

Wzorzec projektowy: Fasada (Facade)

Wojciech Pacocha 272534

1. Problem

W złożonych systemach oprogramowania komponenty często są ze sobą silnie powiązane, co skutkuje trudnościami w ich używaniu oraz konserwacji. Programista korzystający z takiego systemu musi znać wiele klas oraz zależności między nimi, co powoduje nadmierne skomplikowanie interfejsu i obniża czytelność kodu. Problem pojawia się, gdy:

- Zewnętrzne moduły muszą komunikować się z wieloma wewnętrznymi komponentami.
- Chcemy uprościć interfejs do złożonego systemu.
- Chcemy ukryć złożoność implementacji przed użytkownikiem końcowym (np. innym deweloperem).

2. Rozwiązanie

Wzorzec **Fasada** polega na utworzeniu jednej klasy (lub zestawu klas), która stanowi uproszczony interfejs do bardziej złożonego podsystemu. Klasa ta ukrywa szczegóły implementacyjne oraz zależności pomiędzy poszczególnymi komponentami systemu, delegując wywołania do odpowiednich obiektów wewnętrznych.

Główne elementy wzorca:

- Fasada – główny punkt dostępu, który udostępnia uproszczony interfejs.
- Komponenty podrzędne – klasy, które wykonują rzeczywistą pracę, ale są ukryte za fasadą.

Zalecenia implementacyjne:

- Fasada nie powinna całkowicie ukrywać komponentów podrzędnych – powinna umożliwiać dostęp do nich, jeśli zaistnieje taka potrzeba.
- Może zawierać wiele metod wysokiego poziomu, które wykonują określone sekwencje operacji z wykorzystaniem komponentów podrzędnych.

3. Konsekwencje

Zalety:

- Upraszcza życie złożonego systemu, prezentując spójny i przejrzysty interfejs.
- Redukuje zależności pomiędzy klientem a komponentami wewnętrznymi.
- Ułatwia testowanie i rozwój – klient komunikuje się tylko z fasadą.
- Pomaga w organizacji kodu i zwiększa jego czytelność.

Wady:

- Nadmierna ilość logiki w klasie fasady może prowadzić do jej przeciążenia.
- Może ukryć ważne możliwości i szczegóły komponentów podrzędnych.
- Jeśli projekt fasady nie będzie odpowiednio przemyślany, może stać się trudna do utrzymania.