INSTITUTO FEDERAL
Santa Catarina

Técnico em Desenvolvimento de Sistemas

Projeto de Software I

Juliano Lucas Gonçalves

juliano.goncalves@ifsc.edu.br / julianolg@gmail.com

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Agenda

Padrões de Projetos

- Estruturais
- Comportamentais

INSTITUTO FEDERAL
Santa Catarina

Padrões estruturais

PADRÕES ESTRUTURAIS – 7 padrões

– De que maneira as classes e objetos são compostos para a formação de grandes estruturas (Erich Gamma, et al.).

INSTITUTO FEDERAL
Santa Catarina

Padrões Estruturais (7 padrões)

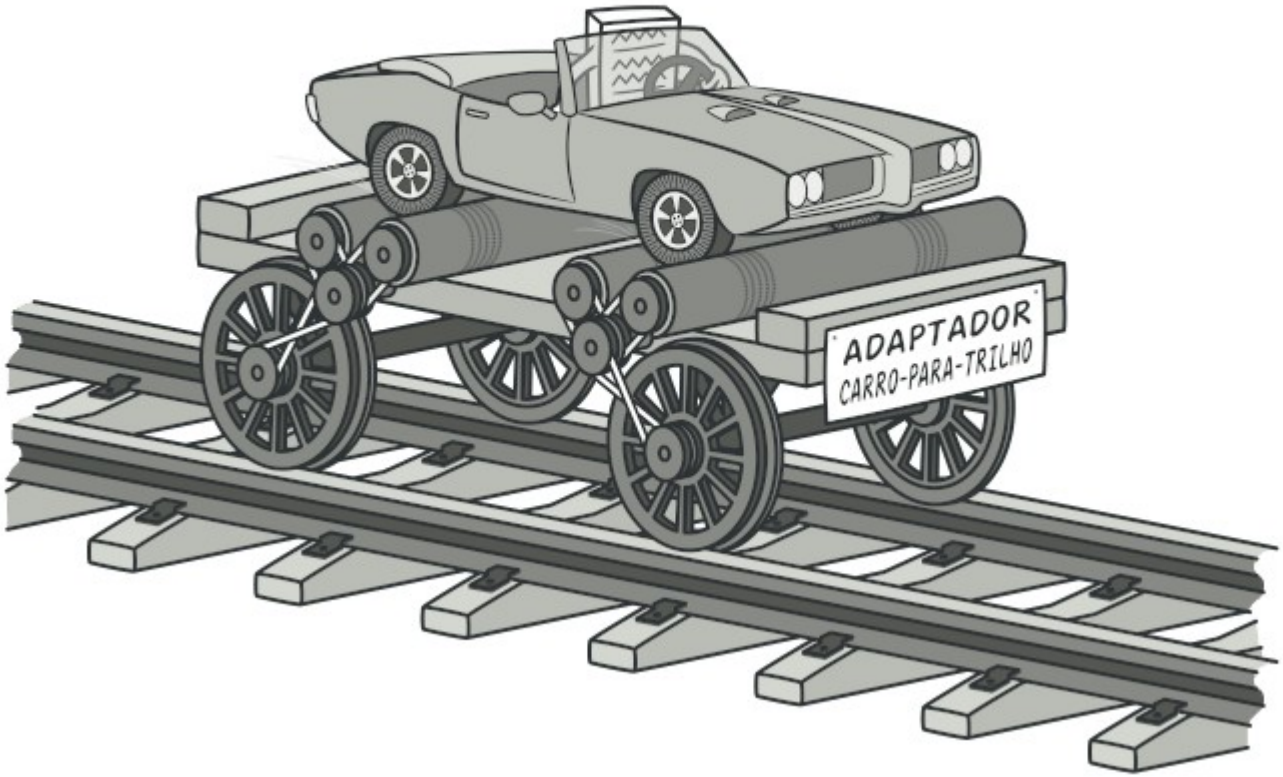
Quais são eles:

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

INSTITUTO FEDERAL
Santa Catarina

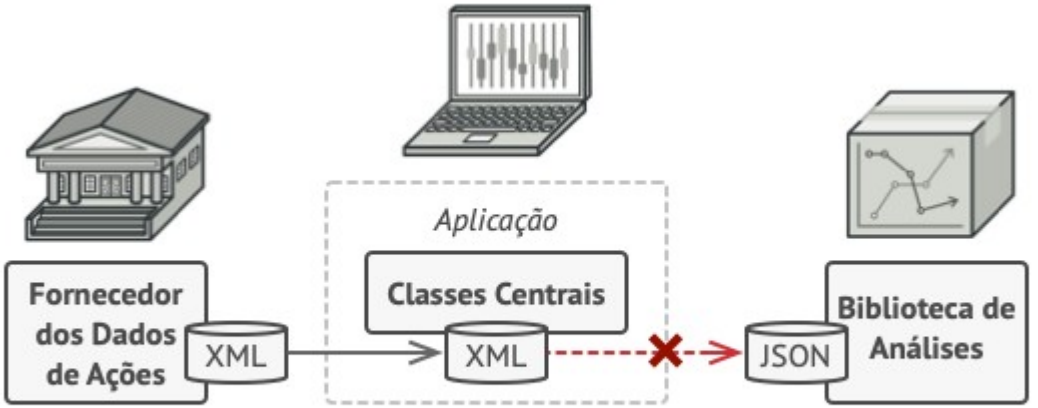
Adapter

O **Adapter** é um padrão de projeto estrutural que permite objetos com interfaces incompatíveis colaborarem entre si.



INSTITUTO FEDERAL
Santa Catarina

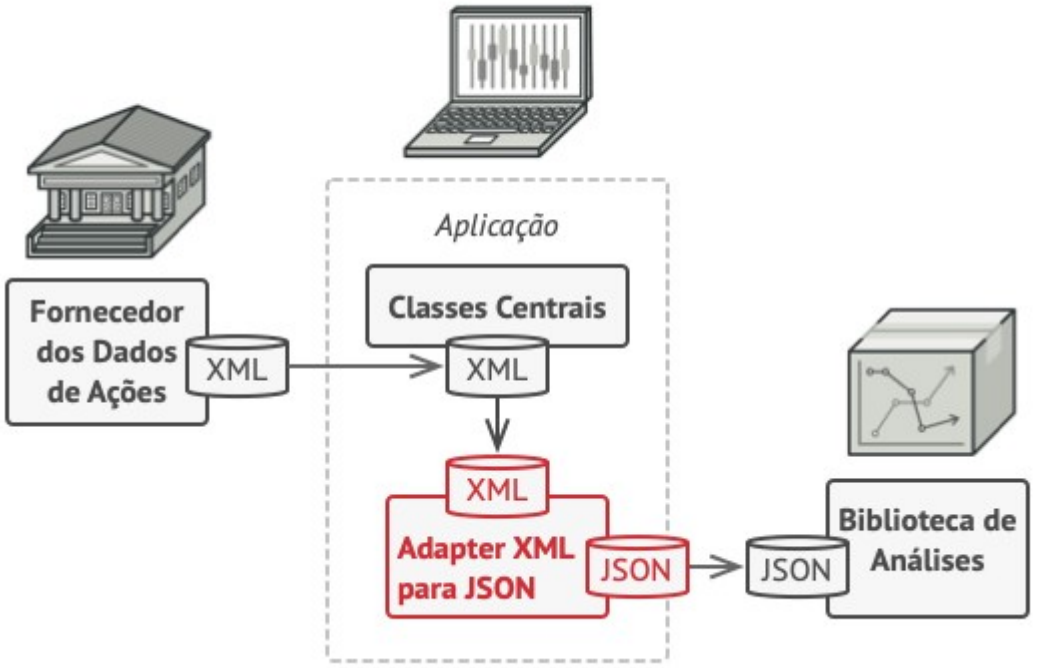
Adapter (Problema)



Você não pode usar a biblioteca "como ela está" porque ela espera os dados em um formato que é incompatível com sua aplicação.

