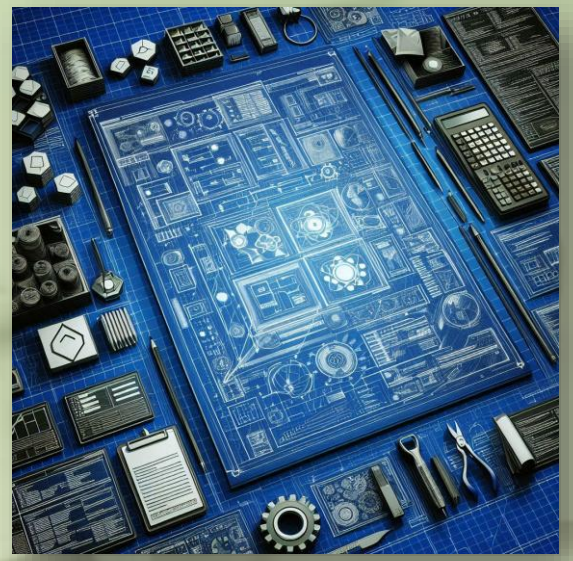
Padrões de Projetos

Prof. Henrique Louro

Introdução aos Padrões de Projeto

- **Definição**: Soluções típicas para problemas comuns em projeto de software.
- **Analogia**: Semelhante a plantas de obras pré-fabricadas.
- **Aplicação**: Não é código específico, mas um conceito adaptável.



- **Diferença entre Padrões e Algoritmos**
  - **Algoritmo**: Passos claros, como uma receita de comida.
  - **Padrão de projeto**: Estrutura geral, como uma planta de obra.
  - **Implementação**: O mesmo padrão pode ser aplicado de maneiras diferentes em programas distintos.

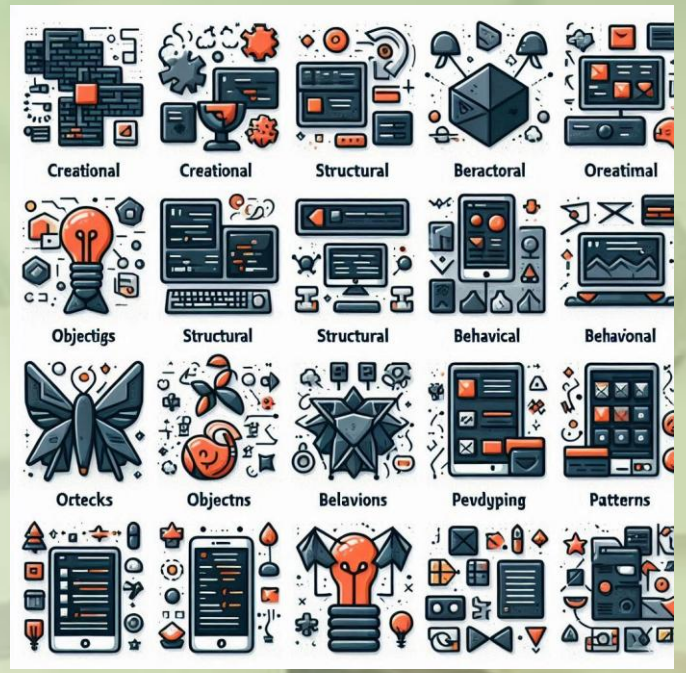
- **Estrutura de um Padrão de Projeto**
  - **Propósito**: Define o problema e solução.
  - **Motivação**: Explica a fundo a necessidade do padrão.
  - **Estruturas de classes**: Mostra componentes e relações.
  - **Exemplos de código**: Facilita a compreensão prática.

Introdução aos Padrões de Projeto

- **Benefícios dos Padrões de Projeto**
  - Soluções reutilizáveis e testadas.
  - Melhoria na comunicação entre desenvolvedores.
  - Uso de uma linguagem comum no desenvolvimento.

- **Tipos de Padrões de Projeto**
  - **Criacionais**: Facilitam a criação de objetos, aumentando a reutilização de código.
  - **Estruturais**: Organizam classes e objetos em estruturas eficientes.
  - **Comportamentais**: Melhoram a comunicação e atribuição de responsabilidades.

