Patrón de Diseño
Facade

Blanco Julio Cesar
DNI 39193627
01/04/20

Problema
Se tiene un conjunto amplio de objetos que pertenecen a librerías o clases complejas. Normalmente se tendría que inicializar los objetos y ejecutar los métodos en el orden correcto. Esto genera alto acoplamiento entre las implementaciones.

Contexto

Facade se utiliza cuando se necesita un interfaz limitada, simple y sencilla para un subsistema complejo. También cuando se necesita estructurar al subsistema en capas.

Solución

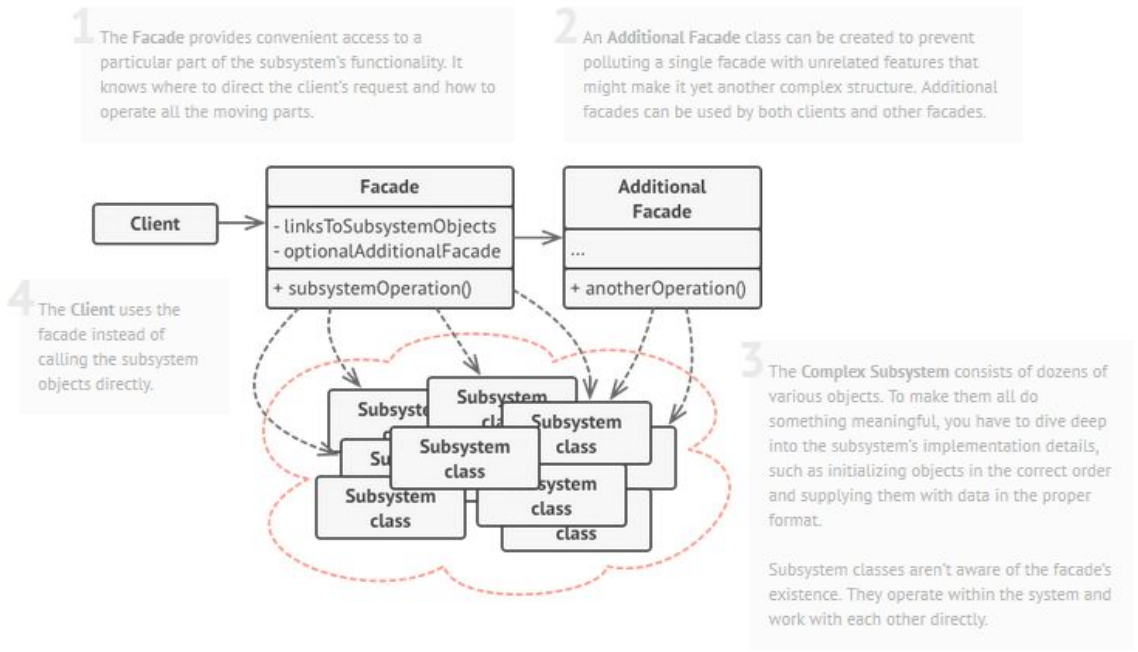
Se crea una facade, que es un clase que proporciona una interfaz simple contra uno y varios subsistemas complejos. Esta facade puede limitar la funcionalidad del subsistema, aunque contiene sólo las características que importa al cliente.
Resulta útil a la hora de integrar la aplicación con un framework o librería compleja, pero no se utiliza en su totalidad el mismo.

Objetivo

El patrón Facade es un patrón de diseño estructural que provee una interfaz simplificada para una librería, framework o un conjunto complejo de clases.

Implementación

a) Comprobar si es posible proveer una interfaz más simple que alguno de los subsistemas.
b) Implementar la clase Facade, esta clase debería enviar mensajes a los objetos apropiados del subsistema.
c) El código cliente deberá acceder al subsistema únicamente a traves de la clase Facade.



Patrones Relacionados

Singleton: el patrón Singleton puede ser aplicado a Facade, ya que solamente es necesario una instancia de la clase Facade.

Adapter: El patrón Adapter se aplica a solamente un objeto, mientras que Facade se aplica a un subsistema entero.

Proxy: Ambos patrones funcionan como buffer para el subsistema, la diferencia radica en que Proxy utiliza la misma interfaz que el objeto.