INSTITUTO FEDERAL
Santa Catarina

Adapter (Solução)



INSTITUTO FEDERAL
Santa Catarina

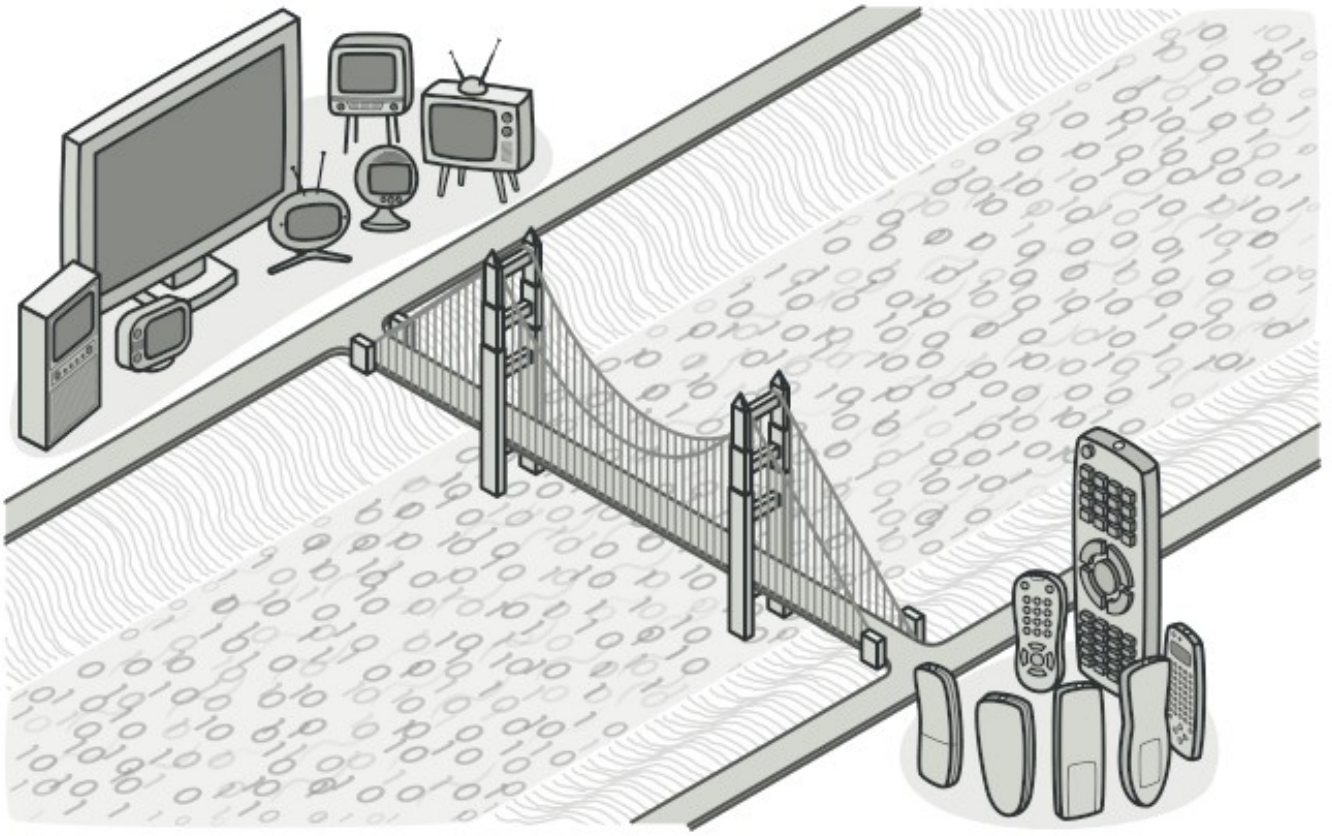
Adapter (Analogia com o mundo real)

Uma mala antes e depois de uma viagem ao exterior.

INSTITUTO FEDERAL
Santa Catarina

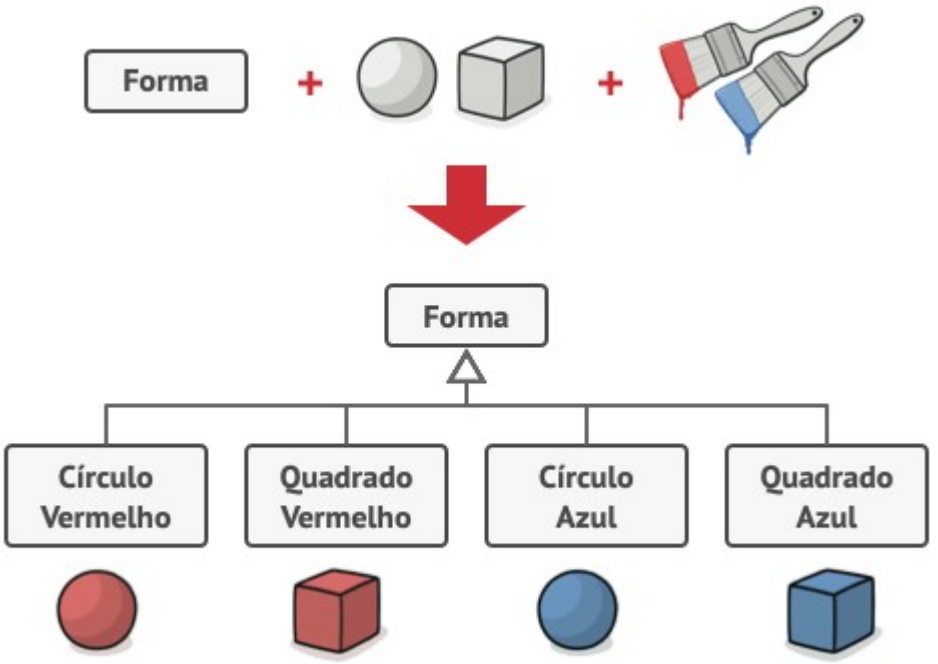
Bridge

O **Bridge** é um padrão de projeto estrutural que permite que você divida uma classe grande ou um conjunto de classes intimamente ligadas em duas hierarquias separadas—abstração e implementação—que podem ser desenvolvidas independentemente umas das outras.



