



ARQUITETURA DE SOFTWARE

Padrões de projeto comportamentais

Geucimar Briatore
geucimar@up.edu.br

Atualizado em 10/2022

Padrões de projeto do grupo comportamental

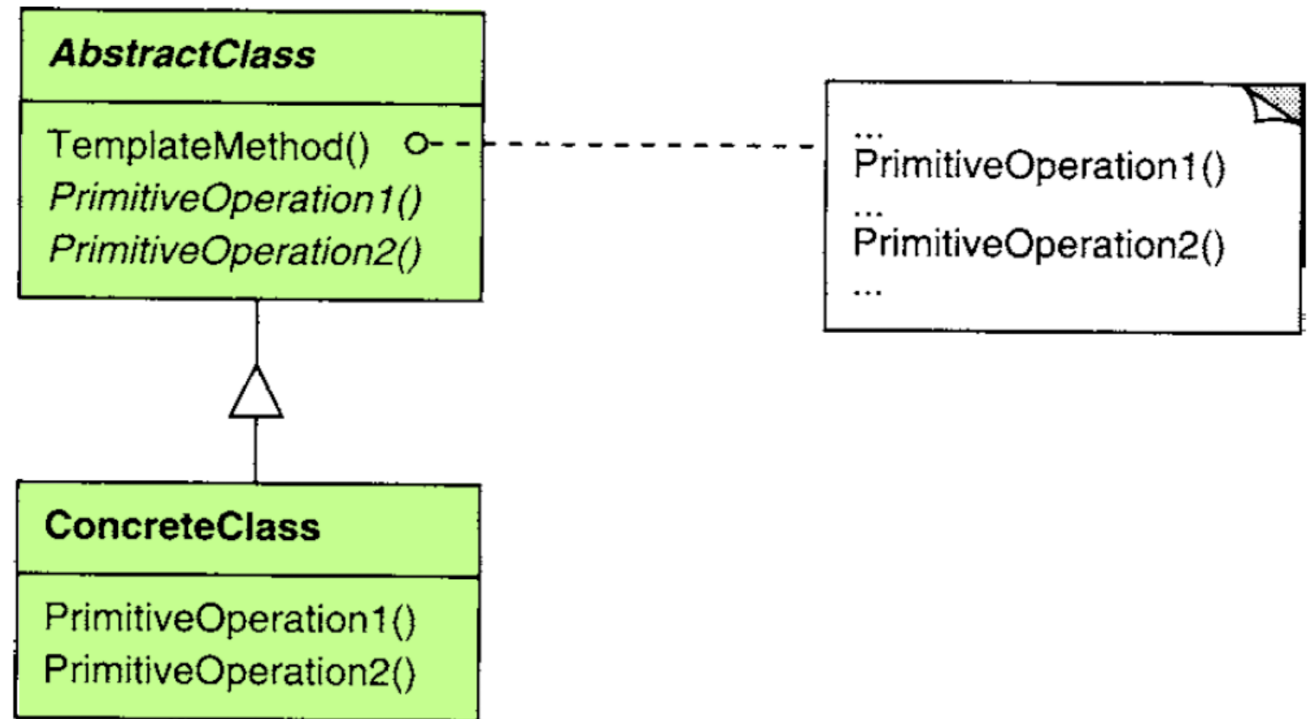
		Propósito		
		1. Criacional	2. Estrutural	3. Comportamental
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento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

Template Method

Tipo: Comportamental

Propósito: definir o esqueleto de um algoritmo em uma operação, postergando alguns passos para as subclasses. Permite que subclasses redefinam certos passos de um algoritmo sem mudar a estrutura do mesmo;

Aplicação: utilizar para implementar as partes invariantes de um algoritmo uma só vez e deixar para as subclasses a implementação do comportamento que pode variar; quando o comportamento comum entre subclasses deve ser fatorado e concentrado numa classe comum para evitar a duplicação de código; para controlar extensões de subclasses.