- **Alguns Padrões Existentes**
  - Singleton
  - Factory
  - Adapter
  - Strategy
  - Observer

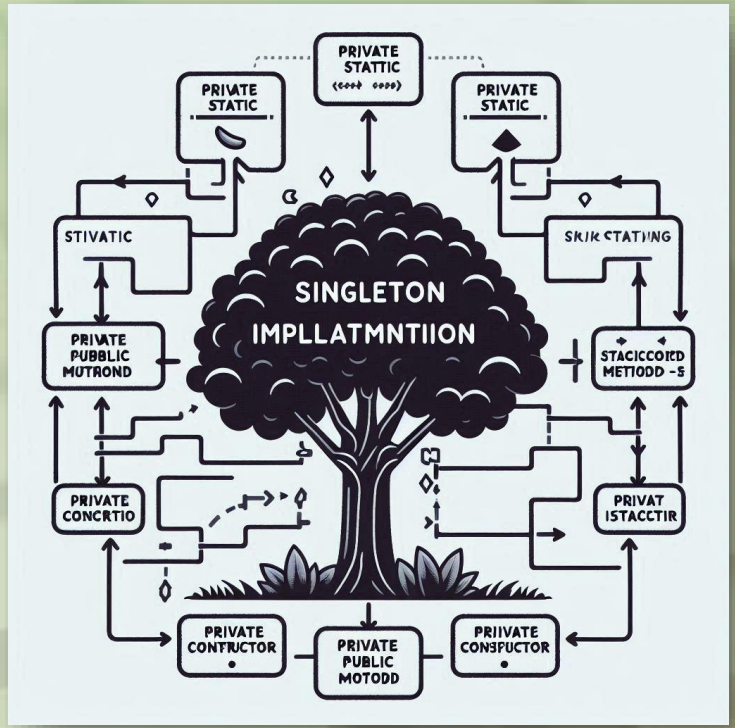


Padrão de Projeto Singleton

Conceito e Aplicabilidade

O que é Singleton?

- Padrão de projeto criacional que garante uma única instância de uma classe e fornece acesso global a ela.
- Soluciona dois problemas:
  1. Restrição de instâncias – evita múltiplas criações e controla acesso a recursos compartilhados.
  2. Ponto de acesso global – similar a variáveis globais, mas mais seguro.



Aplicabilidade

- Quando é necessário garantir uma única instância (exemplo: banco de dados compartilhado).
- Quando há necessidade de um controle mais seguro sobre variáveis globais.

Padrão de Projeto Singleton

Implementação, Prós e Contras

Como Implementar?

1. Criar um campo estático privado para armazenar a instância.
2. Criar um método público estático para acesso à instância.
3. Aplicar inicialização preguiçosa no método de criação.
4. Definir o construtor como privado.
5. Substituir chamadas diretas do construtor pelo método estático.

Vantagens:

- Garante uma única instância.
- Proporciona um ponto de acesso global.
- Inicialização ocorre apenas quando necessário.

Desvantagens:

- Viola o princípio de responsabilidade única.
- Pode mascarar um design ruim.
- Requer cuidados em ambientes multithreaded.
- Dificulta testes unitários.

Fatec
Faculdade de Tecnologia
Jacareí

Padrão de Projeto Factory

Conceito e Problema

O que é Factory Method?

- Padrão criacional que define uma interface para criar objetos.
- Permite que subclasses determinem o tipo de objeto a ser criado.

Problemas:

- Código fortemente acoplado a classes específicas, dificultando mudanças e expansões.
- Modificações exigem alterações extensivas no código base.

Solução:

- Substituir construtores (new) por um método fábrica dentro da classe criadora.
- Subclasses podem definir o tipo de objeto retornado.
- Produtos devem seguir uma interface comum.

Estrutura

- Produto – Interface comum a todos os objetos criados.
- Produtos Concretos – Implementações diferentes da interface.
- Criador – Contém o método fábrica.
- Criadores Concretos – Subclasses que alteram o retorno do método fábrica.