INSTITUTO FEDERAL
Santa Catarina

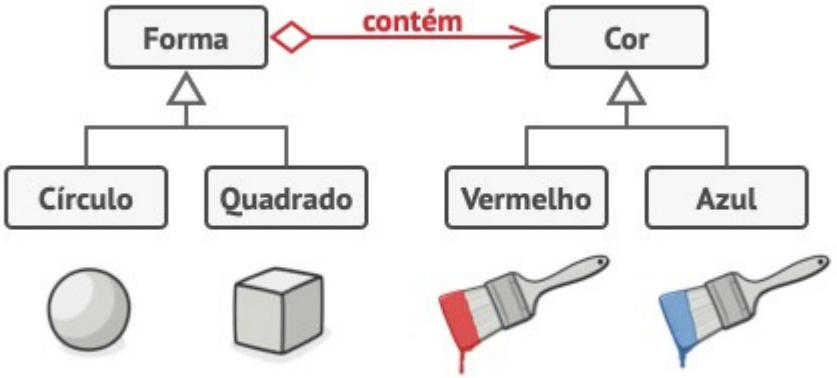
Bridge (Problema)



O número de combinações de classes cresce em progressão geométrica.

INSTITUTO FEDERAL
Santa Catarina

Bridge (Solução)

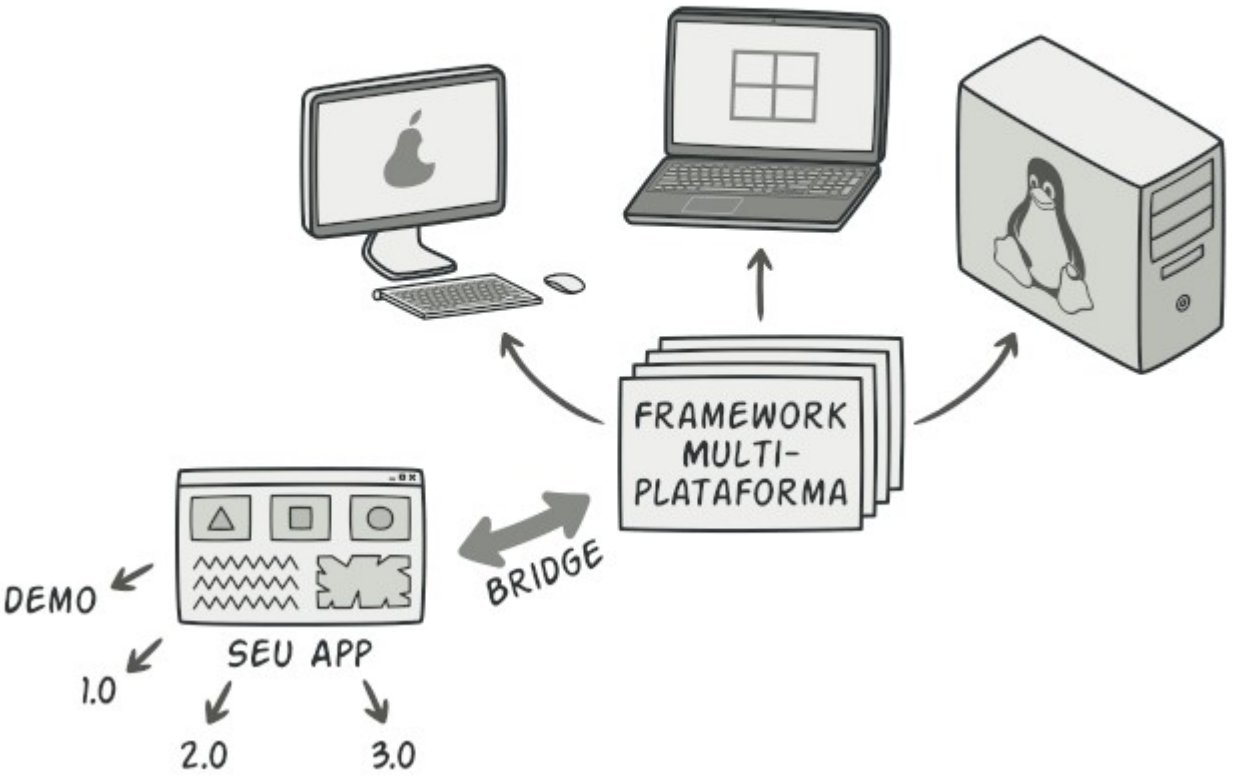


Você pode prevenir a explosão de uma hierarquia de classe ao transformá-la em diversas hierarquias relacionadas.

INSTITUTO FEDERAL
Santa Catarina

Bridge (Analogia com o mundo real)

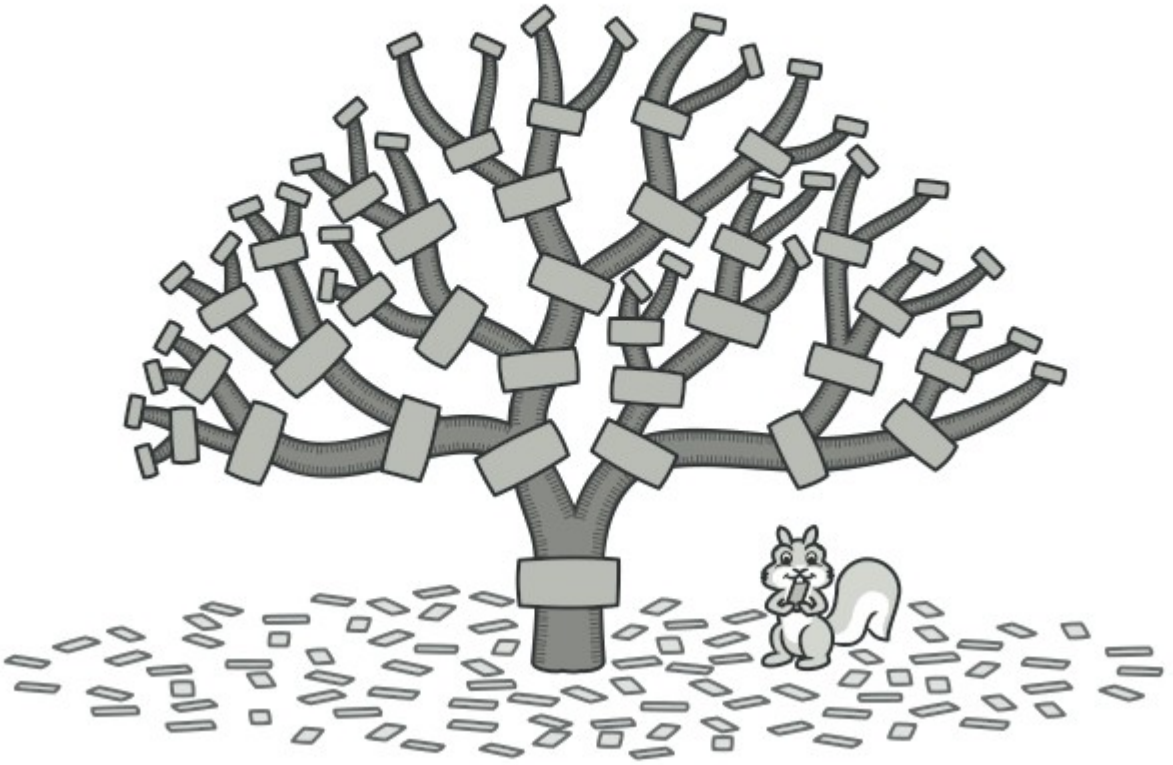
- Abstração: a camada GUI da aplicação.
- Implementação: As APIs do sistema operacional.



