
Padrões de Projeto Bridge

— Dra. Alana Morais —

Objetivo da Aula

Entender e aplicar conceitos relacionados ao padrão Bridge

Será que os conceitos anteriores estão claros?

Questão 1

Julgue o item como Verdadeiro ou Falso. "Adapter é um padrão estrutural utilizado para compatibilizar interfaces de modo que elas possam interagir".

Questão 2

O Governo Federal, por meio do Banco Central, está desenvolvendo um sistema que possibilitará a todos os Bancos do país o acesso a algumas de suas informações. Um requisito fundamental desse sistema é que a taxa de juros utilizada em todas as transações de todos os Bancos seja a mesma e haja um único acesso a essa informação. Além disso, esse sistema deve poder ser executado em diferentes plataformas, como computadores e diversos dispositivos móveis.

Para garantir que a taxa de juros seja única e para evitar a necessidade de criar diferentes soluções para cada plataforma, a empresa desenvolvedora decidiu adotar padrões de projeto.

O primeiro padrão deverá garantir uma única instanciamento para a classe “Taxa de juros”; e o segundo padrão deverá definir uma família de componentes para cada plataforma e uma implementação que os instancie de acordo com a plataforma na qual a aplicação estará sendo executada.

- a) Prototype e Adapter;
- b) Singleton e Abstract Factory;
- c) Template Method e Prototype;
- d) Adapter e Singleton;
- e) Abstract Factory e Command.

Introdução

Focaremos sempre em discutir sobre o nome, problema, solução e consequências de usar um padrão.

Existem diversas estratégias especiais para agrupar os padrões de projeto

Seguiremos a classificação de Gamma.

Padrões GoF – Padrões de Estrutura

		Propósito		
		1. Criação	2. Estrutura	3. Comportamento
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

PADRÕES GOF

CLASSIFICAÇÃO – METSKER

Intenção	Padrões
1. Interfaces	<i>Adapter, Facade, Composite, Bridge</i>
2. Responsabilidade	<i>Singleton, Observer, Mediator, Proxy, Chain of Responsibility, Flyweight</i>
3. Construção	<i>Builder, Factory Method, Abstract Factory, Prototype, Memento</i>
4. Operações	<i>Template Method, State, Strategy, Command, Interpreter</i>
5. Extensões	<i>Decorator, Iterator, Visitor</i>

Questão 3

Sabendo-se que os padrões de projeto podem ser classificados considerando-se o critério de escopo, qual, dentre os padrões de projeto abaixo, possui escopo de classe?

- a) Adapter
- b) Decorator
- c) Mediator
- d) Proxy
- e) Observer

Bridge

"DESACOPLAR UMA ABSTRAÇÃO DE SUA IMPLEMENTAÇÃO PARA QUE OS DOIS POSSAM VARIAR INDEPENDENTEMENTE."

Bridge

Intenção:

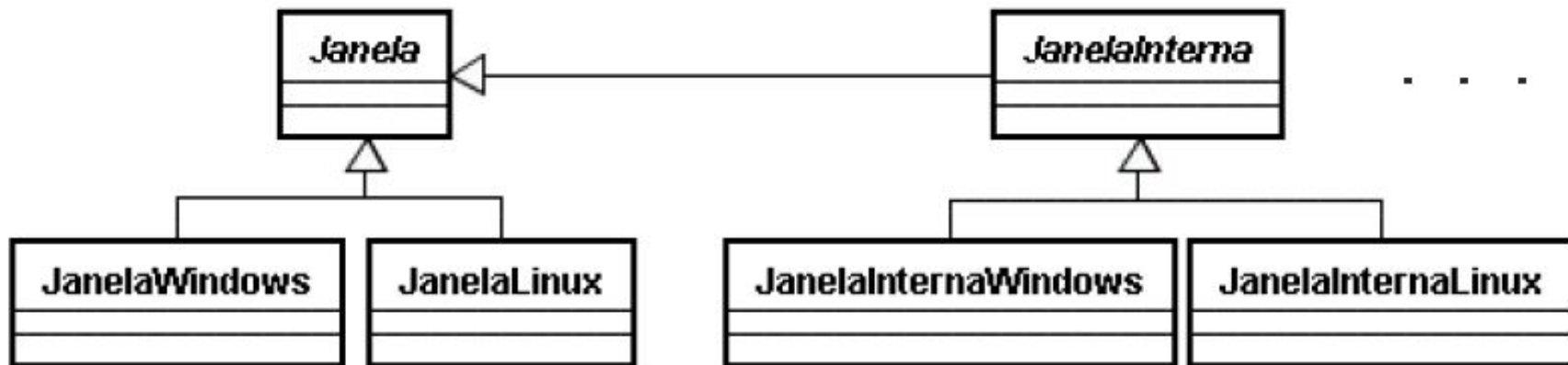
- Desacoplar uma abstração de sua implementação para que ambos possam variar independentemente.

