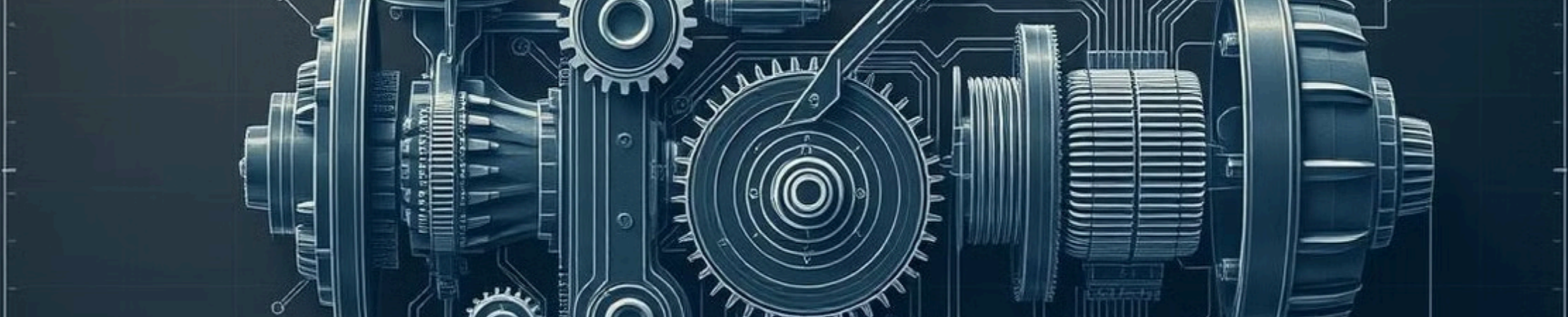




Padrão de Projeto Prototype: Criação Inteligente de Objetos

L por Lucas Feitosa Almeida Rocha



O que é o Padrão Prototype?



Criação por Cópia

Permite copiar objetos sem depender das classes.



Solução Eficiente

Criação de cópias exatas de objetos complexos.

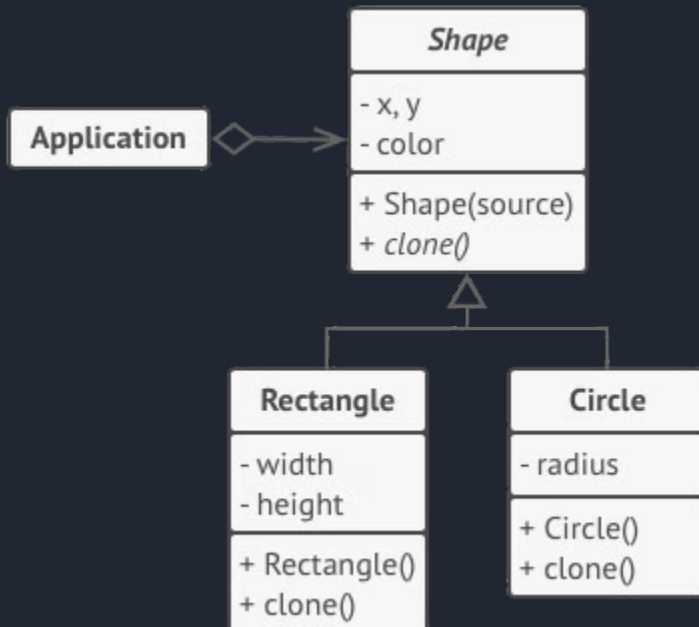


Clonagem

Importante para clonagem eficiente na programação.

O padrão Prototype é um padrão de criação que possibilita a criação de novos objetos através da cópia de objetos existentes. Ele reduz a necessidade de instanciar novas classes.

Estrutura Básica do Padrão Prototype



1

Interface Prototype

Método `clone()` para criar cópias.

2

Classes Concretas

Implementam a interface Prototype.