INSTITUTO FEDERAL
Santa Catarina

Composite

O **Composite** é um padrão de projeto estrutural que permite que você componha objetos em estruturas de árvores e então trabalhe com essas estruturas como se elas fossem objetos individuais.



INSTITUTO FEDERAL
Santa Catarina

Composite (Problema)

Um pedido pode envolver vários produtos, embalados em caixas, que são embalados em caixas maiores e assim em diante. Toda a estrutura se parece com uma árvore de cabeça para baixo.

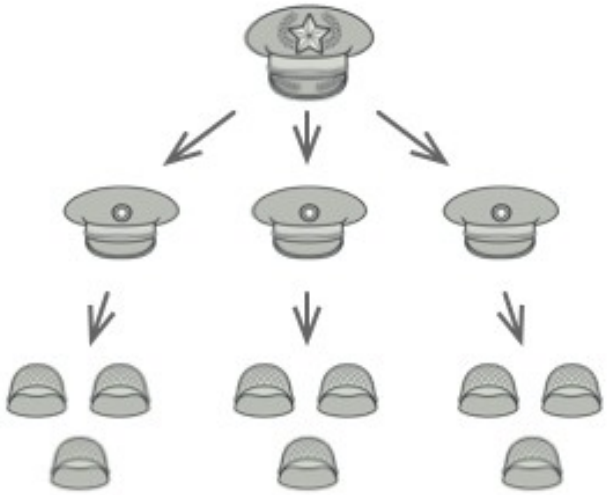
INSTITUTO FEDERAL
Santa Catarina

Composite (Solução)

Uma caixa verifica cada produto, pergunta o preço e retorna o valor ao final

INSTITUTO FEDERAL
Santa Catarina

Composite (Analogia com o mundo real)

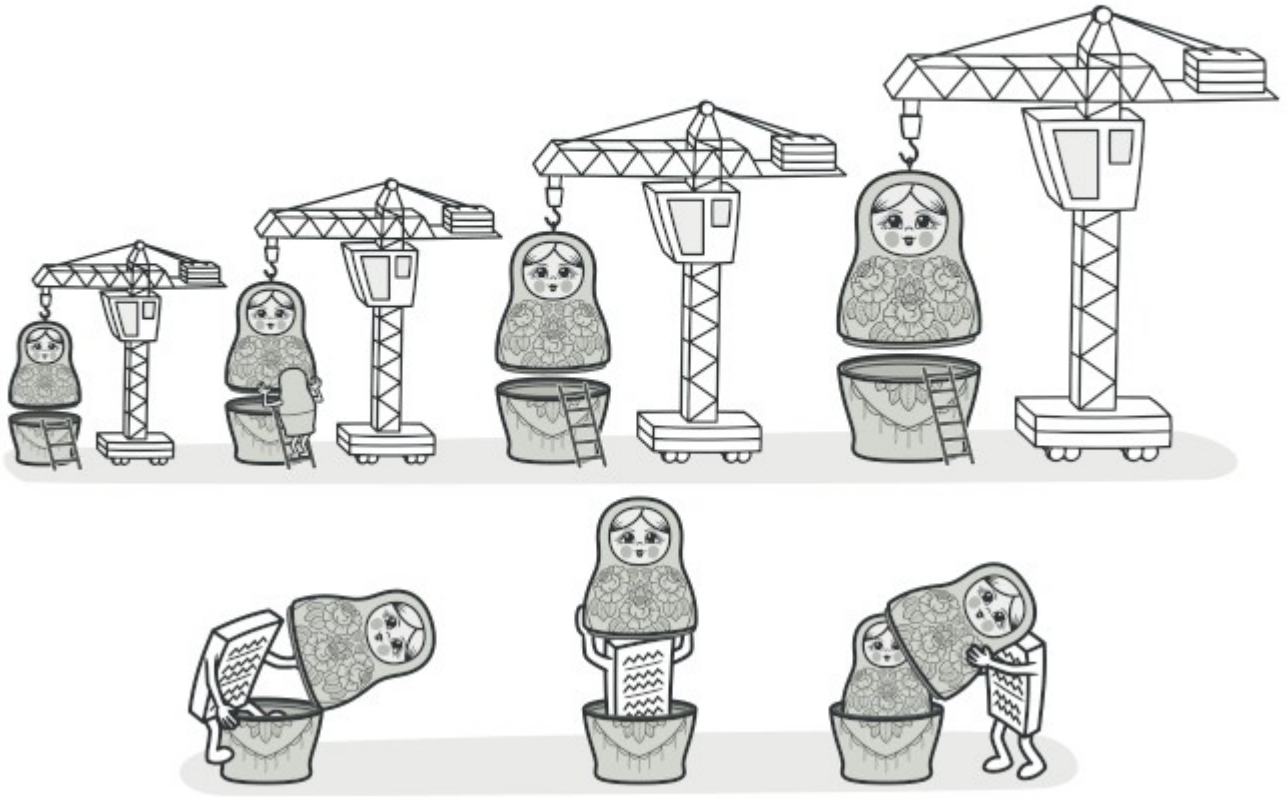


Um exemplo de uma estrutura militar.

INSTITUTO FEDERAL
Santa Catarina

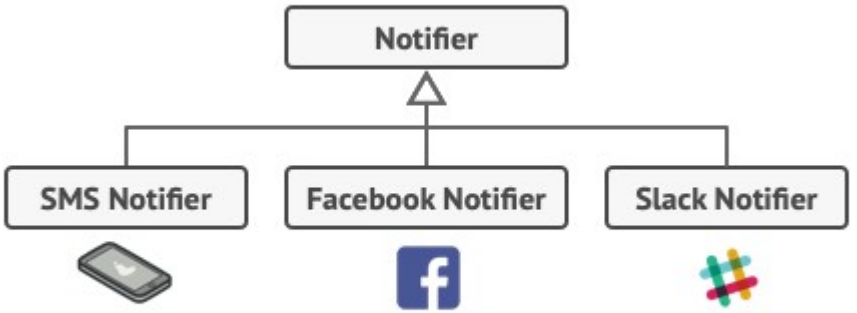
Decorator

O **Decorator** é um padrão de projeto estrutural que permite que você acople novos comportamentos para objetos ao colocá-los dentro de invólucros de objetos que contém os comportamentos.