Também conhecido como:

- Handle, Body
- Exemplo: clientes (programadores) somente precisam interagir com JFrame, JButton, etc. ao manipular interfaces gráficas.

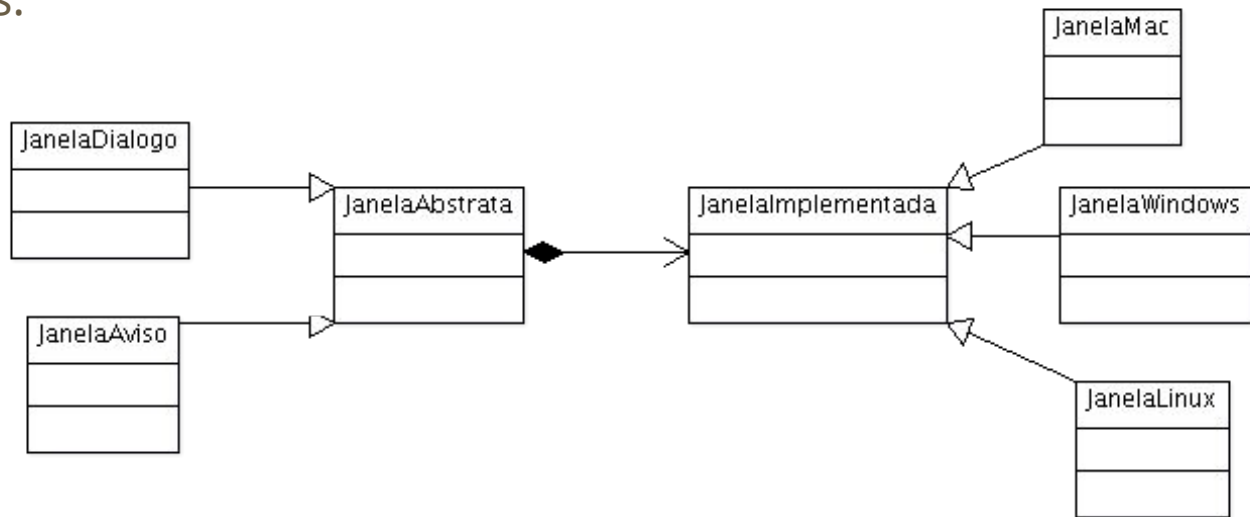
Bridge - Problema

- Componentes gráficos devem ser implementados para várias arquiteturas;
- Cada novo componente exige várias implementações;
- Cada nova arquitetura também.

