



ARQUITETURA DE SOFTWARE

Padrões de projeto estruturais

Geucimar Briatore
geucimar@up.edu.br

Atualizado em 10/2022

Padrões de projeto do grupo estrutural

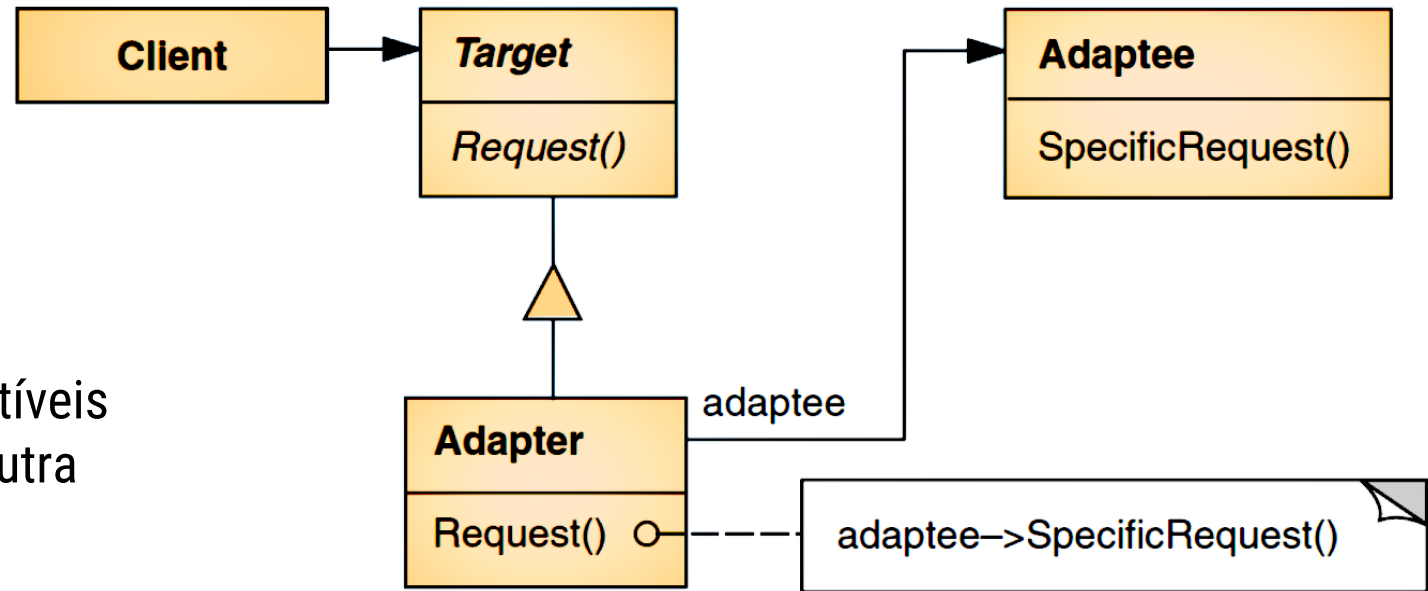
		Propósito		
		1. Criacional	2. Estrutural	3. Comportamental
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento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

Adapter (Objeto)

Tipo: Estrutural

Propósito: converter a interface de uma classe em outra interface, esperada pelos clientes. Permite que classes com interfaces incompatíveis trabalhem em conjunto – o que, de outra forma, seria impossível;

Aplicação: utilizar quando existe uma classe, mas sua interface não corresponde à interface de que necessita; for preciso criar uma classe reutilizável que coopere com classes não-relacionadas ou não-previstas, ou seja, classes que não necessariamente tenham interfaces compatíveis; (somente para adaptadores de objetos) for preciso usar várias subclasses existentes, porém, for impraticável adaptar essas interfaces criando subclasses para cada uma.