INSTITUTO FEDERAL
Santa Catarina

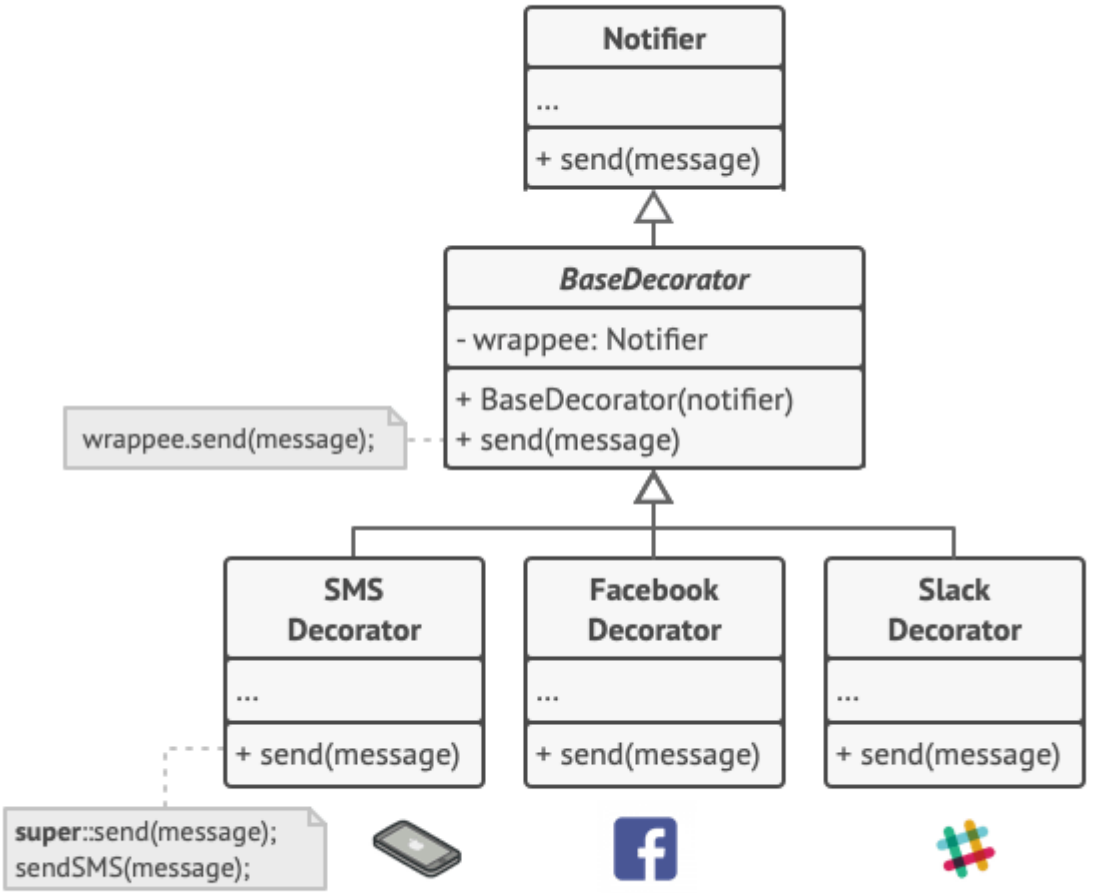
Decorator (Problema)



Cada tipo de notificação é implementada em uma subclasse do notificador.

INSTITUTO FEDERAL
Santa Catarina

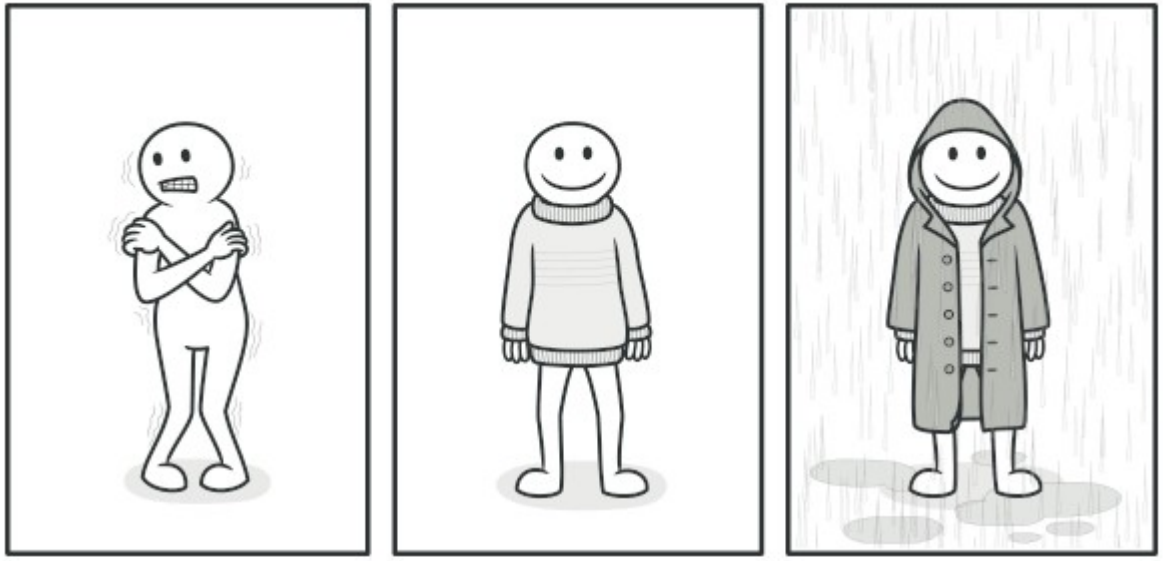
Decorator (Solução)



Vários métodos de notificação se tornam decoradores.

INSTITUTO FEDERAL
Santa Catarina

Decorator (Analogia com o mundo real)



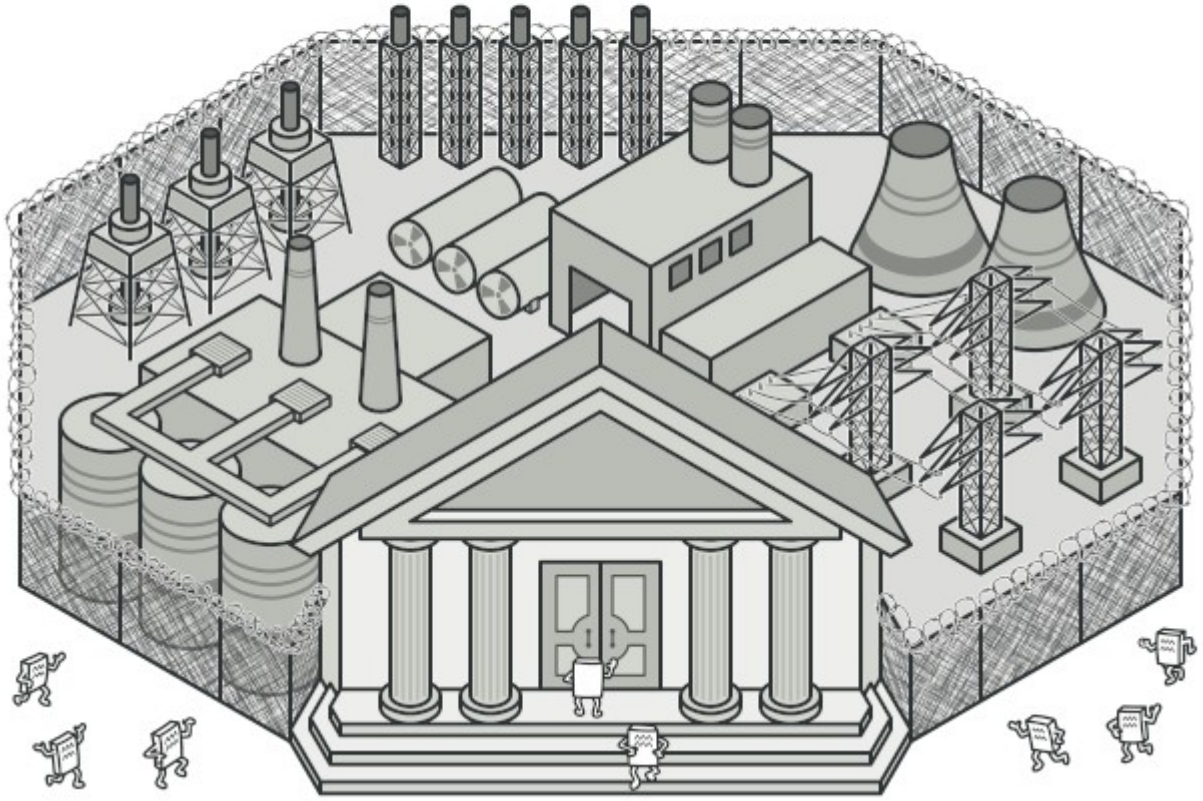
Você tem um efeito combinado de usar múltiplas peças de roupa.