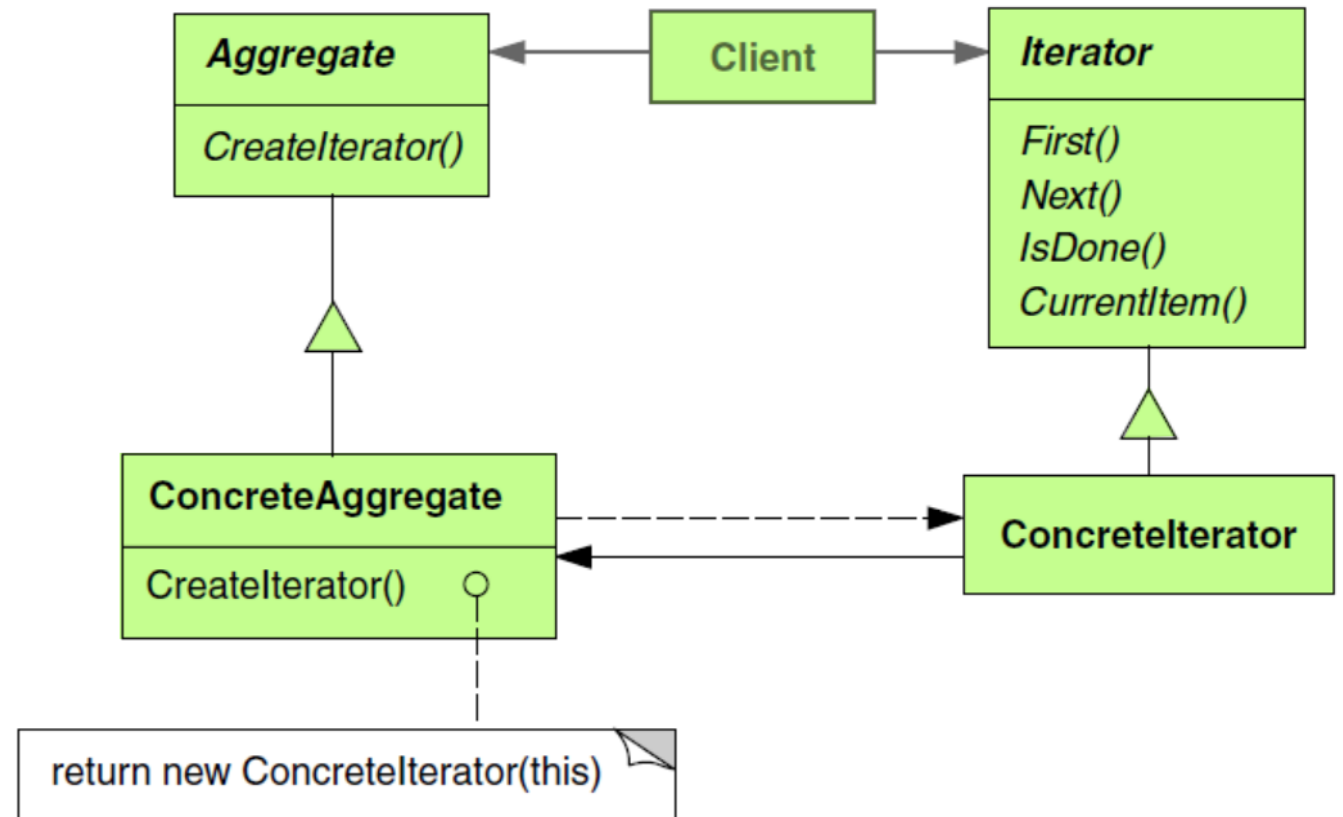


Iterator

Tipo: Comportamental

Propósito: fornecer um meio de acessar, seqüencialmente, os elementos de um objeto agregado sem expor a sua representação subjacente;