Padrão de Projeto Factory

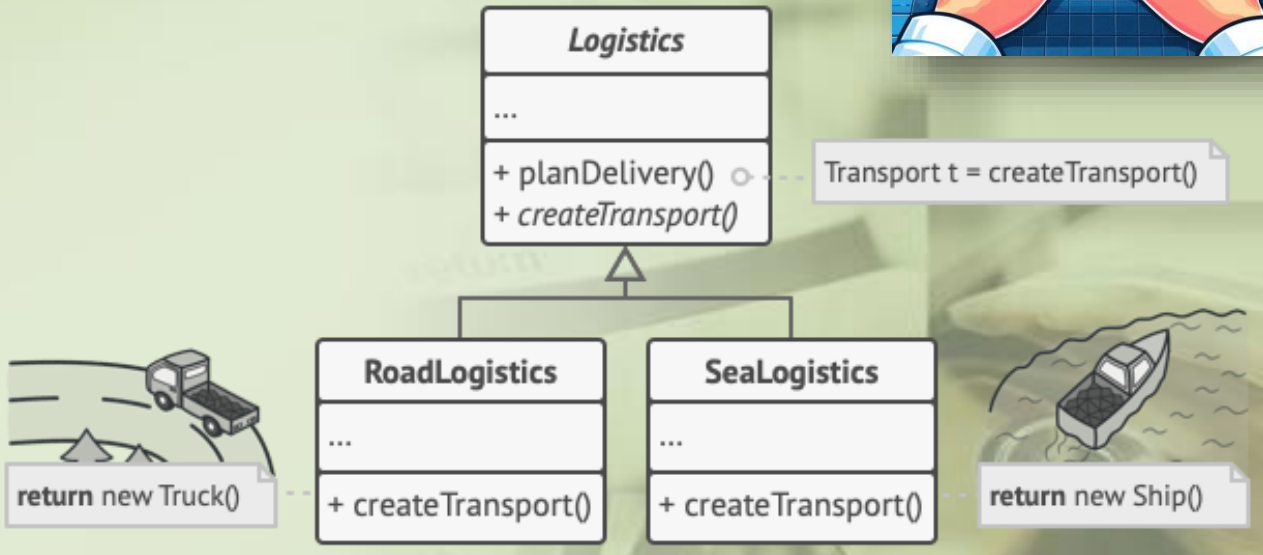
Aplicação e Benefícios

Exemplo Prático

- Criação de elementos de UI multiplataforma (botões Windows/Linux).
- Facilita a expansão sem modificar código existente.

Vantagens

- Maior flexibilidade e desacoplamento.
- Facilita expansão sem grandes mudanças.
- Melhora organização do código.



Alteração de classe de objetos

- Subclasses podem modificar o tipo de objeto retornado pelo método fábrica.

Motivação da mudança

- Inicialmente, parece uma simples realocação da chamada do construtor.
- A vantagem real surge ao permitir a sobrescrita do método fábrica.

Limitação da abordagem

- Subclasses só podem retornar objetos de classes com uma interface ou classe base em comum.

Fatec
Faculdade de Tecnologia
Jacareí

Padrão de Projeto Adapter

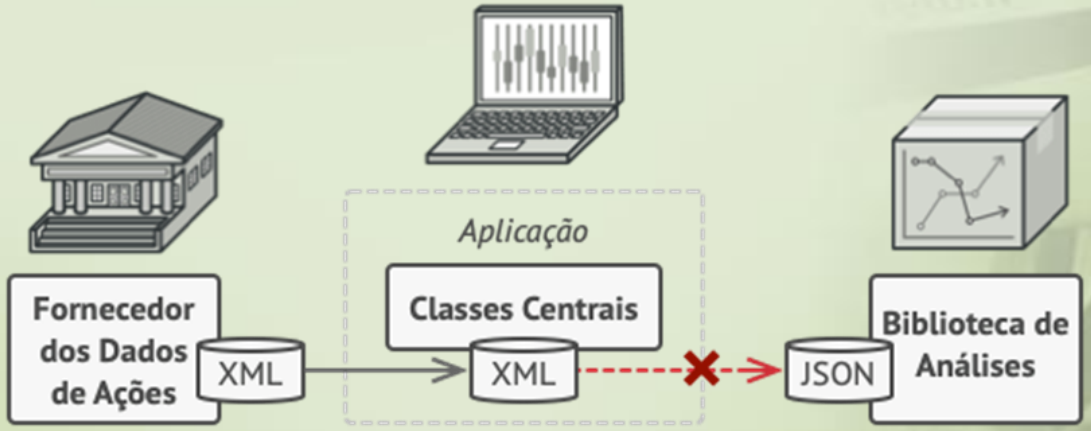
Conceito e Problema

O que é o padrão Adapter?