3

Clonagem Rasa/Profunda

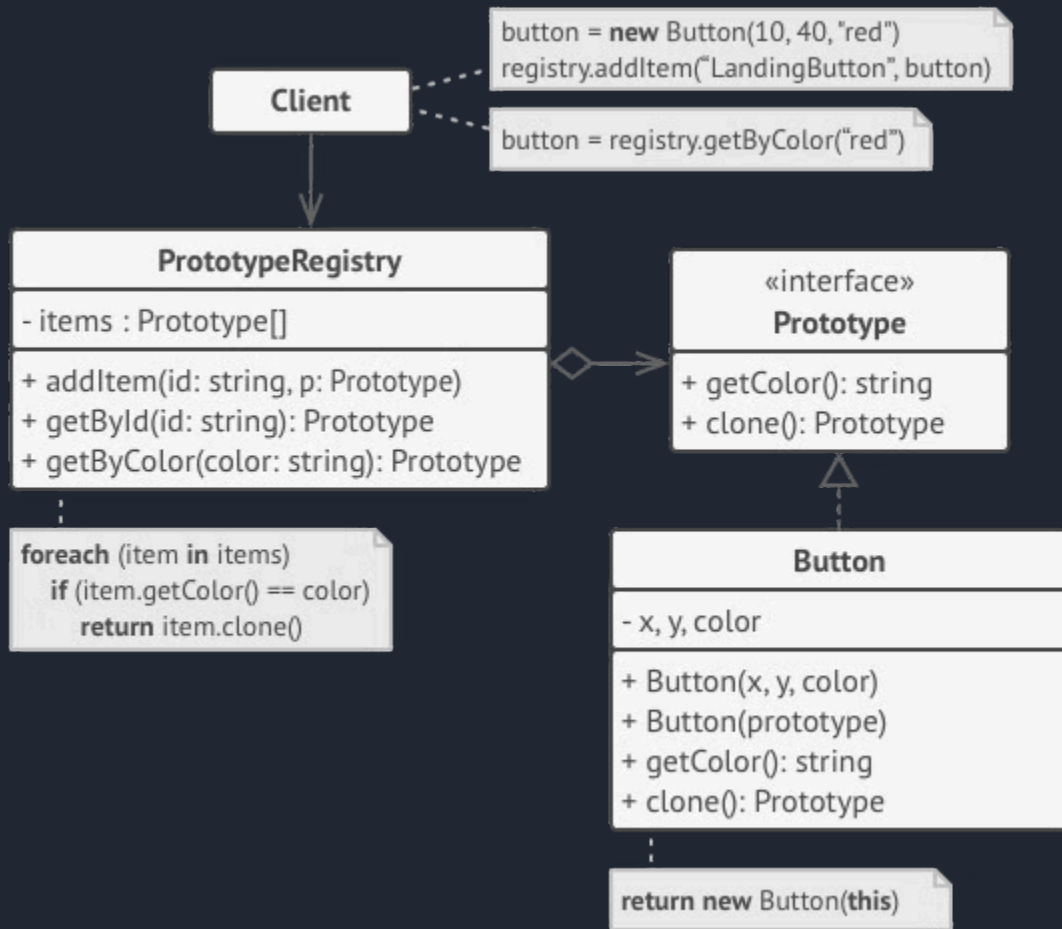
Implementações para diferentes necessidades.

4

Registro de Protótipos

Opcional, para acesso fácil aos protótipos.

A estrutura básica do padrão Prototype envolve uma interface que declara o método `clone()`. As classes concretas implementam essa interface.



Implementação de Registro

O registro armazena protótipos pré-construídos prontos para serem clonados.

Vantagens do Padrão Prototype



Código Limpo

Redução de código repetitivo.



Desacoplamento

Criação sem acoplamento com classes.