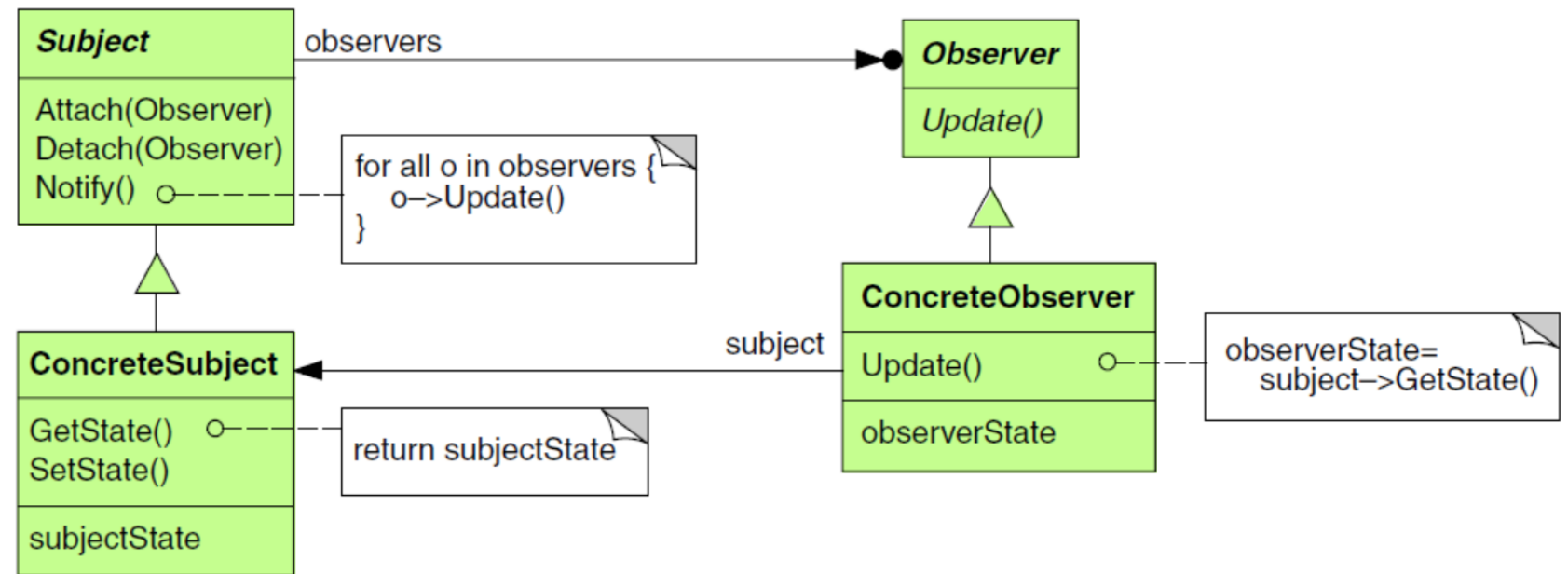
Aplicação: utilizar para acessar os conteúdos de um objeto agregado sem expor a sua representação interna; para suportar múltiplos percursos de objetos agregados; para fornecer uma interface uniforme que percorra diferentes estruturas agregadas (ou seja, para suportar a iteração polimórfica).



Observer

Tipo: Comportamental

Propósito: definir uma dependência um-para-muitos entre objetos, de maneira que quando um objeto muda de estado todos os seus dependentes são notificados e atualizados automaticamente;



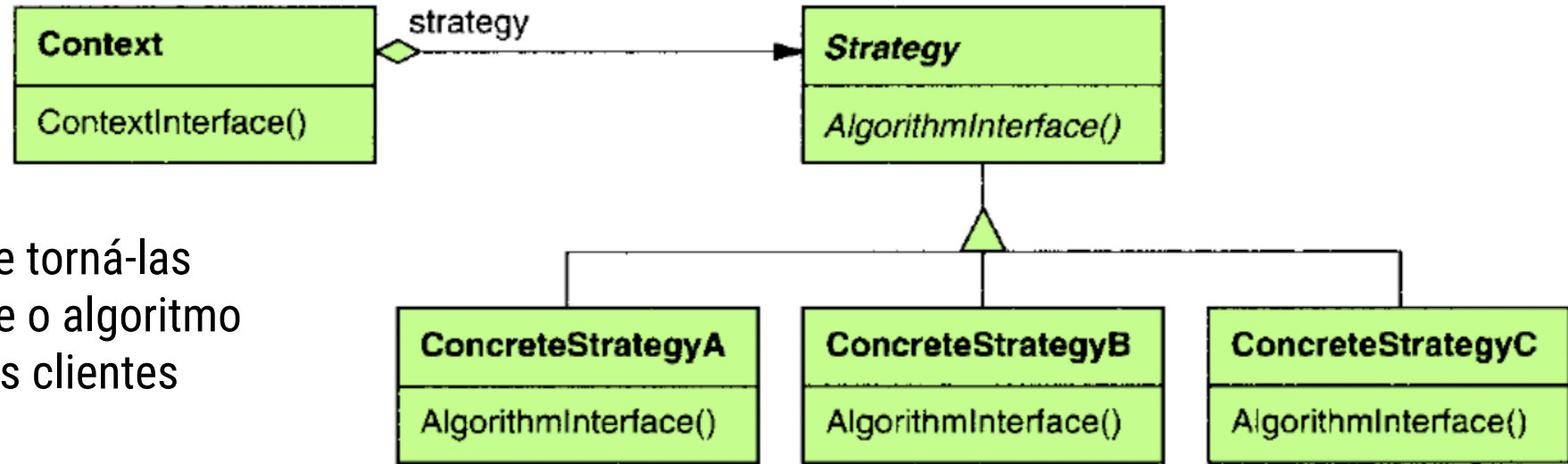
Aplicação: utilizar quando uma abstração tem dois aspectos, um dependente do outro; Encapsulando esses aspectos em objetos separados, permite-se variá-los e reutilizá-los independentemente; quando uma mudança em um objeto exige mudanças em outros, e você não sabe quantos objetos necessitam ser mudados; quando um objeto deveria ser capaz de notificar outros objetos sem fazer hipóteses, ou usar informações, sobre quem são esses objetos. Em outras palavras, você não quer que esses objetos sejam fortemente acoplados.

