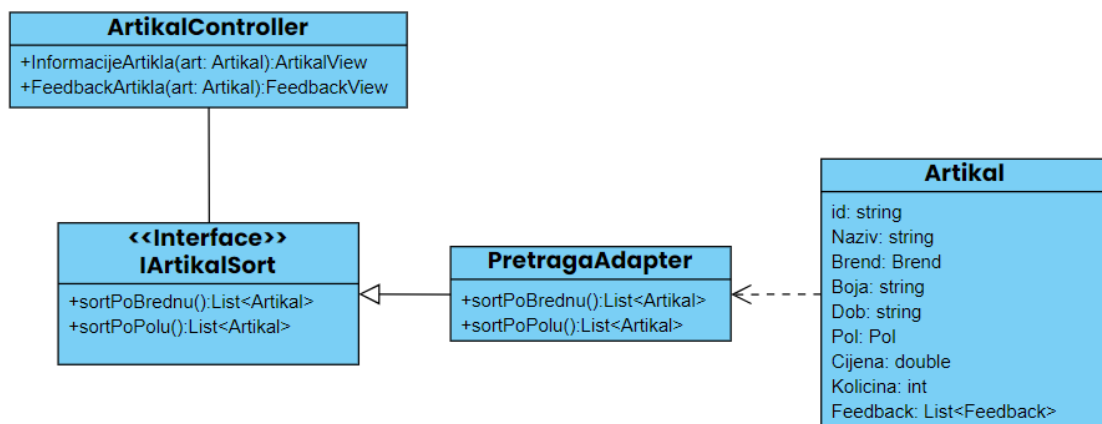


Strukturalni paterni

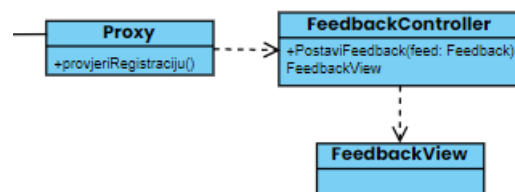
Adapter patern:

Adapter pattern ćemo iskoristiti za sortiranje artikala po različitim kriterijima. Za sad smo planirali dvije vrste sortiranja: po brendu i polu, ali uvijek postoje mogućnosti da se u budućnosti dodaju dodatne vrste sortiranja.



Proxy patern

U našem slučaju proxy pattern smo primjenili pri pokušaju postavljanja feedbacka. Kreirana je klasa proxy koja provjerava da li je korisnik registrovan ili ne, ako jeste onda se dozvoljava postavljanje feedbacka.



Kako bismo mogli iskoristiti ostale strukturalne paterne?

Composite patern

Composite patern omogućava grupisanje objekata u hijerarhijske strukture, gde se pojedinačni objekti i grupe tretiraju na isti način. U našem slučaju kompozitni patern ćemo primjeniti na organizaciju kategorija i podkategorija proizvoda. Svaka kategorija može biti predstavljena kao

objekat koji sadrži podkategorije ili pojedinačne proizvode. Na ovaj način, moći ćemo jednostavno navigirati kroz hijerarhiju kategorija i prikazati odgovarajuće proizvode.

Bridge patern:

Bridge patern odvaja apstrakciju (kako nešto koristimo) od implementacije (kako nešto radi), omogućavajući im da se razvijaju nezavisno jedno od drugog. Mogao bi se iskoristiti prilikom prikaza proizvoda. Na primer, može biti apstraktna klasa "Proizvod" koja sadrži osnovne informacije o proizvodu, poput imena, cijene i slike. A zatim, kreiranje konkretne implementacije mosta koje nasljeđuju apstraktnu klasu "Proizvod" i dodaju specifične informacije o svakom tipu igračke, kao što su "Igračka za bebe" ili "Igračka za predškolarce". Ovaj patern omogućava fleksibilnost u dodavanju novih vrsta igračaka bez potrebe za menjanjem osnovne strukture.

Facade patern

Facade patern pruža jednostavan interfejs koji sakriva kompleksnost unutrašnjih podsistema i olakšava njihovo korišćenje. Može se kreirati facade patern koji pruža jednostavan interfejs za pretragu proizvoda na osnovu različitih kriterijuma, kao što su cijena, marka, starosna grupa i tako dalje.

Flyweight patern

Flyweight patern dijeli objekte kako bi se smanjila potrošnja resursa, dijeleći zajedničke informacije između više objekata umesto da se svaki put iznova stvaraju. Mogli bi kreirati objekte koji čuvaju zajedničke informacije koje se ponavljaju među različitim proizvodima, kao što su ikone, boje ili slike. Umjesto da za svaki proizvod se čuvaju zasebne instance ovih informacija, djelile bi se između više proizvoda. Ovo bi smanjilo memorijski prostor koji je potreban i poboljšalo performanse prikaza stranice.

Decorator patern

Decorator patern omogućava dinamičko dodavanje dodatnih funkcionalnosti ili dekoracija postojećim objektima. U našem slučaju decorator patern ćemo koristiti i za dodavanje opcija za recenzije i ocjene proizvoda. Kreiranjem dekoratera "Dodajte recenziju" i "Prikazujte prosečnu ocenu" ćemo omogućiti korisnicima da ostavljaju povratne informacije o proizvodima i pružiti dodatne informacije drugim kupcima.