

Design Patterns

O conceito de padrões de projeto foi primeiramente descrito por Christopher Alexander, em seu livro “Uma Linguagem de Padrões”. A ideia foi seguida por quatro autores: Erich Gamma, John Vlissides, Ralph Johnson, e Richard Helm, que publicaram em 1994 o livro “Padrões de Projeto — Soluções Reutilizáveis de Software Orientado a Objetos”, no qual eles aplicaram o conceito de padrões de projeto para programação mostrando 23 padrões que resolvem vários problemas de projeto orientado a objetos. Devido a seu longo título, as pessoas começaram a chamá-lo simplesmente de “o livro da Gangue dos Quatro (Gang of Four)” que logo foi simplificado para o “livro GoF”.

Os padrões de projeto são um kit de ferramentas para soluções tentadas e testadas para problemas comuns em projeto de software e definem uma linguagem comum que pode ser usada para se comunicar mais eficientemente.

Classificação

Padrões Criacionais

Fornecem mecanismos de criação de objetos que aumentam a flexibilidade e a reutilização de código:

- Factory Method
- Abstract Factory
- Builder
- Prototype
- Singleton

Padrões Estruturais

Explicam como montar objetos e classes em estruturas maiores, enquanto ainda mantém as estruturas flexíveis e eficientes:

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

Padrões Comportamentais

Cuidam da comunicação eficiente e da assinalação de responsabilidades entre objetos:

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

Decorator e Proxy

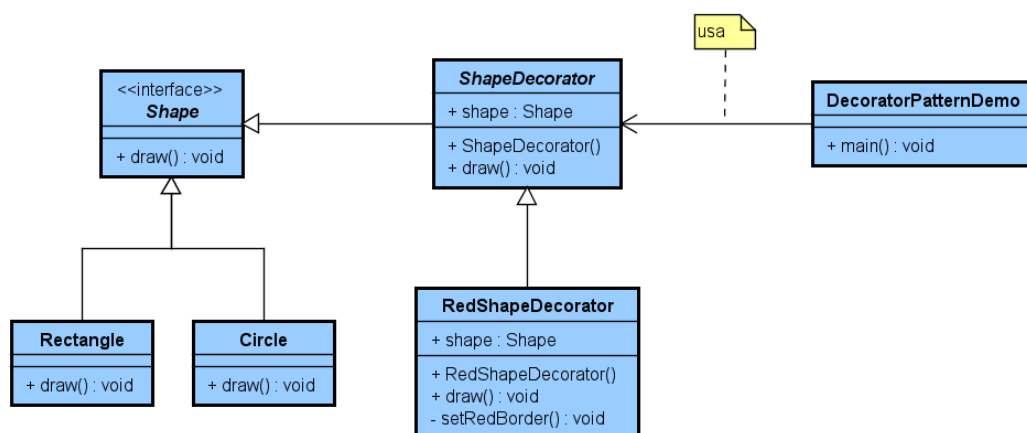
Decorator

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

Ele resolve o problema da complicação ou impossibilidade de estender o comportamento de um objeto usando herança, por exemplo quando a classe é *final*.

Sua aplicação é referente à quando se faz necessário projetar comportamentos adicionais para objetos em tempo de execução sem quebrar o código que usa esses objetos.

Exemplo de aplicação do padrão Decorator (Diagrama de Classe):



Proxy

O Proxy é um padrão de projeto estrutural que fornece um objeto que atua como um substituto para um objeto de serviço real usado por um cliente, recebendo suas solicitações e as transmitindo para o real.

Ele resolve o problema do controle de acesso a objetos, por exemplo, quando se tem um objeto enorme que consome muitos recursos de sistema e é utilizado ocasionalmente.

Sua aplicação é útil em situações onde se deseja controlar o acesso a um objeto ou recurso de maneira mais granular, permitindo ou negando o acesso conforme necessário.

Exemplo de aplicação do padrão Proxy (Diagrama de Classe):