Strategy

Tipo: Comportamental

Propósito: definir uma família de algoritmos, encapsular cada uma delas e torná-las intercambiáveis. Permite que o algoritmo varie independentemente dos clientes que o utilizam;

Aplicação: utilizar quando muitas classes relacionadas diferem somente no seu comportamento; é necessário variantes de um algoritmo; podem ser usadas quando essas variantes são implementadas como uma hierarquia de classes de algoritmos; um algoritmo usa dados dos quais os clientes não deveriam ter conhecimento. Para evitar a exposição das estruturas de dados complexas, específicas do algoritmo; uma classe define muitos comportamentos, e estes aparecem em suas operações como múltiplos comandos condicionais da linguagem. Em vez de usar muitos comandos condicionais, mova os ramos condicionais relacionados para a sua própria classe Strategy.

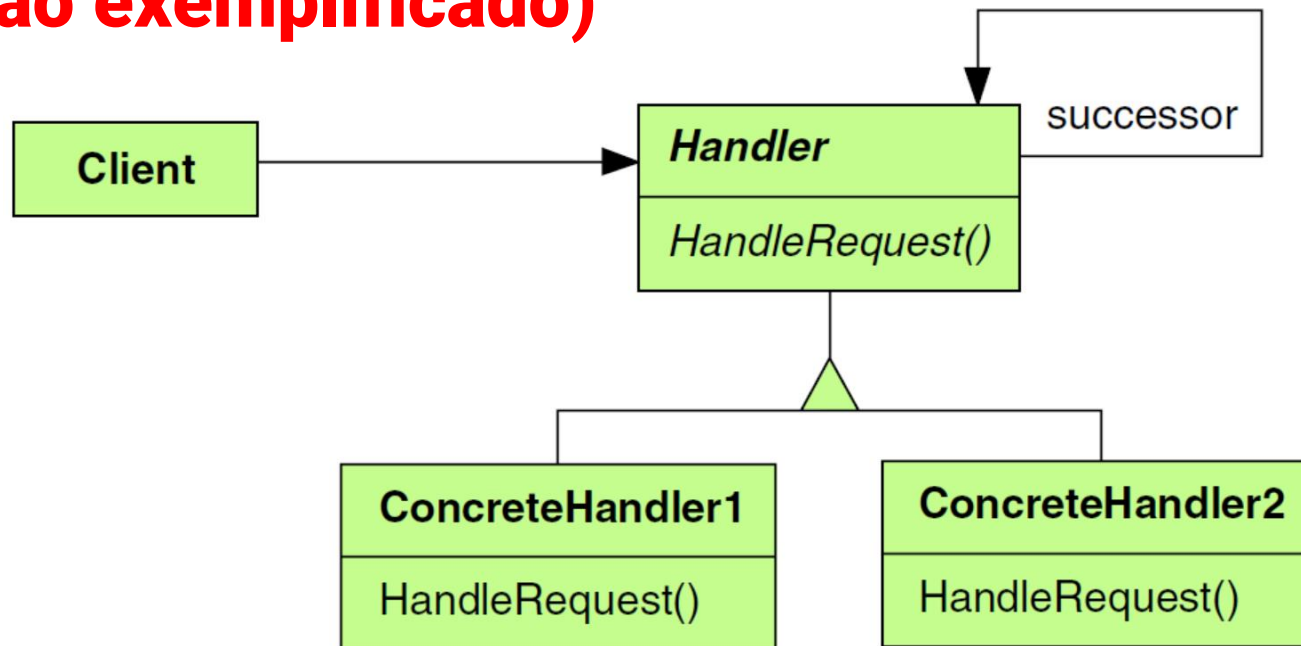


Chain Of Responsibility (Não exemplificado)

Tipo: Comportamental

Propósito: evitar o acoplamento do remetente de uma solicitação ao seu receptor, ao dar a mais de um objeto a oportunidade de tratar a solicitação. Encadear os objetos receptores, passando a solicitação ao longo da cadeia até que um objeto a trate;

Aplicação: utilizar quando mais de um objeto pode tratar uma solicitação e o objeto que a tratará não conhecido a priori; o objeto que trata a solicitação deve ser escolhido automaticamente; você quer emitir uma solicitação para um dentre vários objetos, sem especificar explicitamente o receptor; o conjunto de objetos que pode tratar uma solicitação deveria ser especificado dinamicamente.