Vestir roupas é um exemplo de usar decoradores. Quando você está com frio, você se envolve com um suéter. Se você ainda sente frio com um suéter, você pode vestir um casaco por cima. Se está chovendo, você pode colocar uma capa de chuva. Todas essas vestimentas "estendem" seu comportamento básico mas não são parte de você, e você pode facilmente remover uma peça de roupa sempre que não precisar mais dela.

INSTITUTO FEDERAL
Santa Catarina

Facade

O **Facade** é um padrão de projeto estrutural que fornece uma interface simplificada para uma biblioteca, um framework, ou qualquer conjunto complexo de classes.



INSTITUTO FEDERAL
Santa Catarina

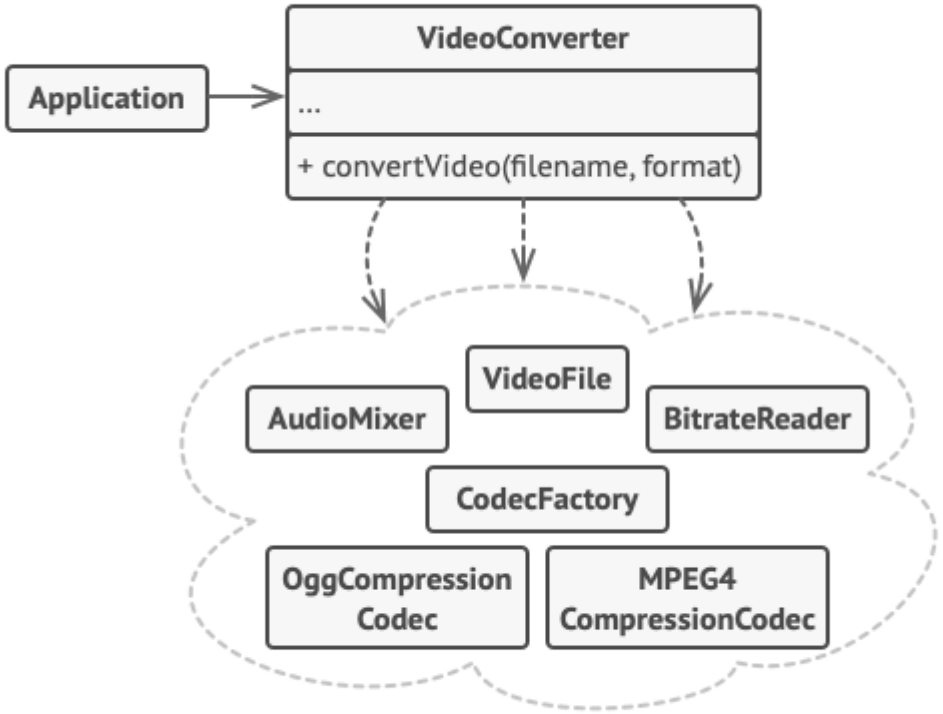
Facade (Problema)

Imagine que você precisa fazer seu código funcionar com um amplo conjunto de objetos que pertencem a uma sofisticada biblioteca ou framework. Normalmente, você precisaria inicializar todos aqueles objetos, rastrear as dependências, executar métodos na ordem correta, e assim por diante.

Como resultado, a lógica de negócio de suas classes vai ficar firmemente acoplada aos detalhes de implementação das classes de terceiros, tornando difícil compreendê-lo e mantê-lo.

INSTITUTO FEDERAL
Santa Catarina

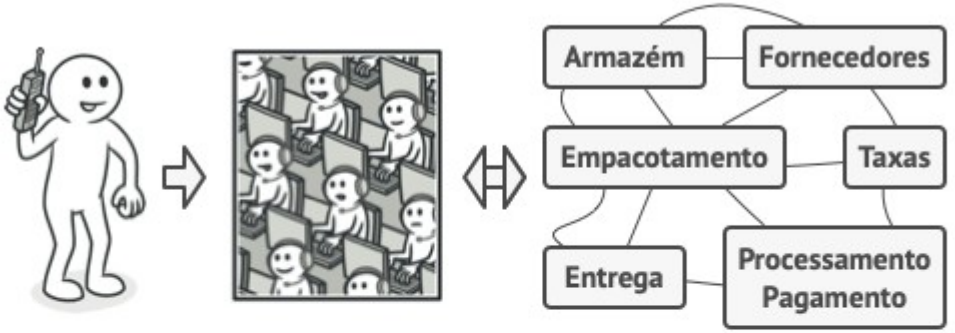
Facade (Solução)



Um exemplo de isolamento de múltiplas dependências dentro de uma única classe fachada.

INSTITUTO FEDERAL
Santa Catarina

Facade (Analogia com o mundo real)



Fazer pedidos por telefone.

Quando você liga para uma loja para fazer um pedido, um operador é sua fachada para todos os serviços e departamentos da loja. O operador fornece a você uma simples interface de voz para o sistema de pedido, pagamentos, e vários sistemas de entrega.

INSTITUTO FEDERAL
Santa Catarina

Padrões comportamentais

PADRÕES COMPORTAMENTAIS – 11 padrões

– Preocupam-se com algoritmos e atribuição de responsabilidades entre objetos. Descrevem, também, um padrão de comunicação entre classes ou objetos (Erich Gamma, et al.).

