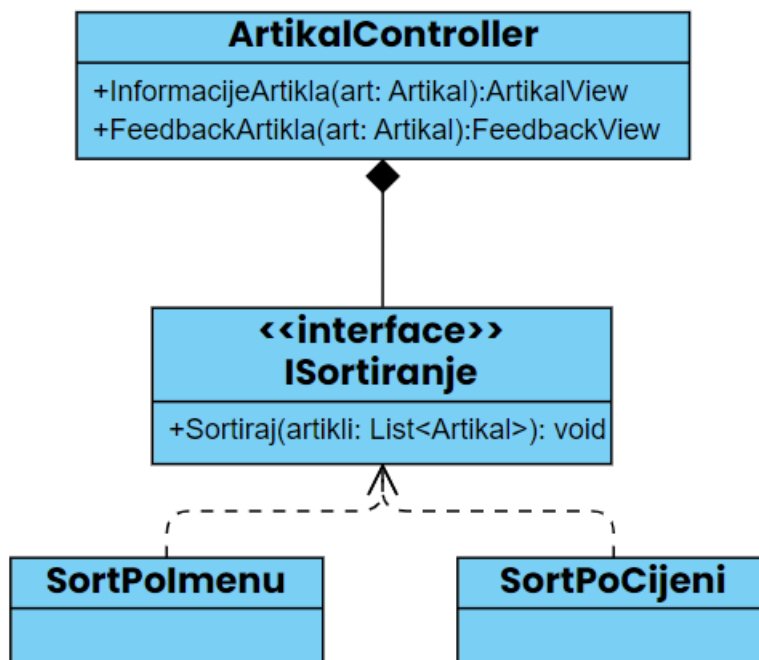
Paterni ponašanja

Strategy patern

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

Primjena:

Ovaj patern cemo iskoristiti za sortiranje artikala To ćemo uraditi tako što ćemo u kontrolerskoj klasi Artikal imati metodu Sortiraj(), u interfejsu ISortiranje je potrebno naslijediti neke vrste sortiranja (npr. Abecedno, OpadajucaCijena, RastucaCijena, ...).

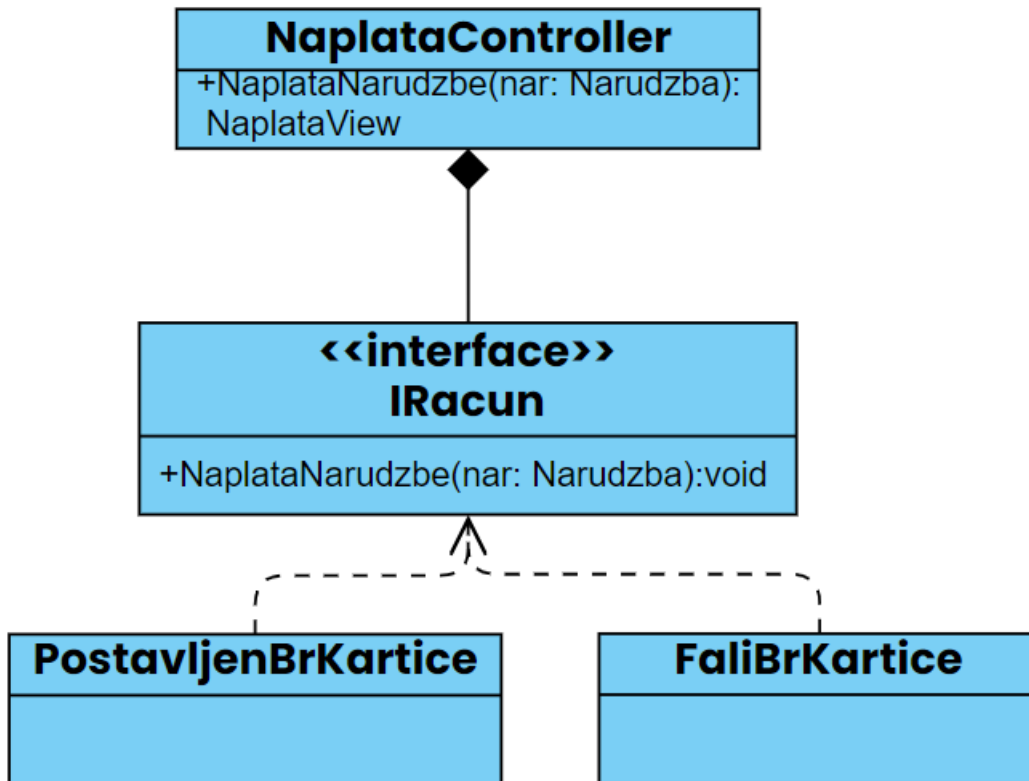


State patern

Objekat mijenja način ponašanja na osnovu trenutnog stanja. Postiže se promjenom podklase unutar hijerarhije klasa.

Primjena:

Ukoliko korisnik nije postavio broj kartice nije mu dozvoljeno izvorsiti kupovinu, nije moguca naplata.



TemplateMethod paterna

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

Primjena:

Trenutno nemamo neki kompleksan algoritam koji bi trebali izdvojiti na male dijelov. Kada bi prosirivali/dodavali nove alogoritme za pregled/sortiranje/filtriranje artikala mogli bi iskoristiti ovaj patern

Observer patern

Uloga Observer paterna je da uspostavi relaciju na način da neki objekti bivaju obaviješteni o pojedinim promjenama tog objekta ili nekih drugih objekata.

Primjena:

Ovaj patern bi mogli koristiti za slanje notifikacija o uspješnoj izvršenoj kupovini, kada bi imali wishlist također bi mogli slati notifikacije da je artikal na pupustu.

Iterator patern

Iterator patern omogućava pristup elementima kolekcije bez poznavanja kako je kolekcija strukturirana.

Primjena:

Sto se ovoga patern tice mogli bi ga iskoristiti za adminov prolazak kroz listu uposlenika. Imali bi dodatne klase za iteriranje koje bi na predstavljale rezlicte nacin prolaska kroz listu korisnika kao sto su npr. ime, broj kupovina, iznos potrosenog novca ...

Chain of responsibility patern

Chain of responsibility omogućava razdvajanje jednog kompleksnog procesa obrade na više objekata koji na različite načine procesiraju primljene podatke.

Primjena:

Ovaj patern bi mogli iskoristiti za izvršenje naplate, prilikom provjere narudžbe mogli bismo imati više handlera kao sto su: handler za potvrdu narudžbe, handler za provjeru uplate iznosa, handler za provjeru broja artikala na stanju ...

Mediator patern

Enkapsulira protokol za komunikaciju među objektima dozvoljavajući da objekti komuniciraju bez međusobnog poznavanje interne strukture objekta

Primjena:

Ovaj patern bi mogli iskoristiti kod davanja feedback-a (ocjena ili komentar) za artikal. Implementirale bi interfejs koji bi provjeravao da li je registrovani korisnik kupio artikal za koji želi da postavi feedback.