- Também chamado de Adaptador ou Wrapper.
- Permite que objetos com interfaces incompatíveis colaborem.

Problema

- Uma aplicação recebe dados em XML, mas uma biblioteca de análise exige JSON.
- Modificar a biblioteca pode ser impossível, especialmente se o código fonte não estiver acessível.

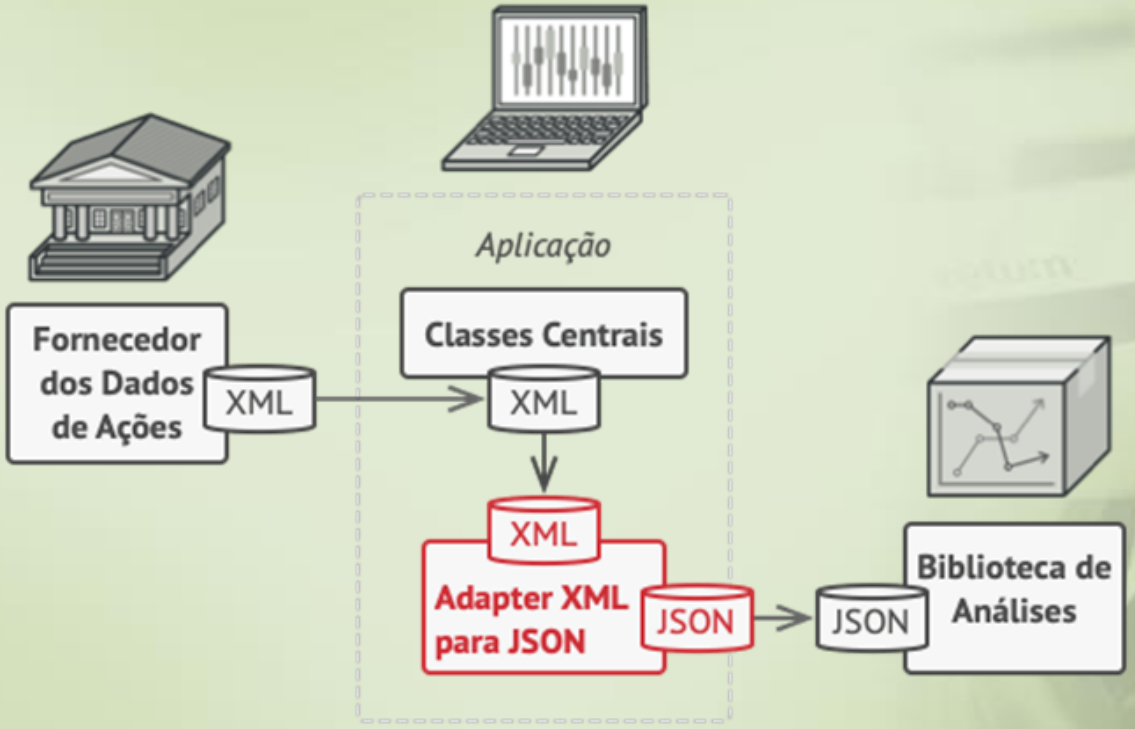


Padrão de Projeto Adapter

Solução

Solução

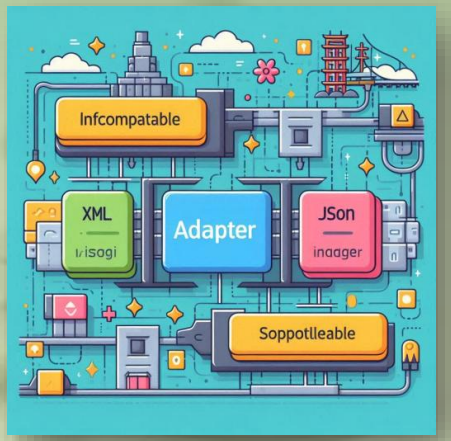
- Criar um adaptador, que converte a interface para ser compatível.
- Encobrimento do objeto, permitindo colaboração sem que ele perceba a conversão.
- Exemplo: Um adaptador XML-para-JSON ajusta o formato antes de enviar os dados à biblioteca.