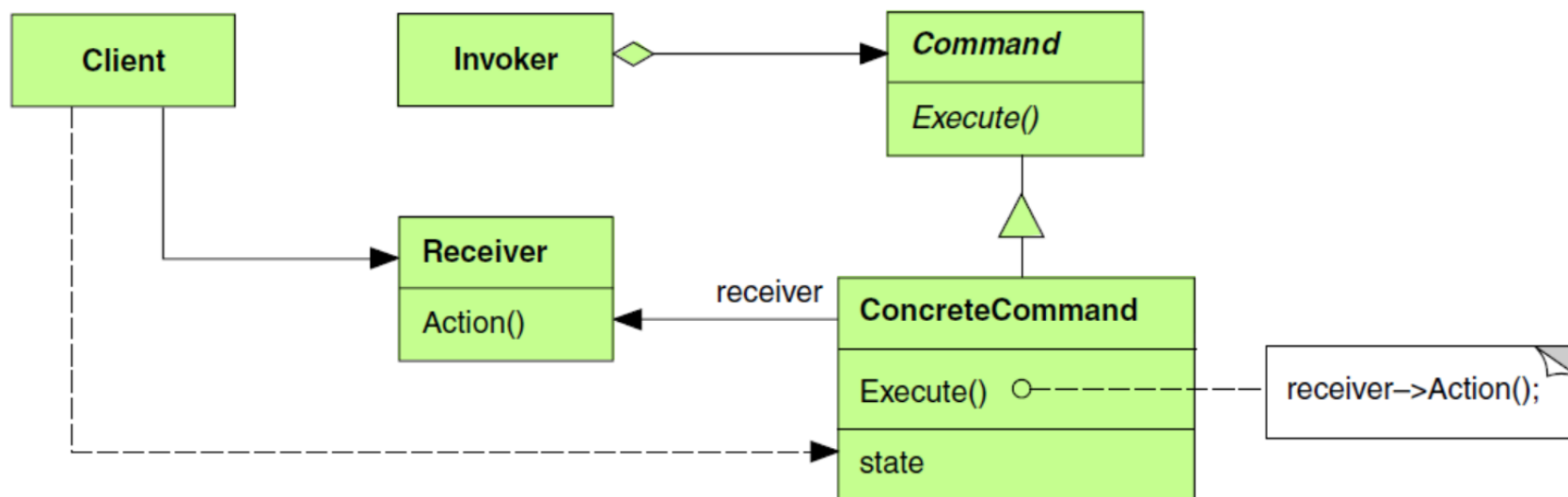


Command (Não exemplificado)

Tipo: Comportamental

Propósito: encapsular uma solicitação como um objeto, desta forma permitindo parametrizar clientes com diferentes solicitações, enfileirar ou fazer o registro (log) de solicitações e suportar operações que podem ser desfeitas;

Aplicação: utilizar para especificar, enfileirar e executar solicitações em tempos diferentes; commands, na orientação a objetos, substituem os callbacks da linguagem funcional, ou seja, uma função que é registrada em algum lugar para ser chamada em um momento mais tarde.

