



**Univerzitet u Sarajevu**

Elektrotehnički fakultet u Sarajevu Odsjek za  
računarstvo i informatiku



**Objektno Orijentisana Analiza i Dizajn**

**TechHaven**  
Paterni ponašanja

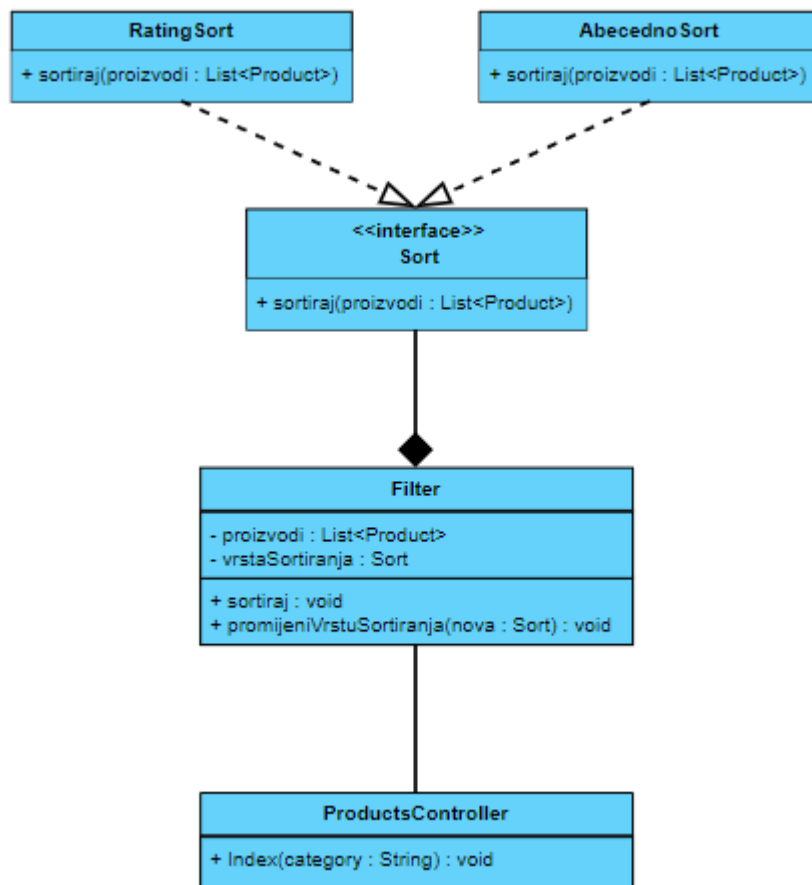
**Članovi tima:**

Amel Ridžal  
Alem Muratović  
Kapić Emir  
Sikirić Muaz  
Alibalić Eman

## Strategy patern

Strategy patern omogućava izdvajanje algoritma iz osnovne klase i smještanje u posebne klase. Ovaj patern omogućava korisniku (User) da izabere jedan od više algoritama iz porodice algoritama.

U kontekstu našeg projekta, Strategy patern je primijenjen kada želimo da pretražujemo proizvode (Product) po zadanim kriterijima. U zavisnosti od načina sortiranja kojeg bira korisnik bira se algoritam po kojem će se sortiranje vršiti.



## Chain of responsibility patern

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

U našem sistemu primjena ovog patern mogla bi biti prilikom checkouta odnosno kupovine proizvoda iz korpe gdje bi se provjeravalo kroz niz koraka validnost podataka i procesuirala bi se kupovina.

## State patern

State patern predstavlja dinamičku verziju Strategy patern. Postiže se promjenom podklase unutar hijerarhije klasa.

U našem sistemu ovo bi se moglo primijeniti kada bi korisnik koji je registrovan mogao postati korisnik sa vip profilom i obrnuto.

## Template patern

Ovaj patern izdvaja određene korake nekog kompleksnog algoritma u podklase. Izdvajaju se manje cjeline (operacije) koje se mogu implementirati na različite načine.

U kontekstu naše online prodavnice, korisnik koji je registrovan i gost imaju većinu istih funkcionalnosti, međutim, registrovan korisnik ima pristup Wishlisti i može kupiti neki proizvod odnosno izvršiti checkout proizvoda iz ShoppingCarta. Ovdje bi se mogao implementirati ovaj patern obzirom da vidimo dosta sličnosti između ovih objekata.

## Observer patern

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

Ovaj patern mogao bi biti primjenjen na način da korisnik dobije obavještenje kada drugi korisnik pokuša da se prijavi na njegov profil.

## Iterator patern

Iterator patern omogućava sekvencijalno kretanje kroz elemente neke kolekcije bez poznavanja kako je kolekcija struktuirana.

U našem sistemu ovaj patern možemo primijeniti tako što se iterira rezultat neke korisničk pretrage

## Mediator patern

Ovaj patern omogućava da objekti komuniciraju bez međusobnog poznavanja interne strukture objekta.

U našem sistemu ovaj patern primijenili smo na način da omogućavamo klijentu komunikaciju sa filterom.

