

Strukturalni paterni

Sistem **BookMyStyle** obuhvata veliki broj različitih funkcionalnosti i aktera koji učestvuju u rezervaciji termina, slanju obavještenja, upravljanju recenzijama. Strukturalni paterni se koriste za pojednostavljenje odnosa između klasa, smanjenje međuovisnosti i omogućavanje većeg stepena proširivosti.

U nastavku je prikazana primjena **Adapter**, **Bridge**, **Composite**, **Decorator**, **Facade** i **Proxy** paterna u okviru sistema BookMyStyle.

Adapter pattern

Za fleksibilno slanje obavještenja korisnicima putem različitih kanala (email, SMS, push). Uvodi se interfejs `INotificationService`, a `EmailNotificationAdapter` implementira konkretno slanje. Klasa `Obavijest` koristi interfejs bez brige o implementaciji, lako proširivo na `SmsNotificationAdapter` ili `PushNotificationAdapter`.

Prednosti:

- Eliminacija direktne zavisnosti od email sistema
- Podrška za više tipova obavještenja

Bridge pattern

Za različite načine **plaćanja** u sistemu. Iako još nije implementiran, sistem može uvesti module za: plaćanje karticom, u salonu.

Apstraktna klasa `Placanje`

Implementacije: `PlacanjeKarticom`, `PlacanjeUPoslovnici`, `PlacanjePouzecem`

Prednosti:

- Odvajanje apstrakcije od implementacije
- Moguće proširenje bez promjene postojećih klasa

Composite pattern

Modeliranje složenih **paketa usluga** koje se sastoje od više pojedinačnih usluga (npr. šišanje + pranje + feniranje).

- IServiceComponent – interfejs
- SingleService – jednostavna usluga
- ServicePackage – kolekcija usluga

Prednosti:

- Jedinstven tretman jednostavnih i složenih usluga
- Podrška za rekurzivnu kompoziciju

Decorator pattern

Primjenjuje se prilikom odabira vrste šišanja u okviru usluge tipa „Šišanje”.

Različiti dekoratori omogućavaju prilagođavanje usluge šišanja u zavisnosti od spola klijenta i dodatnih zahtjeva.

- Interfejs IVrstaSisanja

Predstavlja osnovni ugovor (Component) za sve varijacije šišanja. Definiše metodu `dajVrstu()` koja vraća opis vrste šišanja.

- SisanjeComponent

Konkretna implementacija interfejsa IVrstaSisanja koja sadrži osnovne vrste šišanja (musko, zensko). Koristi se kada je potrebno prikazati standardne opcije bez dodatnih proširenja. Ova klasa predstavlja **osnovnu uslugu šišanja**.

- SisanjeDecorator

Apstraktni dekorator (Decorator) koji implementira IVrstaSisanja i sadrži referencu na drugi objekat istog interfejsa. Omogućava nadogradnju osnovne usluge bez mijenjanja postojeće klase.

- ConcreteDecoratorM

Dodaje specifičnosti šišanja za **muške klijente**. Može uključivati dodatne korake (npr. brijanje vrata, geliranje).

- ConcreteDecoratorZ

Dodaje prilagođene korake šišanja za **ženske klijente**, poput ravnjanja vrhova, feniranja ili oblikovanja šiški.

Prednosti:

- Fleksibilna kontrola pristupa nad dijelovima korisničkog profila
- Bez potrebe za višestrukim nasljeđivanjem

Facade pattern

Kod **validacije podataka korisnika** (email adrese, lozinke, datuma) tokom registracije, zakazivanja ili uređivanja termina.

- ValidacijaFacade skriva složenu logiku provjere (regex, duplikati, pozivanje API-ja)

Prednosti:

- Smanjenje kompleksnosti kod interakcije sa podsistemima
- Jednostavno korištenje iz aplikacije

Proxy pattern

Zaštita pristupa administrativnim funkcijama – samo AdminProxy može proslijediti poziv pravim klasama nakon autentifikacije.

- Interfejs IAdminAkcija
- RealAdminAkcija – sadrži logiku
- AdminProxy – kontrola pristupa

Prednosti:

- Ograničavanje pristupa osjetljivim metodama
- Enkapsulacija kontrole pristupa na jednom mjestu