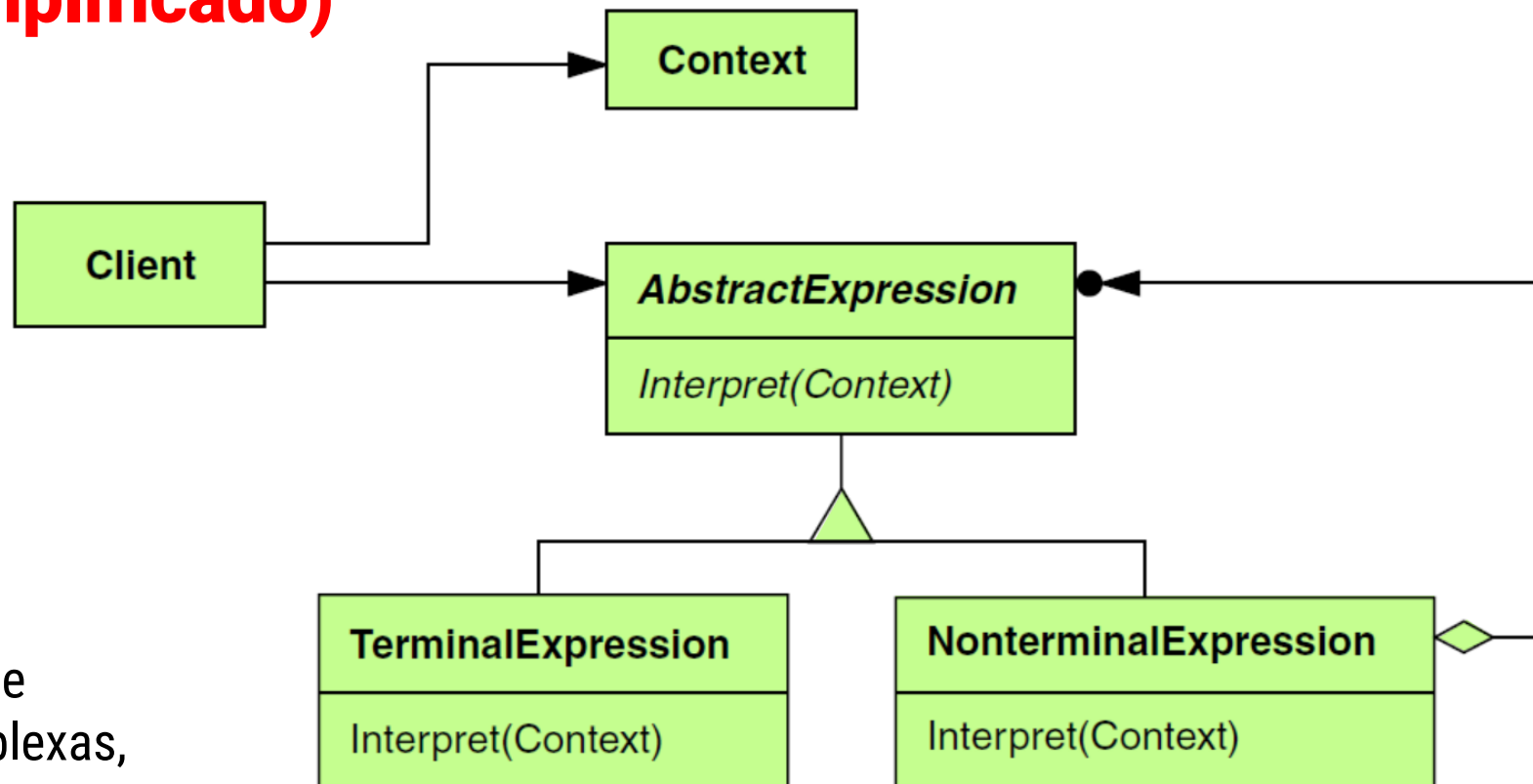


Interpreter (Não exemplificado)

Tipo: Comportamental

Propósito: dada uma linguagem, definir uma representação para a sua sintaxe juntamente com um interpretador que usa a representação para interpretar sentenças dessa linguagem;

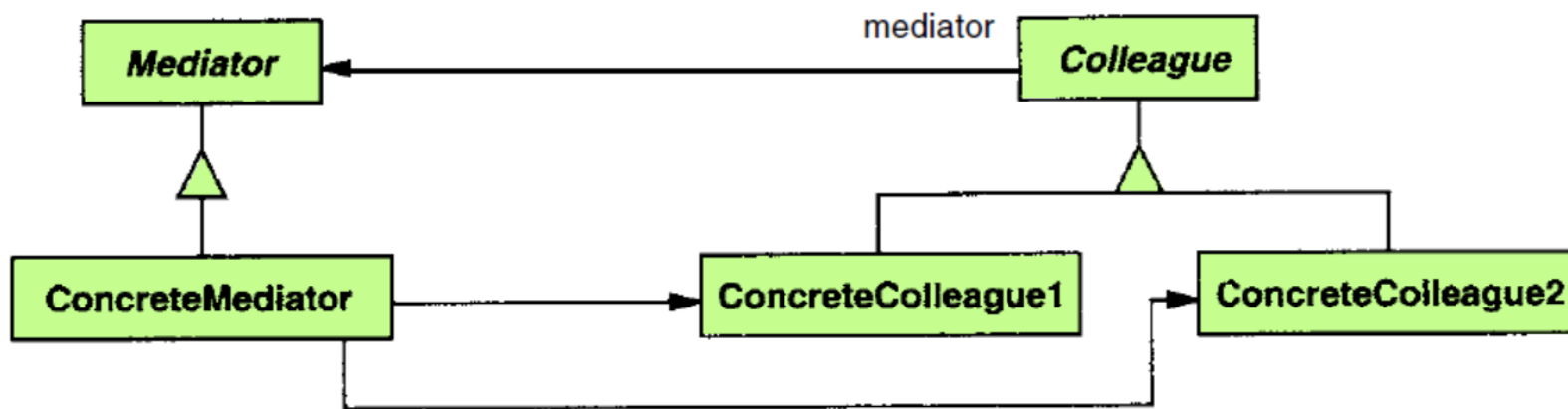
Aplicação: utilizar quando a sintaxe é simples. Para sintaxes são complexas, a hierarquia de classes para a sintaxe se torna grande e incontrolável. Em tais casos, ferramentas tais como geradores de analisadores são uma alternativa melhor. Elas podem interpretar expressões sem a construção de árvores sintáticas abstratas, o que pode economizar espaço e, possivelmente, tempo; a eficiência não é uma preocupação crítica. Os interpretadores mais eficientes normalmente não são implementados pela interpretação direta de árvores de análise sintática, mas pela sua tradução para uma outra forma.



