Strukturalni paterni

Adapter pattern

Adapter pattern bi se mogao iskoristiti u sortiranju proizvoda ako nekad budemo dodavali drugačije načine soritranja. Trenutno je osmišljeno da postoji Tip sortiranja - highest price first, lowest price first itd i metoda Sort prima taj tip sortiranja. Ako bi nekad uveli drugačije sortiranje koje ne prima ni jedan Tip sortiranja nego nešto drugo onda bi konverziju umjesto mijenjanja postojećih metoda mogli uraditi pomoću ovog pattern-a.

Facade pattern

Facade pattern bismo mogli iskoristiti prilikom implementacije validiranja kreditne kartice.Ako iskoristimo neku eksternu biblioteku za ovu svrhu ovaj pattern bi bio koristan da sakrije kompleksnost tog podsistema iza interfejsa.

Decorator pattern

Decorator pattern bismo mogli iskoristiti za dodavanje novih funkcionalnosti u procesu registracije nekad u budućnosti na način da se npr. doda funkcionalnost slanja email-a nakon uspješne registracije ili slično, a što nije trenutno podržano.

Bridge pattern

Imamo sučelje Recommendation koje će imati dvije implementacije: jednu za korisnika (User) i drugu za kupca (Customer), jer se preporuke za njih razlikuju. S druge strane, klasa Analyzer će pružati proizvode na dva načina: ili analizirajući jedan proizvod i vraćajući slične proizvode (kao što je primjenjivo u prikazu detalja proizvoda), ili vraćajući proizvode na temelju prethodnih kupovina korisnika (kao što je na početnoj stranici).

Ovaj pristup omogućava fleksibilnost u pružanju preporuka, jer se različite strategije preporuka mogu primijeniti ovisno o kontekstu korisničkog interakcije.

Proxy pattern

Proxy pattern bismo mogli iskoristiti prilikom dodavanja novih proizvoda. Prodavač umjesto da direktno dodaje proizvode u shop mora to raditi preko posrednika tj. Protection Proxy koji će prvo validirati stanje proizvoda tj. da li sve ima smisla, prije nego što pokuša dodati isti u bazu.

Composite pattern

Composite pattern ima smisla jedino u hijerarhijskim tree-like strukturama.
Ovaj pattern međutim bi potencijalno mogli primjeniti ako bi recimo dodali menadžere kao administratore. U tom slučaju bi dodali List<Administrator> u našu već postojeću Administrator klasu što bi označavalo da je taj administrator menadžer drugim administratorima.

Flyweight pattern

Tipična web-stranica sadrži mnogo slika i dešava se da se ista slika pojavljuje na više mjesta. Bilo bi uzaludno da se svaki put ponovo učitava slika pa ovdje možemo iskoristiti Flyweight pattern za učitavanje svake slike samo jednom. (Imat ćemo jednu sliku za jedan tip podataka npr. Svi isti modeli automobila imaju jednu sliku)