Modelagem de Sistemas

Prof. Dr. Antonio Marcos Selmini

SISTEMAS DE INFORMAÇÃO **ESPM**



Padrão de Projeto Estrutural Facade

Padrão Estrutural Facade

Segundo GoF:

Fornecer uma interface unificada para um conjunto de interfaces em um subsistema. **Façade** define uma interface de nível mais alto que torna o subsistema mais fácil de ser usado.

Estruturar um sistema em subsistemas ajuda a reduzir a complexidade. Um objetivo comum de todos os projetos é minimizar a comunicação e as dependências entre subsistemas. Uma maneira de atingir esse objetivo é introduzir um objeto **façade** (fachada), o qual fornece uma interface única e simplificada para os recursos e facilidades mais gerais de um subsistema.

Padrão Estrutural Facade – aplicabilidade

Complexidade do sistema: quando um sistema ou subsistema se torna muito complexo, com muitas classes e interações entre elas, o Facade pode ser utilizado para reduzir a complexidade visível para o cliente. Ele facilita a interação com um conjunto de classes de forma organizada e coesa.

Redução de acoplamento: se um cliente está altamente acoplado a muitos detalhes internos do sistema, o Facade pode ser implementado para desacoplar o cliente das classes de implementação. Isso possibilita maior flexibilidade e manutenção, permitindo mudanças internas sem quebrar a interface usada pelo cliente.

Múltiplos subsistemas: Quando um sistema é composto por diversos subsistemas que são interdependentes e devem ser coordenados para realizar uma tarefa, o Facade pode ser útil para coordenar a interação entre eles e simplificar as chamadas para o cliente.

Padrão Estrutural Facade – aplicabilidade

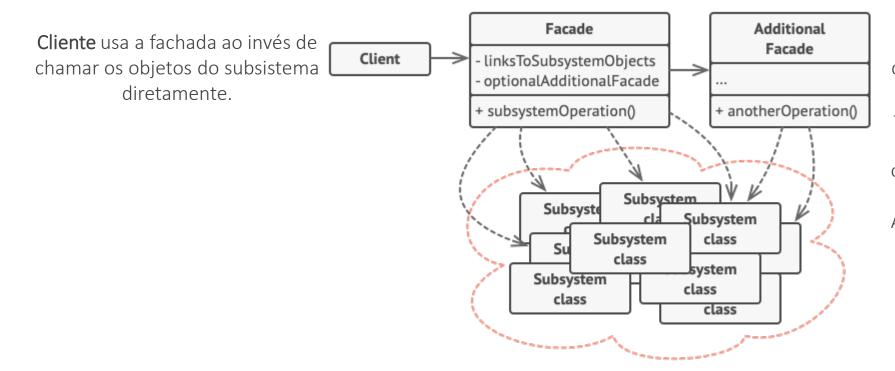
Múltiplos subsistemas: quando um sistema é composto por diversos subsistemas que são interdependentes e devem ser coordenados para realizar uma tarefa, o **Facade** pode ser útil para coordenar a interação entre eles e simplificar as chamadas para o cliente.

Interação complexa com uma biblioteca: em sistemas que integram bibliotecas de terceiros, as classes e métodos oferecidos por essas bibliotecas podem ser complexos e difíceis de usar diretamente. Nesse caso, o Facade pode encapsular essa complexidade e fornecer uma interface mais amigável e coesa.

Manutenção e evolução: quando você deseja isolar o código cliente das mudanças internas do sistema. O uso de um Facade permite que o sistema subjacente evolua sem impactar diretamente os clientes, pois a interface do Facade permanece a mesma enquanto a implementação interna pode mudar.

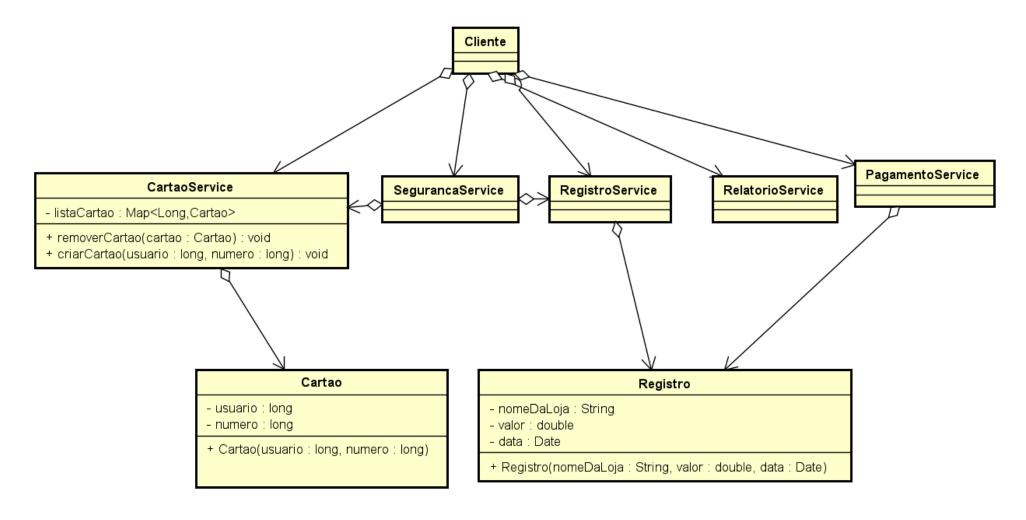
Padrão Estrutural Facade – componentes

Fachada fornece um acesso conveniente para uma parte particular da funcionalidade do subsistema. Ela sabe onde direcionar o pedido do cliente e como operar todas as partes móveis.



Subsistema Complexo consiste em dúzias de objetos variados. Para tornar todos eles em algo que signifique alguma coisa, você tem que mergulhar fundo nos detalhes de implementação do subsistema, tais como objetos de inicialização na ordem correta e suprí-los com dados no formato correto. As classes do subsistema não estão cientes da existência da fachada. Elas operam dentro do sistema e trabalham entre si diretamente.

Padrão Estrutural Facade – exemplo



Padrão Estrutural Facade – exercício

Você deverá desenvolver um sistema para simular um sistema de gerenciamento de pedidos online que permita aos clientes realizarem compras e ao sistema processar pagamentos e gerar notas fiscais. O sistema possui várias operações internas, como validação de estoque, processamento de pagamento e geração de nota fiscal, que são complexas. Para simplificar a interação do cliente com o sistema, você irá aplicar o padrão **Facade**.

Requisitos: o sistema deve ter as seguintes operações principais:

- Realizar pedido: verificar o estoque, processar o pagamento e gerar a nota fiscal para o cliente.
- Cancelar pedido: verificar se o pedido pode ser cancelado, realizar o estorno do pagamento e atualizar o estoque.

Padrão Estrutural Facade – exercício

Classes de subsistema:

- **Estoque**: classe responsável por verificar e atualizar a quantidade de produtos no estoque.
- Pagamento: classe responsável por processar o pagamento (apenas simular a operação).
- NotaFiscal: classe responsável por gerar uma nota fiscal (simulada com a exibição de um número fictício).
- Pedido: classe para manter o estado do pedido (produtos comprados, status de pagamento e nota fiscal).
- Classe Facade: crie a classe PedidoFacade, que deve simplificar a interação com o sistema. O cliente deve poder realizar pedidos e cancelá-los através dessa classe.
- O cliente (classe Main) interage apenas com a classe PedidoFacade, sem necessidade de conhecer os detalhes das operações internas.

SISTEMAS DE INFORMAÇÃO **ESPM**