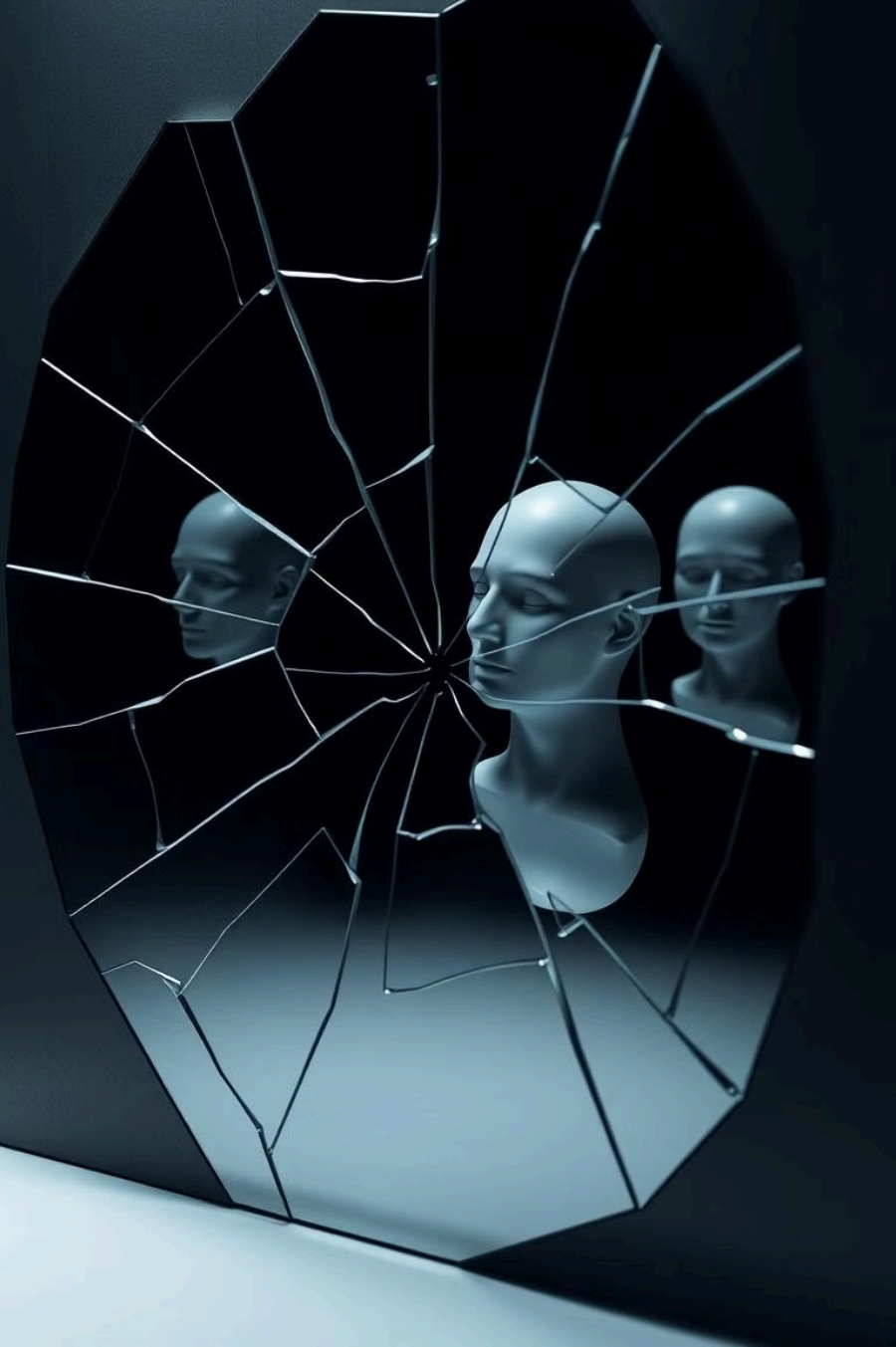
Performance

Maior rapidez na criação de objetos.



Flexibilidade

Geração flexível de novos objetos.



Desvantagens do Padrão Prototype



Complexidade

Aumento na complexidade do código.



Referências

Possíveis problemas de referências circulares.



Overhead

Consumo excessivo de memória.

Conclusão: Quando Usar Prototype



O Prototype é ideal para objetos complexos e sistemas com criação dinâmica. Avalie alternativas como Factory Method e o custo-benefício da implementação.