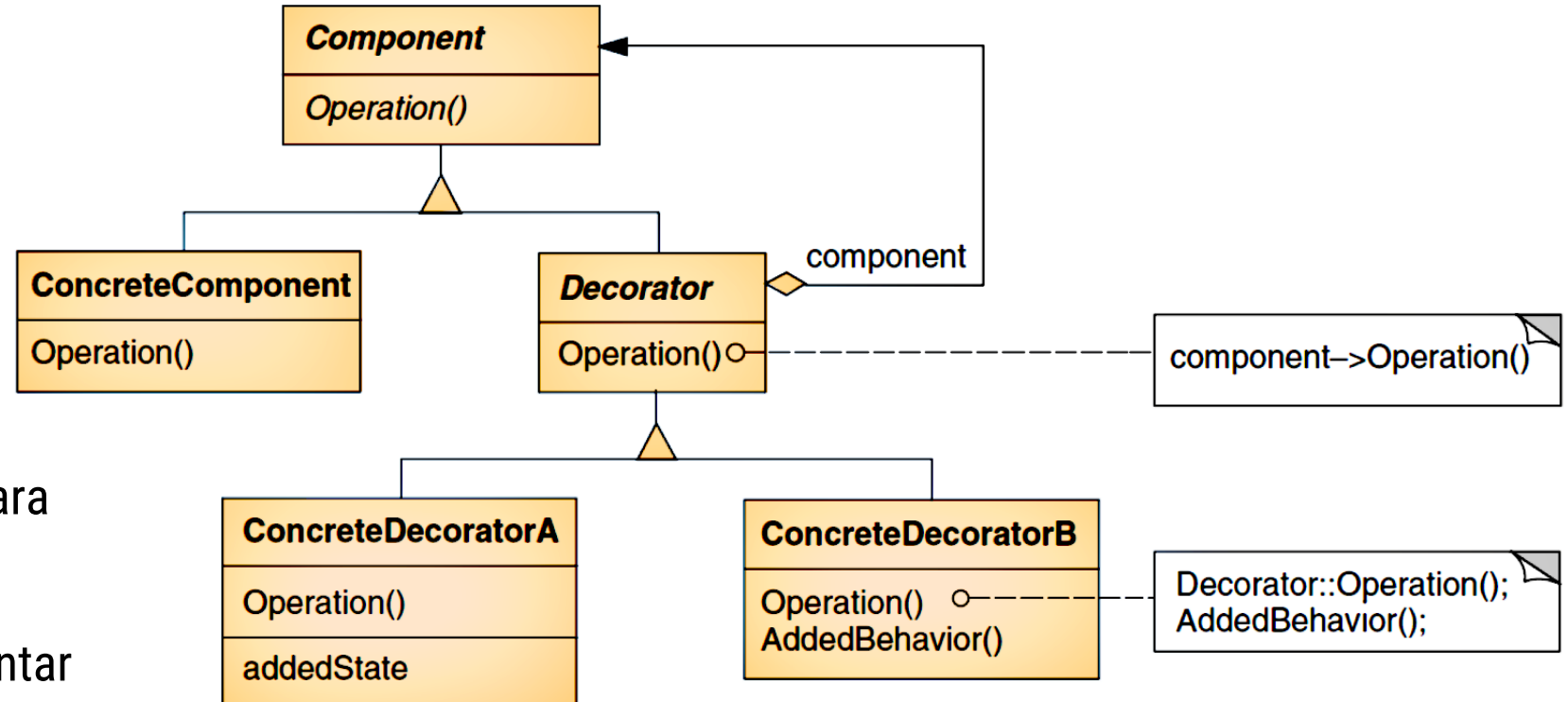
Decorator

Tipo: Estrutural

Propósito: dinamicamente, agregar responsabilidades adicionais a um objeto.