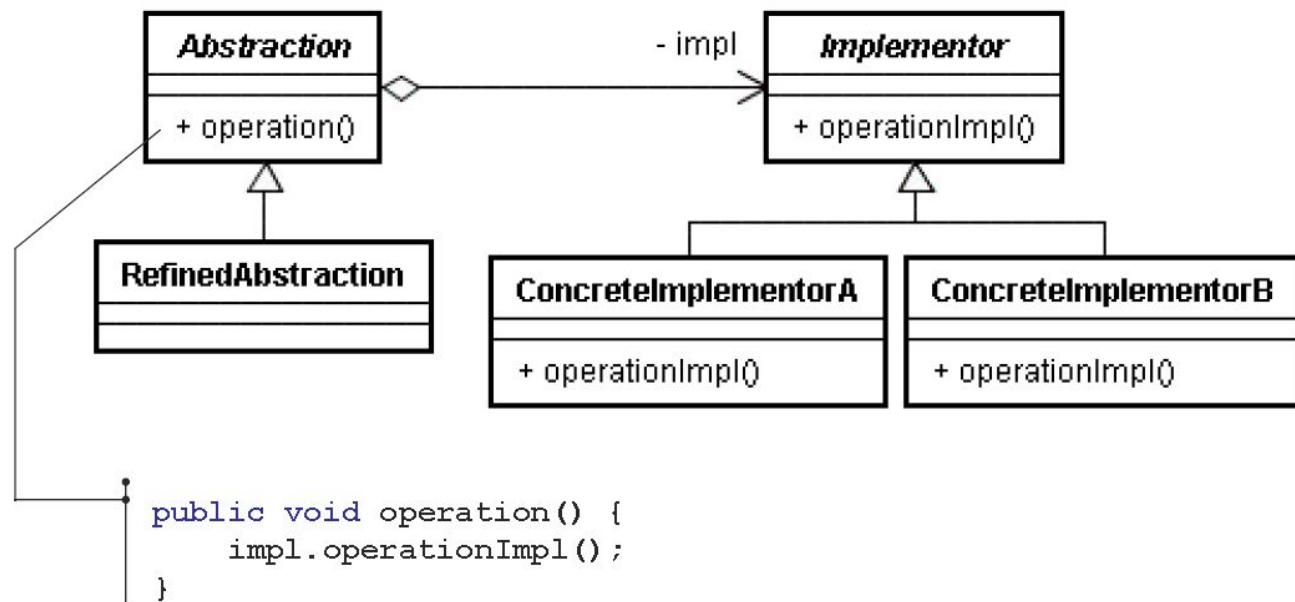


Bridge - Solução

- Janela concentra métodos que utilizam recursos específicos de plataforma;
- Subclasses utilizam os métodos de Janela para implementar itens específicos.



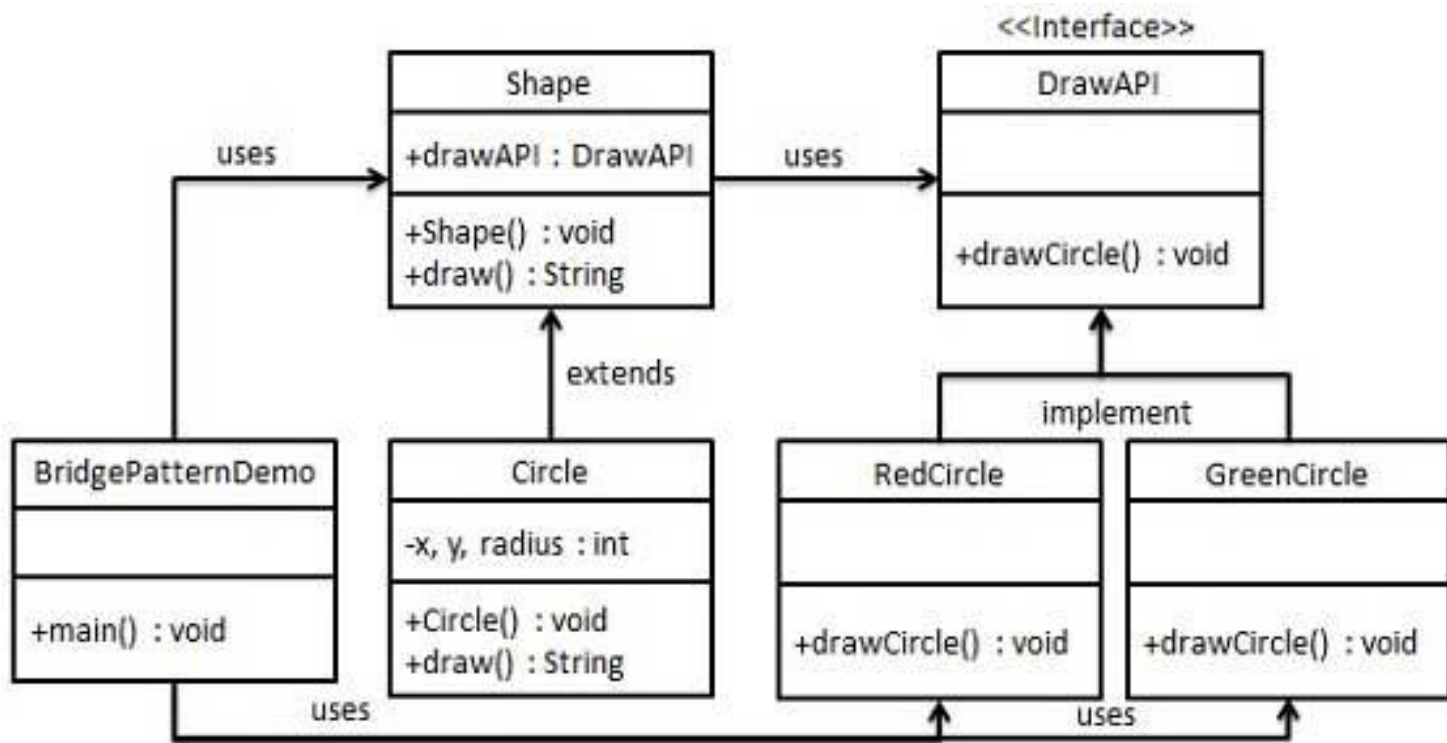
Bridge - Estrutura



Bridge - Estrutura

- Abstraction
 - Define a interface de abstração. Mantém uma referência a um objeto do tipo Implementador.
- RefinedAbstraction
 - Estende a interface definida por Abstração.
- Implementor
 - Define a interface para classes de implementação. Esta não tem a obrigação de corresponder exatamente à interface de abstração. De fato, as duas interfaces podem ser bastante diferentes. Tipicamente, a interface de implementação fornece apenas operações primitivas, cabendo à abstração a responsabilidade de definir operações de alto nível baseadas nestas primitivas.
- ConcretImplementorA e ConcretImplementorB
 - Implementação concreta da interface definida por Implementador.