INSTITUTO FEDERAL
Santa Catarina

Padrões Comportamentais (11 padrões)

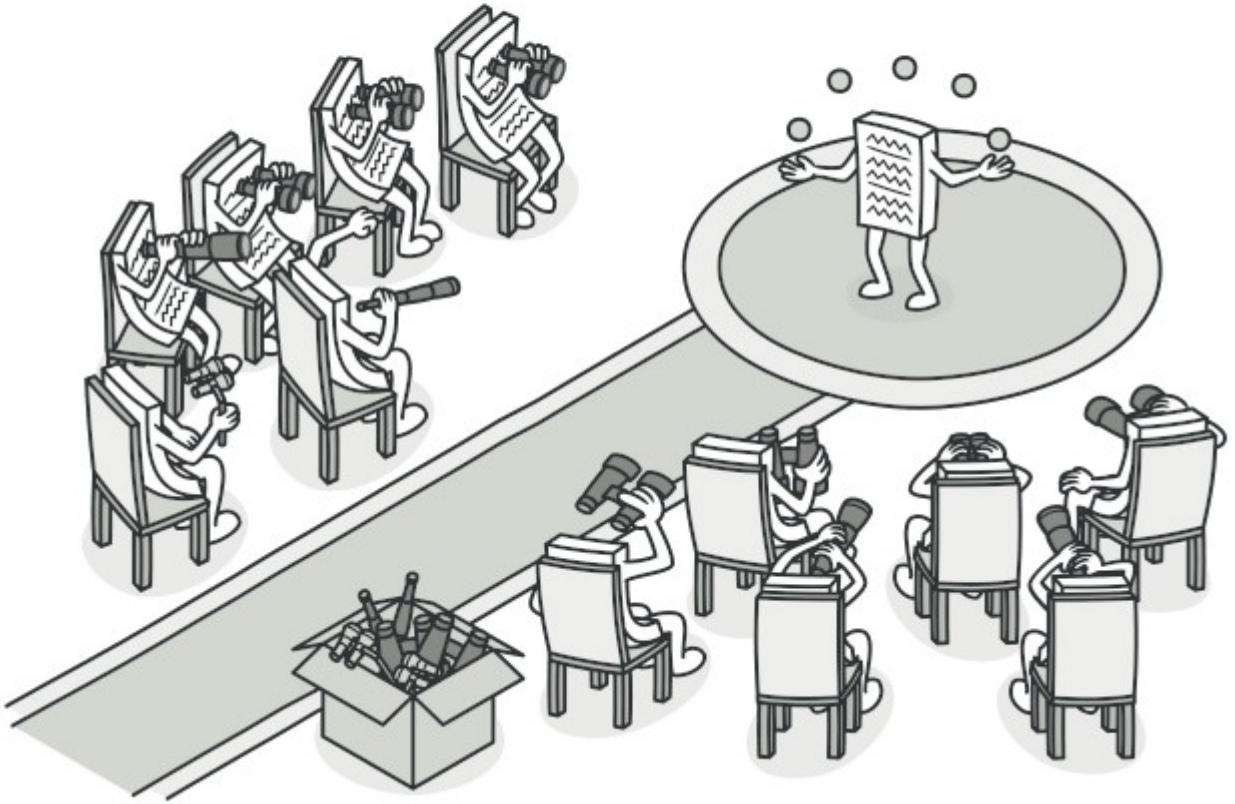
Quais são eles:

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

INSTITUTO FEDERAL
Santa Catarina

Observer

O **Observer** é um padrão de projeto comportamental que permite que você defina um mecanismo de assinatura para notificar múltiplos objetos sobre quaisquer eventos que aconteçam com o objeto que eles estão observando.

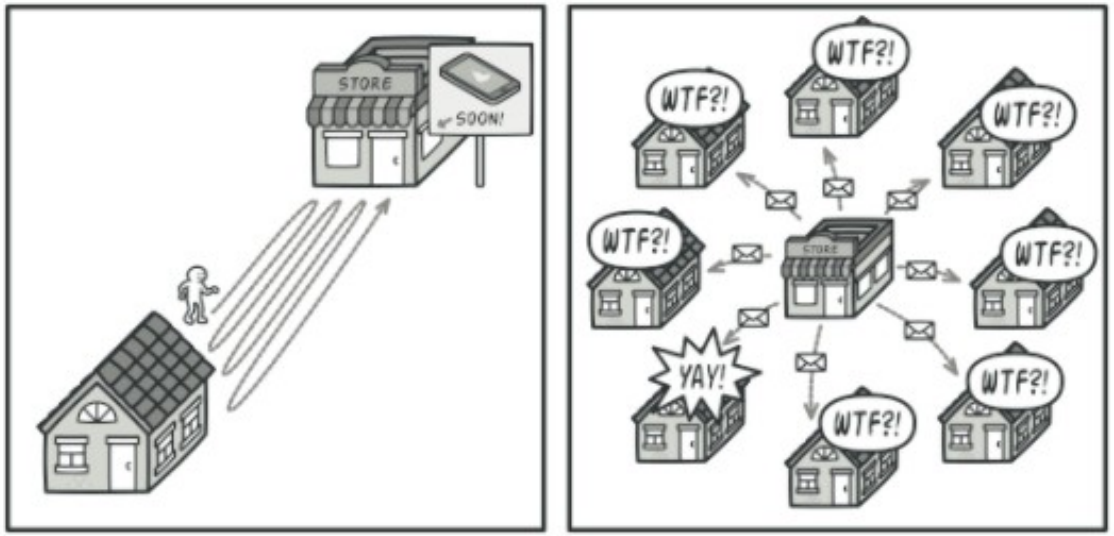


INSTITUTO FEDERAL
Santa Catarina

Observer (Problema)

Imagine que você tem dois tipos de objetos: um `Cliente` e uma `Loja`. O cliente está muito interessado em uma marca particular de um produto (digamos que seja um novo modelo de iPhone) que logo deverá estar disponível na loja.

O cliente pode visitar a loja todos os dias e checar a disponibilidade do produto. Mas enquanto o produto ainda está a caminho, a maioria desses visitas serão em vão.



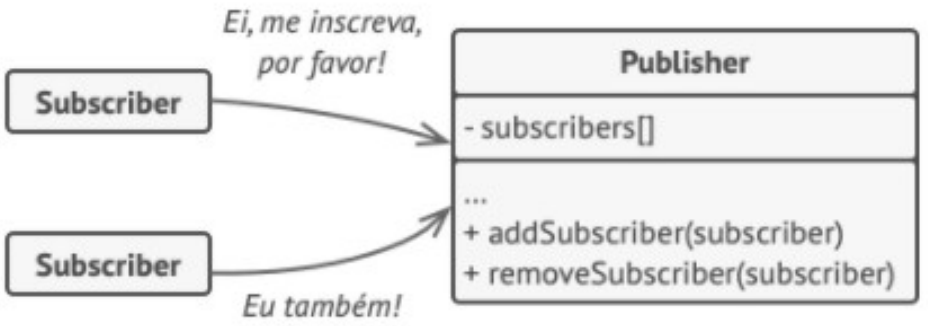
Visitando a loja vs. enviando spam

Por outro lado, a loja poderia mandar milhares de emails (que poderiam ser considerados como spam) para todos os clientes cada vez que um novo produto se torna disponível. Isso salvaria alguns clientes de incontáveis viagens até a loja. Porém, ao mesmo tempo, irritaria outros clientes que não estão interessados em novos produtos.