Fornecem uma alternativa flexível ao uso de subclasses para extensão de funcionalidades;

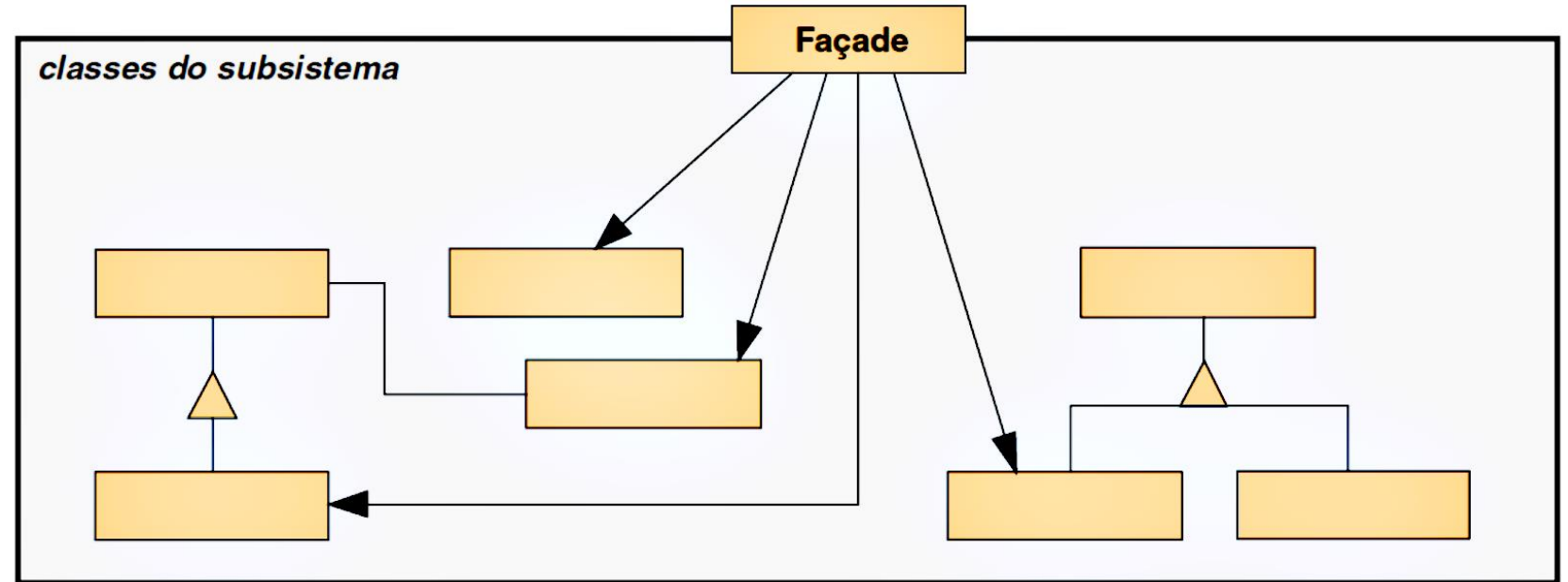
Aplicação: utilizar para acrescentar responsabilidades a objetos individuais de forma dinâmica e transparente, ou seja, sem afetar outros objetos; para responsabilidades que podem ser removidas; quando a extensão através do uso de subclasses não é prática. Às vezes, um grande número de extensões independentes é possível e isso poderia produzir uma explosão de subclasses para suportar cada combinação. Ou a definição de uma classe pode estar oculta ou não estar disponível para a utilização de subclasses.



Façade

Tipo: Estrutural

Propósito: fornecer uma interface unificada para um conjunto de interfaces em um subsistema. Define uma interface de nível mais alto que torna o subsistema mais fácil de ser usado;