Bridge - Exemplo 2



Bridge - Quando Usar?

- Quando for necessário evitar uma ligação permanente entre a interface e implementação.
- Quando alterações na implementação não puderem afetar clientes.
- Quando tanto abstrações como implementações precisarem ser capazes de suportar extensão por meio da herança.
- Quando implementações são compartilhadas entre objetos desconhecidos do cliente.

Bridge - Vantagens e desvantagens

- Desacopla a implementação:
 - Podendo até mudá-la em tempo de execução.
- Melhora a extensibilidade:
 - É possível estender a abstração e a implementação separadamente.
- Esconde detalhes de implementação:
 - Clientes não precisam saber como é implementado.

Questão

Analise as seguintes afirmações:

I. Fornece uma interface para a criação de uma família de objetos relacionados ou dependentes sem fornecer os detalhes de implementação das classes concretas.

II. Converte uma interface de uma classe existente em outra interface esperada pelos clientes. Permite que algumas classes com interfaces diferentes trabalhem em conjunto.

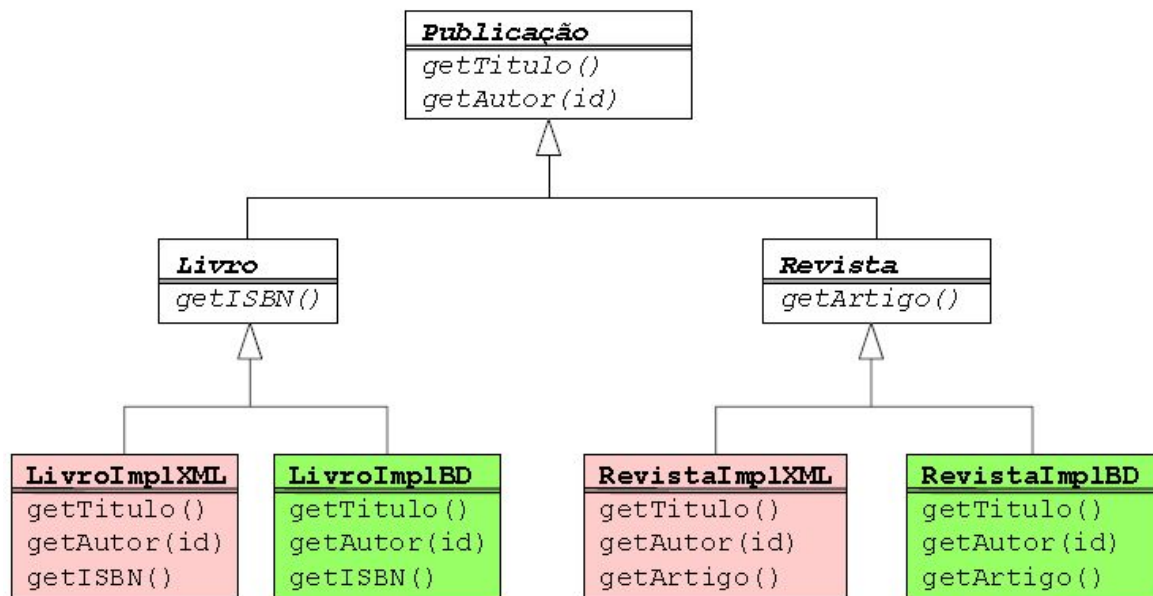
III. Separa uma implementação de sua abstração, de forma que ambas possam variar independentemente.

IV. Separa a construção de um objeto complexo de sua representação, de modo que o mesmo processo possa criar representações diferentes.

- a) Builder - Adapter - Bridge - Abstract Factory.
- b) Abstract Factory - Adapter - Bridge - Builder.
- c) Bridge - Adapter - Builder - Abstract Factory.
- d) Adapter - Builder - Abstract Factory - Bridge.
- e) Builder - Bridge - Abstract Factory - Adapter.

Exercício

Melhore a arquitetura apresentada a seguir utilizando o padrão Bridge. A estrutura abaixo serve para lidar com implementações específicas para tratar um objeto em diferentes meios persistentes em um Editora.



Dúvidas?

alanamm.prof@gmail.com