Mediator (Não exemplificado)

Tipo: Comportamental

Propósito: definir um objeto que encapsula a forma como um conjunto de objetos interage. Promove o acoplamento fraco ao evitar que os objetos se refiram uns aos outros explicitamente e permite variar suas interações independentemente;



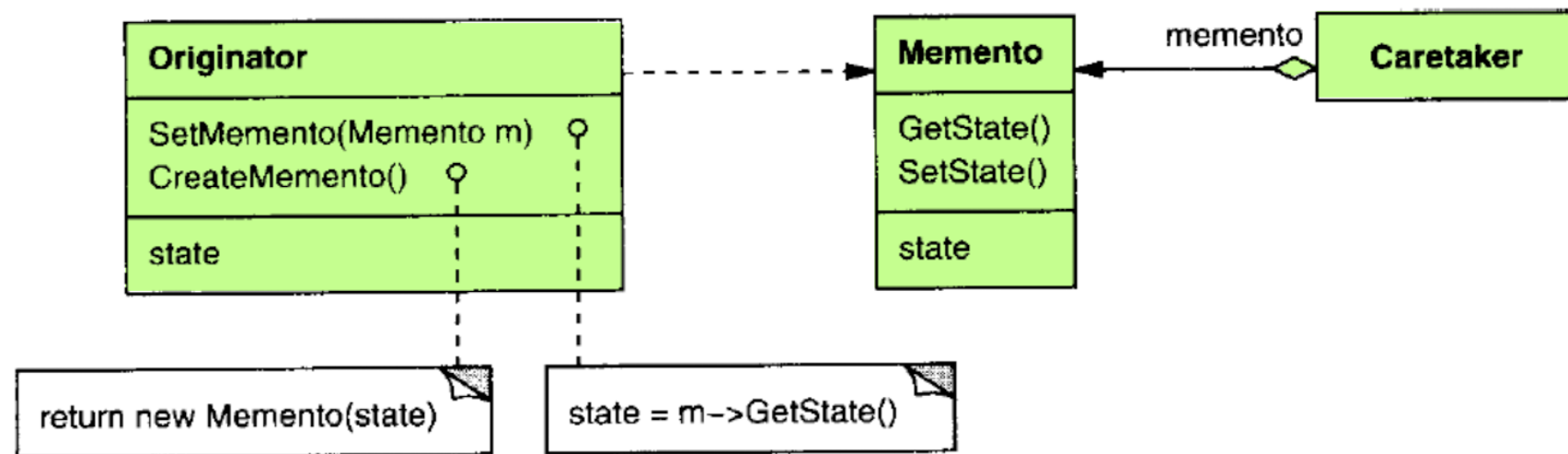
Aplicação: utilizar quando um conjunto de objetos se comunica de maneiras bem-definidas, porém complexas; as interdependências resultantes são desestruturadas e difíceis de entender; a reutilização de um objeto é difícil porque ele referencia e se comunica com muitos outros objetos; um comportamento que está distribuído entre várias classes deveria ser customizável, ou adaptável, sem excessiva especialização em subclasses.