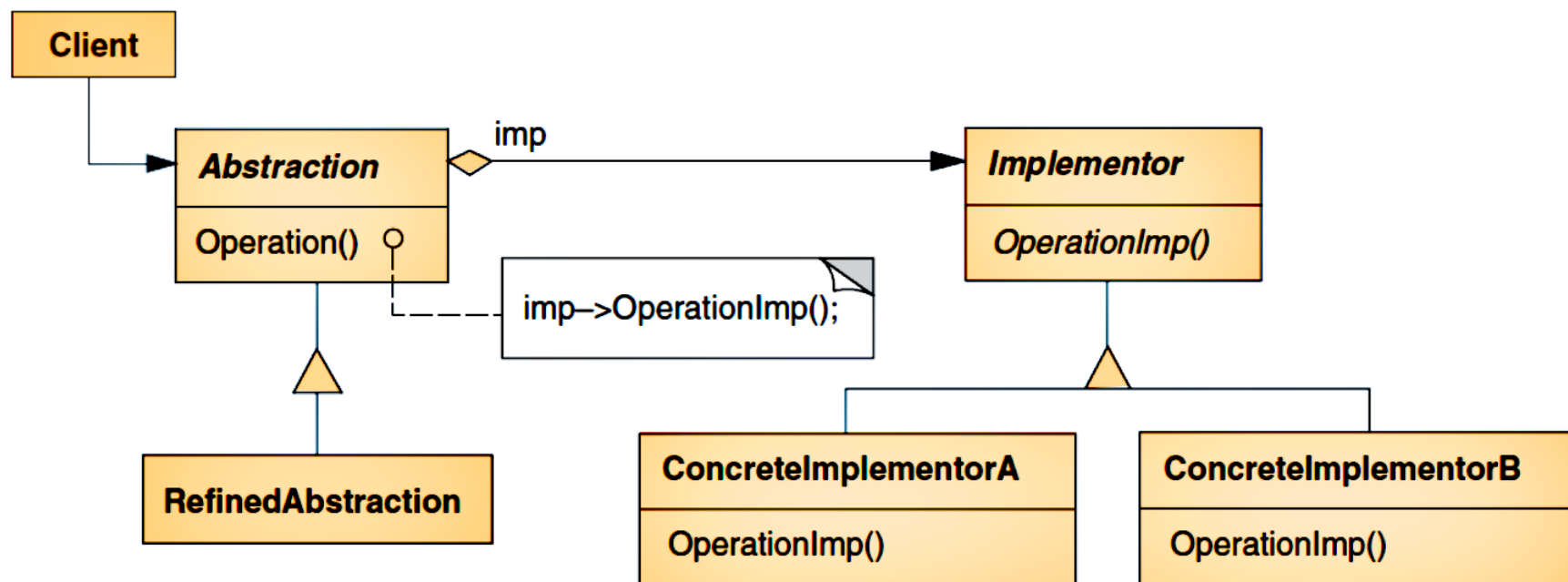
Aplicação: utilizar quando se deseja fornecer uma interface simples para um subsistema complexo; uma fachada pode fornecer, por comportamento-padrão, uma visão simples do sistema, que é boa o suficiente para a maioria dos clientes; existirem muitas dependências entre clientes e classes de implementação de uma abstração; ao introduzir uma fachada para desacoplar o subsistema dos clientes e de outros subsistemas, estar-se-á promovendo a independência e portabilidade dos subsistemas; você deseja estruturar seus subsistemas em camadas; use uma fachada para definir o ponto de entrada para cada nível de subsistema.

Bridge (Não exemplificado)

Tipo: Estrutural

Propósito: desacoplar uma abstração da sua implementação, de modo que as duas possam variar independentemente;

Aplicação: utilizar quando deve-se evitar um vínculo permanente entre uma abstração e sua implementação. Isso pode ocorrer, por exemplo, quando a implementação deve ser selecionada ou alterada em tempo de execução; tanto as abstrações como suas implementações tiverem de ser extensíveis por meio de subclasses. Neste caso, o padrão permite combinar as diferentes abstrações e implementações e estendê-las independentemente.

