Strukturalni paterni

Potrebno je kreirati dokument u kojem se objašnjava potreba za dodavanjem strukturalnih paterna u postojeći sistem. Potrebno je odabrati i primijeniti najmanje dva strukturalna paterna, a za ostale paterne objasniti na kojim mjestima bi se mogli primijeniti i na koji način.

Proxy patern

U slastičarnici, funkcionalnost kreiranja i pregleda izvještaja treba biti dostupna isključivo administratoru sistema. Kako bismo osigurali kontrolu pristupa, neophodno je implementirati mehanizme autorizacije. U tome nam proxy patern može značajno pomoći.

Proxy će djelovati kao posrednik između korisnika i stvarnog sistema izvještaja. Prije nego što dozvoli pristup funkcijama za pregled ili kreiranje izvještaja, proxy će provjeriti korisničke podatke. Ukoliko korisnik nema potrebne privilegije, koje se odnose na administratora, pristup će biti odbijen, čime se štite osjetljivi podaci i funkcionalnosti.

Dekorater patern

U slastičarni, svaki proizvod treba imati mogućnost jednostavnog mijenjanja različitih svojstava kao što su slika, naziv, opis, ocjena, nutritivna vrijednost, dostupnost na stanju, popust i vrijeme pripreme. Primjenom Dekorater paterna omogućavamo da se osnovni proizvod "ukrašava" ili nadograđuje dodatnim svojstvima iz ovih dviju skupina bez potrebe za stvaranjem velikog broja različitih klasa proizvoda. To znači da možemo fleksibilno kombinirati i mijenjati detalje proizvoda i pripreme u bilo kojem trenutku, što olakšava prilagođavanje proizvoda sezonskim ponudama, promotivnim akcijama ili promjenama u sastavu.

Bridge patern

Pri pripremanju različitih vrsta slatkih proizvoda, kao što su torte, kolači ili muffini, možemo koristiti istu metodu `vrijemePripreme()`. Međutim, kod složenijih proizvoda ili onih sa dodatnim elementima (poput personaliziranih ukrasa, dodataka kao što su

svijeće,), potrebno je povećati vrijeme pripreme proporcionalno tim dodacima.Korištenjem Bridge paterna, odvajamo apstrakciju proizvoda (npr. vrstu slatkiša) od implementacije detalja pripreme (kao što su dodaci ili složenost dekoracije). To nam omogućava da nezavisno proširujemo i mijenjamo vrste proizvoda i njihove dodatke, a metoda `vrijemePripreme()` će uvijek vraćati tačno vrijeme koje uključuje i osnovnu pripremu i sve dodatne zahtjeve.

Kompozitni patern

Pri naručivanju, postoji opcija na određenim proizvodima za dodavanje dodatnih elemenata kao što su čestitke i imena na tortama, svijeće, šlag i slično. Kako bismo elegantno upravljali ovim situacijama, možemo koristiti istu metodu unutar interface-a, npr. `dajNutritivnuVrijednost()`. Za proizvode koji imaju dodatke, ova metoda će izračunati ukupnu nutritivnu vrijednost proizvoda zajedno sa svim dodacima, dok će za proizvode bez dodataka vratiti samo osnovnu nutritivnu vrijednost. Ovim pristupom, kroz klasu `Narudzba` možemo pozvati metodu `dajUkupnuNutritivnuVrijednost()`, koja će vratiti zbir kalorija svih proizvoda u narudžbi, uključujući i one sa dodacima. Kompozitni patern nam omogućava da tretiramo pojedinačne proizvode i njihove složene verzije (proizvodi s dodacima) uniformno, bez potrebe za posebnim razlikovanjem u kodu. Time se postiže fleksibilnost i jednostavnost u proširivanju asortimana proizvoda.

Adapterni patern

Adapterni patern je idealan kada želimo integrisati različite izvore podataka u jedinstveni sistem bez potrebe za direktnom izmjenom postojećih klasa, što nam je i vrlo korisno kada govorimo o slastičarni. Jedna od bitnih funkcionalnosti koju želimo imati u sistemu jeste mogućnost pregleda nutritivne vrijednosti proizvoda. Osnovni proizvodi već imaju implementiran interfejs `dajNutritivnuVrijednost()` koji vraća podatke poput broja kalorija, količine šećera, masti i slično. Međutim, dodatni proizvodi nemaju ovu funkcionalnost jer dolaze iz vanjskog izvora podataka ili su definirani bez odgovarajuće strukture.Tu na scenu stupa adapterni patern. Adapter omogućava da dodatni proizvodi koji nemaju interfejs `dajNutritivnuVrijednost()` budu „prilagođeni“ tako da se mogu koristiti kao da ga imaju. Kreiramo posebnu klasu – adapter – koja implementira interfejs `dajNutritivnuVrijednost()` i unutar sebe sadrži dodatni proizvod. Adapter preuzima informacije o nutritivnoj vrijednosti iz dodatnog proizvoda i prezentuje ih u formatu koji naš sistem očekuje. Na taj način, bez potrebe da mijenjamo postojeće klase dodatnih

proizvoda, možemo ih uključiti u sistem i osigurati da svi artikli (bilo osnovni, bilo dodatni) imaju jedinstven način pristupa nutritivnim informacijama.

Fasadni patern

Ovaj patern možemo iskoristiti tako što klase poput Narudzba, DodajProizvod i Proizvod možemo smjestiti u jedan interface. Samim tim pojednostaviti naš sistem. Na taj način omogućavamo korisnicima, odnosno naručiteljima, da koriste jedinstveni pristup za kreiranje i upravljanje narudžbama bez potrebe da direktno upravljaju sa pojedinačnim klasama ili njihovim detaljima. Fasadni patern skriva kompleksnost pozadinskog procesa pripreme i dodavanja proizvoda, te izlaže samo ključne metode koje su potrebne za efikasno rukovanje narudžbama. Time se postiže pregledniji i jednostavniji kod, a eventualne promjene u poslovnoj logici lako se implementiraju unutar fasadnog sloja, bez utjecaja na ostatak sistema. Ovaj pristup omogućava bržu i sigurniju interakciju naručitelja sa sistemom slastičarne.