

Patterni ponašanja

Patterni ponašanja koje smo odlučile implementirati u našem projektu su:

Strategy pattern - design pattern koji se koristi kada želimo omogućiti promjenu ponašanja objekta tokom izvođenja programa. Ovaj pattern omogućuje definiranje različitih strategija (algoritama) koje objekt može koristiti, a odabir strategije se može mijenjati dinamički. U našem projektu, Strategy pattern bi mogao biti primijenjen kod plaćanja narudžbe. U različitim dijelovima našeg koda gdje se vrši kreiranje i manipulacija narudžbama, možemo koristiti Strategy pattern tako da omogućimo odabir različitih strategija plaćanja. Na primjer, prilikom kreiranja narudžbe, klijent može odabrati plaćanje putem PayPal-a, Credit Card-a ili gotovinom. Kreiranjem interfacea `PaymentStrategy` definiramo metode poput `makePayment` i `refund` za različite načine plaćanja. Na primjer, možemo imati implementacije `PayPalStrategy`, `CreditCardStrategy`, `CashStrategy` itd. `Order` klasa bi sadržavala metode poput `setPaymentStrategy` koja postavlja željenu strategiju plaćanja i `makePayment` koja će pozvati metodu `makePayment` na odabranoj strategiji. Implementacija Strategy patterna omogućava fleksibilnost u odabiru strategije plaćanja, omogućujući jednostavno dodavanje novih načina plaćanja u budućnosti. Također, odvaja logiku plaćanja od same klase `Order`, što olakšava održavanje i poboljšava čitljivost koda.