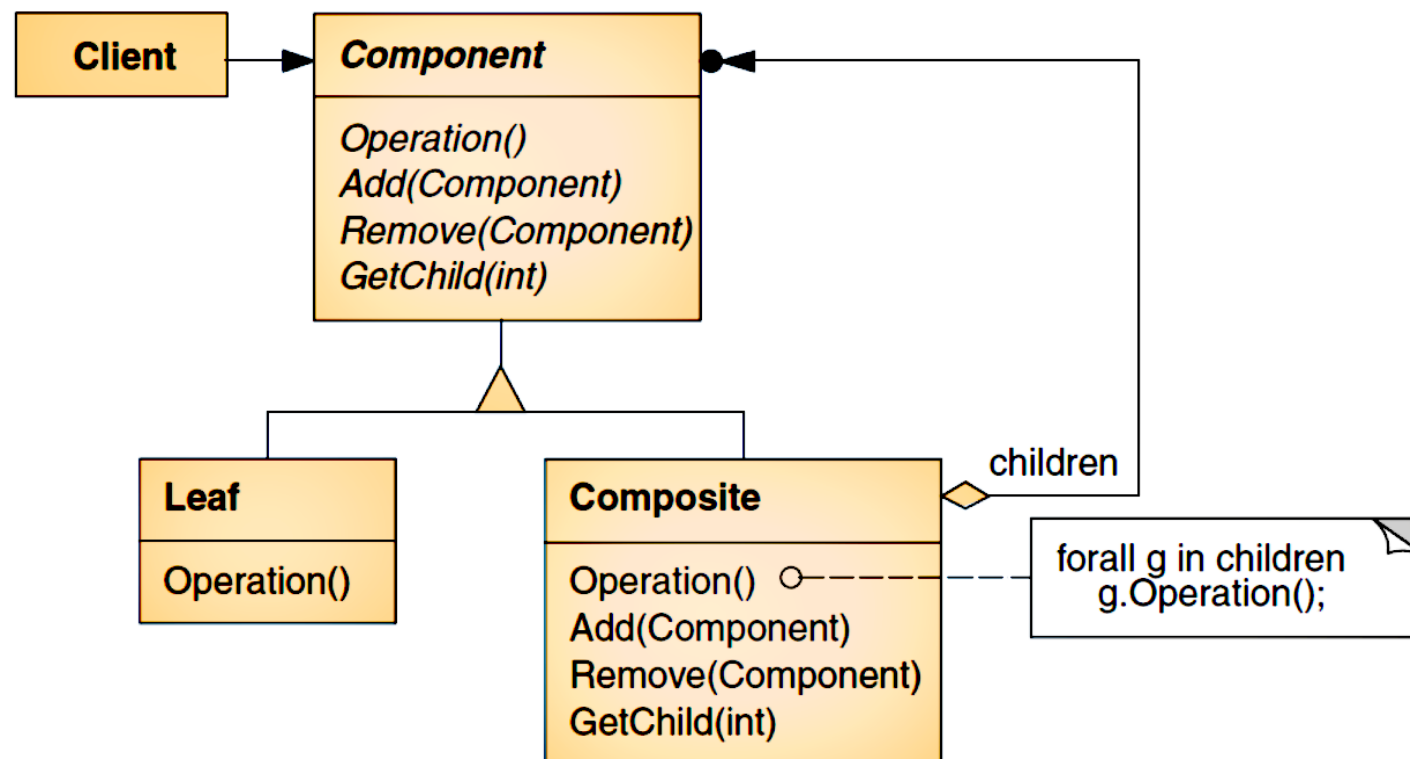


Composite (Não exemplificado)

Tipo: Estrutural

Propósito: compor objetos em estruturas de árvore para representarem hierarquias partes-todo. Permite aos clientes tratarem de maneira uniforme objetos individuais e composições de objetos;

Aplicação: utilizar quando se quiser representar hierarquias partes-todo de objetos; quiser que os clientes sejam capazes de ignorar a diferença entre composições de objetos e objetos individuais. Os clientes tratarão todos os objetos na estrutura composta de maneira uniforme.