Memento (Não exemplificado)

Tipo: Comportamental

Propósito: sem violar o encapsulamento, capturar e externalizar um estado interno de um objeto, de maneira que o objeto possa ser restaurado para esse estado mais tarde;

Aplicação: utilizar quando um instantâneo de (alguma porção do) estado de um objeto deve ser salvo de maneira que possa ser restaurado para esse estado mais tarde; uma interface direta para obtenção do estado exporia detalhes de implementação e romperia o encapsulamento do objeto.

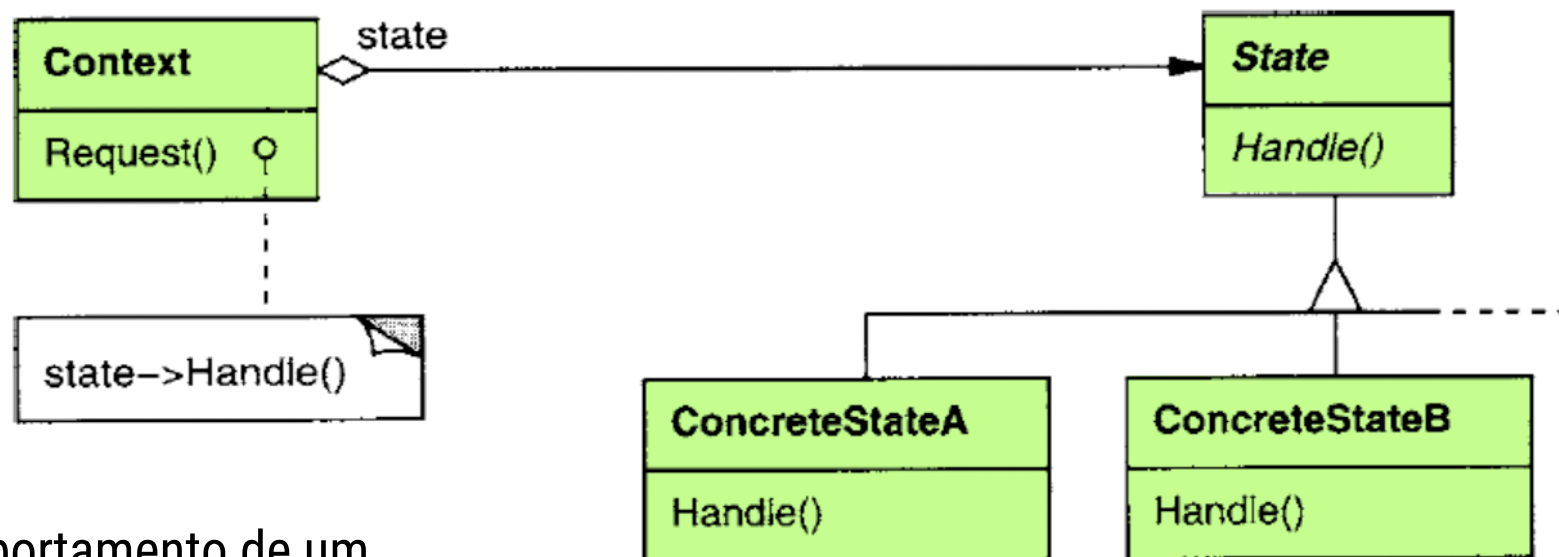


State (Não exemplificado)

Tipo: Comportamental

Propósito: permite a um objeto alterar seu comportamento quando o seu estado interno muda. O objeto parecerá ter mudado sua classe;

Aplicação: utilizar quando o comportamento de um objeto depende do seu estado e ele pode mudar seu comportamento em tempo de execução, dependendo desse estado; operações têm comandos condicionais grandes, de várias alternativas, que dependem do estado do objeto. Esse estado é normalmente representado por uma ou mais constantes enumeradas. Frequentemente, várias operações conterão essa mesma estrutura condicional. O padrão State coloca cada ramo do comando adicional em uma classe separada. Isto lhe permite tratar o estado do objeto como um objeto propriamente dito, que pode variar independentemente de outros obje.



Visitor (Não exemplificado)

Tipo: Comportamental

Propósito: representar uma operação a ser executada nos elementos de uma estrutura de objetos. Permite definir uma nova operação sem mudar as classes dos elementos sobre os quais opera;

Aplicação: utilizar quando uma estrutura de objetos contém muitas classes de objetos com interfaces que diferem e você deseja executar operações sobre esses objetos que dependem das suas classes concretas; muitas operações distintas e não-relacionadas necessitam ser executadas sobre objetos de uma estrutura de objetos; as classes que definem a estrutura do objeto raramente mudam; se as classes da estrutura do objeto mudam com frequência, provavelmente é melhor definir as operações nessas classes.