Parece que temos um conflito. Ou o cliente gasta tempo verificando a disponibilidade do produto ou a loja gasta recursos notificando os clientes errados.

INSTITUTO FEDERAL
Santa Catarina

Observer (Solução)



Um mecanismo de assinatura permite que objetos individuais inscrevam-se a notificações de eventos.

INSTITUTO FEDERAL
Santa Catarina

Observer (Analogia com o mundo real)

Assinaturas de revistas e jornais.

Se você assinar um jornal ou uma revista, você não vai mais precisar ir até a banca e ver se a próxima edição está disponível. Ao invés disso a publicadora manda novas edições diretamente para sua caixa de correio após a publicação ou até mesmo com antecedência.

A publicadora mantém uma lista de assinantes e sabe em quais revistas eles estão interessados. Os assinantes podem deixar essa lista a qualquer momento quando desejarem que a publicadora pare de enviar novas revistas para eles.

INSTITUTO FEDERAL
Santa Catarina

Strategy

Propósito

O **Strategy** é um padrão de projeto comportamental que permite que você defina uma família de algoritmos, coloque-os em classes separadas, e faça os objetos deles intercambiáveis.

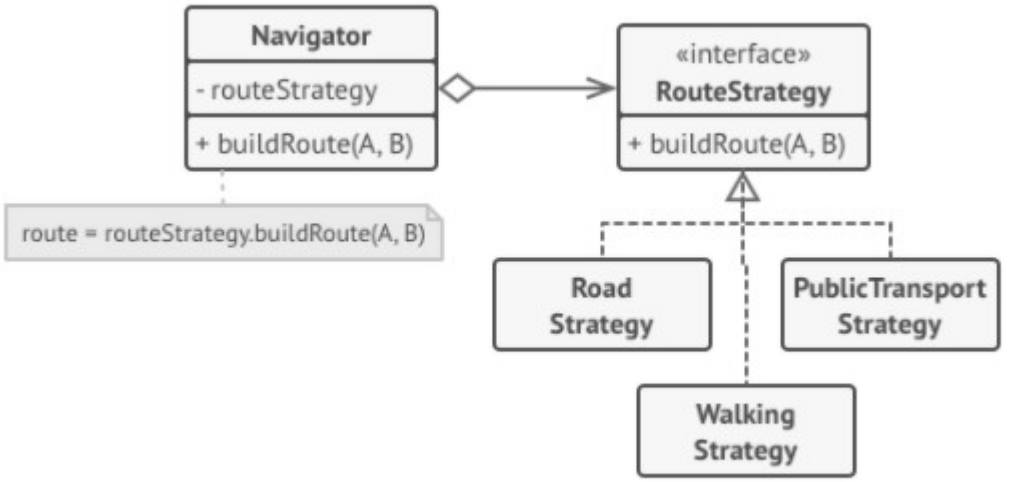
INSTITUTO FEDERAL
Santa Catarina

Strategy (Problema)

- Existem vários algoritmos para um mesmo problema. Exemplo: vários algoritmos de ordenação de array (BubbleSort, QuickSort, etc.)
- O objetivo é implementar e executar diferentes algoritmos usando uma
- mesma interface

INSTITUTO FEDERAL
Santa Catarina

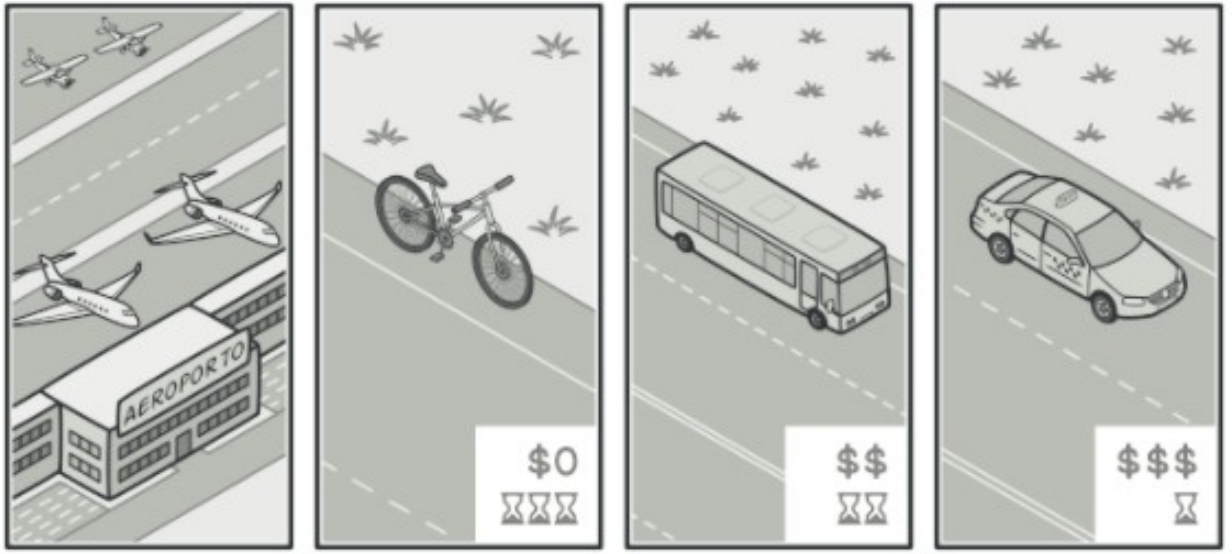
Strategy (Solução)



Estratégias de planejamento de rotas.

INSTITUTO FEDERAL
Santa Catarina

Strategy (Analogia com o mundo real)



Várias estratégias para se chegar ao aeroporto.

Imagine que você tem que chegar ao aeroporto. Você pode pegar um ônibus, pedir um táxi, ou subir em sua bicicleta. Essas são suas estratégias de transporte. Você pode escolher uma das estratégias dependendo de fatores como orçamento ou restrições de tempo.

INSTITUTO FEDERAL
Santa Catarina

Considerações finais

Os padrões de projeto são um kit de ferramentas para soluções tentadas e testadas para problemas comuns em projeto de software. Mesmo que você nunca tenha encontrado esses problemas, saber sobre os padrões é ainda muito útil porque eles ensinam como resolver vários problemas usando princípios de projeto orientado a objetos;

Os padrões de projeto definem uma linguagem comum que você e seus colegas podem usar para se comunicar mais eficientemente. Você pode dizer, “Oh, é só usar um Singleton para isso,” e todo mundo vai entender a ideia por trás da sua sugestão. Não é preciso explicar o que um singleton é se você conhece o padrão e seu nome;

Empresas podem criar o seu próprio padrão de projetos para atender suas necessidades;

INSTITUTO FEDERAL
Santa Catarina

Considerações finais

Qual usar?

Depende do problema, da solução pretendida e possíveis efeitos colaterais;

Preciso conhecer todos os padrões de projetos?
Não, você precisa ter uma noção para que servem, pois quando necessitar poderá se aprofundar.

Não se esqueça padrões de projeto não é um tema fácil e seu conhecimento vem através do estudo e prática na sua utilização e conhecimentos consolidados em POO são pré-requisitos básicos para se trabalhar com eles.