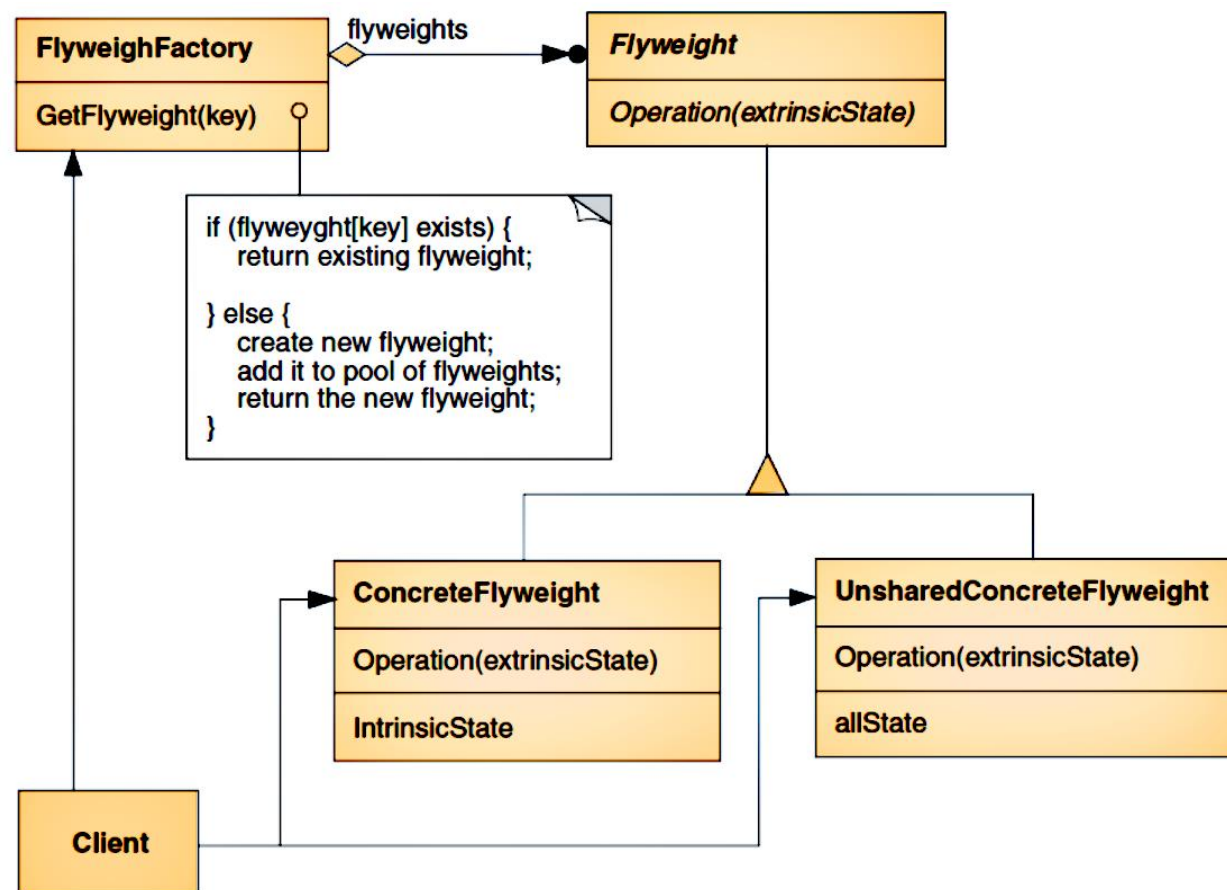


Flyweight (Não exemplificado)

Tipo: Estrutural

Propósito: usar compartilhamento para suportar eficientemente grandes quantidades de objetos de granularidade fina;

Aplicação: utilizar quando uma aplicação utiliza um grande número de objetos; os custos de armazenamento são altos por causa da grande quantidade de objetos; a maioria dos estados de objetos pode ser tornada extrínseca; muitos grupos de objetos podem ser substituídos por relativamente poucos objetos compartilhados, uma vez que estados extrínsecos são removidos; a aplicação não depende da identidade dos objetos; uma vez que objetos podem ser compartilhados, testes de identidade produzirão o valor verdadeiro para objetos conceitualmente distintos.



Proxy (Não exemplificado)

Tipo: Estrutural

Propósito: fornece um substituto ou marcador da localização de outro objeto para controlar o acesso a esse objeto;

Aplicação: é aplicável sempre que há necessidade de uma referência mais versátil, ou sofisticada, do que um simples apontador para um objeto; carregar um objeto persistente para a memória quando ele for referenciado pela primeira vez; verificar se o objeto real está bloqueado antes de ser acessado, para assegurar que nenhum outro objeto possa mudá-lo.

