

Strukturalni Paterni

Za naš projekat koristit ćemo Facade i Adapter pattern-e.

Facade Pattern

Opis:

Facade patern pojednostavljuje rad korisniku skrivajući kompleksnu logiku sistema iza jednog interfejsa. Naš sistem omogućava korisniku da rezerviše termin, kreira plan ishrane i obavi kupovinu. Bez fasade, korisnik bi morao komunicirati s više klasa direktno.

Primjena u sistemu:

Koristi se za objedinjavanje složenih operacija rezervacije treninga i planiranja ishrane u jednu tačku pristupa.

Klase:

- FitnessFacade – Centralizuje kompleksne operacije:
 - zakaziTrening(...)
 - generisiPlanIshrane(...)
 - obaviKupovinu(...)

Adapter Pattern

Opis:

Adapter omogućava komunikaciju između nekompatibilnih interfejsa. Kada imamo eksterne trenere iz drugih sistema koji ne koriste našu klasu Korisnik, adapter omogućava njihovu integraciju.

Primjena u sistemu:

Povezivanje eksternih trenera (npr. EksterniTrener objekti) sa sistemom koji koristi Korisnik kao trenera za termine i treninge.

Klase:

- EksterniTrener – Klasa izvan naše domene (npr. vanjski API)
- TrenerAdapter – Implementira potrebni interfejs i prevodi pozive na EksterniTrener

Preostali paterni koje nećemo koristiti:

Decorator Patern

Decorator (Dekorater) je dizajnerski obrazac koji omogućava dinamičko dodavanje novih funkcionalnosti objektima bez mijenjanja njihovog koda.

Koristi se kada želimo proširiti ponašanje objekta bez korištenja nasljeđivanja. Omogućava kreiranje više "dekoratora" koji oblažu osnovni objekt i dodaju mu nove mogućnosti.

Bridge Patern

Bridge (Most) je obrazac koji odvaja apstrakciju od njene implementacije tako da obje mogu nezavisno evoluirati.

Služi za izbjegavanje krute hijerarhije klasa (previše nasljeđivanja) i omogućava fleksibilnost tako što se logika raspoređuje između dvije nezavisne hijerarhije – apstrakcije i implementacije.

Composite Patern

Composite (Kompozitni) je obrazac koji omogućava da se pojedinačni objekti i njihove grupe tretiraju na isti način.

Koristi se kada imamo hijerarhijske strukture poput stabala, gdje lista objekata i pojedinačni objekti trebaju imati identičan interfejs.

Proxy Patern

Proxy (Zastupnik) je obrazac koji pruža zamjenski objekt koji kontrolira pristup stvarnom objektu.

Koristi se za kontrolu pristupa, optimizaciju performansi (npr. keširanje), sigurnost ili odgođeno učitavanje. Proxy objekt djeluje kao posrednik prema stvarnom objektu.