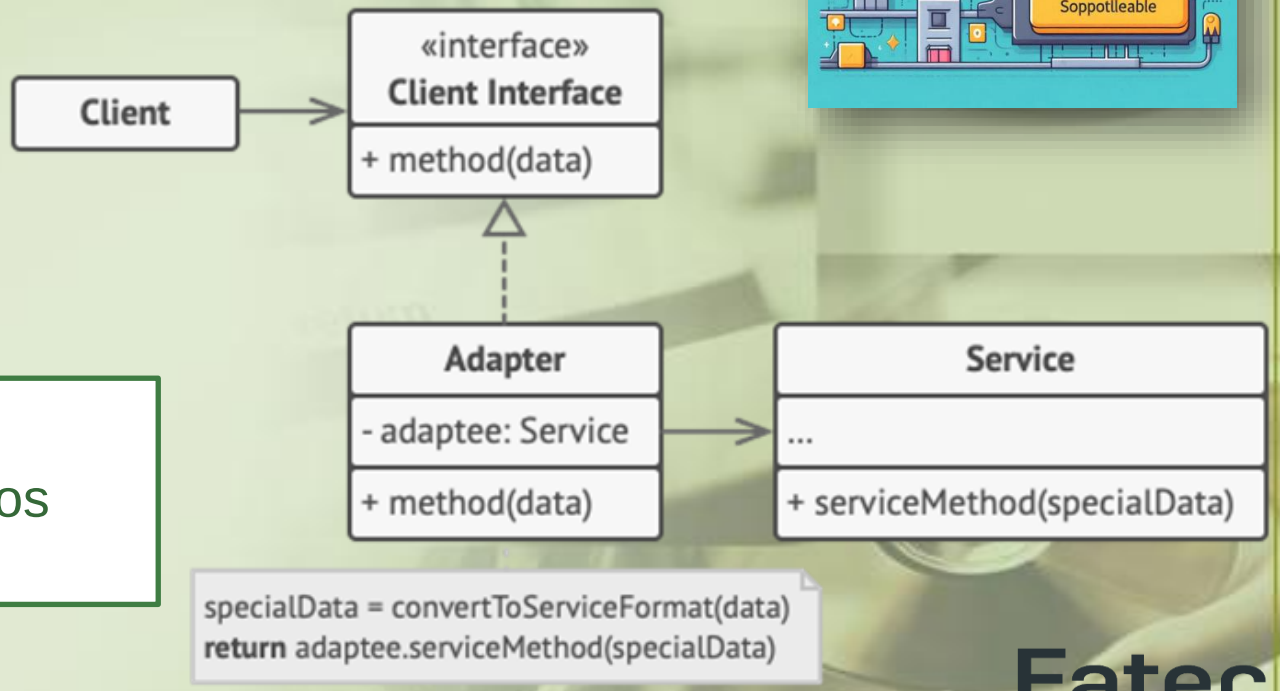


Padrão de Projeto Adapter

Estrutura

- **Cliente** – Lógica de negócio da aplicação existente.
- **Interface do Cliente** – Define protocolo de comunicação com o cliente.
- **Adaptador** – Conecta cliente e serviço, convertendo dados e chamadas.
- **Serviço** – Classe útil, mas com interface incompatível.





Código Cliente

Permanece desacoplado, podendo usar diversos adaptadores.

Dúvidas

Fatec
Faculdade de Tecnologia
Jacareí

FIM

Obrigado !

Prof. Henrique Louro

henrique.louro@fatec.sp.gov.br

Fatec
Faculdade de Tecnologia
Jacareí