Revisão sobre Design Patterns

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Conceitos

Certified Tech Developer
The Ultimate Degree

1 **CRIACIONAIS**

Fornecem vários mecanismos de criação de objetos, que aumentam a flexibilidade e reutilização de código já existente.

2 **ESTRUTURAIS**

Explicam como montar objetos e classes em estruturas maiores mas ainda mantendo essas estruturas flexíveis e eficientes.

3 **COMPORTAMENTAIS**

São voltados aos algoritmos e a designação de responsabilidades entre objetos.

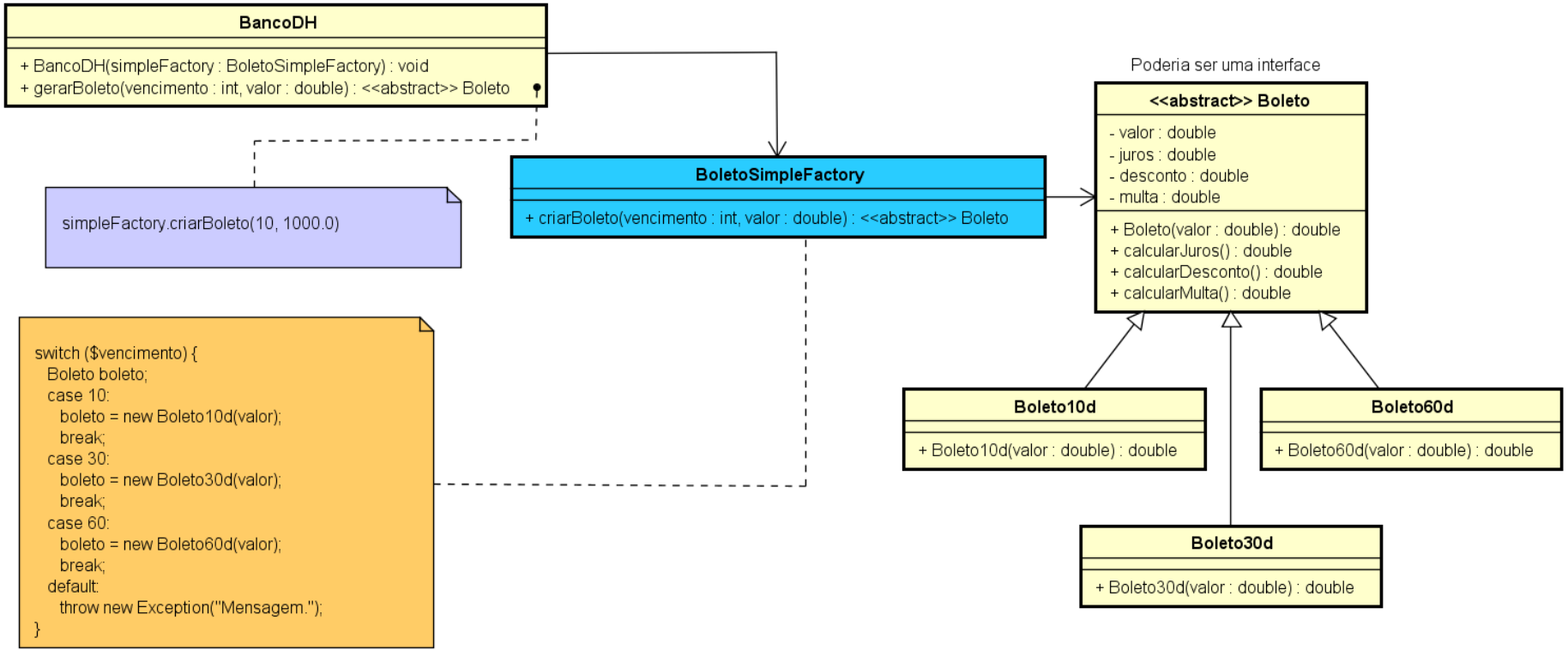
1 | Factory

Criacional

Certified Tech Developer
The Ultimate Degree

O **Factory Method** é um padrão criacional de projeto que fornece uma interface para criar objetos em uma superclasse, mas permite que as subclasses alterem o tipo de objetos que serão criados.

UML – Factory Method



Vantagens

Certified Tech Developer
The Ultimate Degree

- Você evita acoplamentos firmes entre o criador e os produtos concretos.
- *Princípio de responsabilidade única*. Você pode mover o código de criação do produto para um único local do programa, facilitando a manutenção do código.
- *Princípio aberto/fechado*. Você pode introduzir novos tipos de produtos no programa sem quebrar o código cliente existente.

Desvantagens

Certified Tech Developer
The Ultimate Degree

- O código pode se tornar mais complicado, pois você precisa introduzir muitas subclasses novas para implementar o padrão. O melhor cenário é quando você está introduzindo o padrão em uma hierarquia existente de classes criadoras.

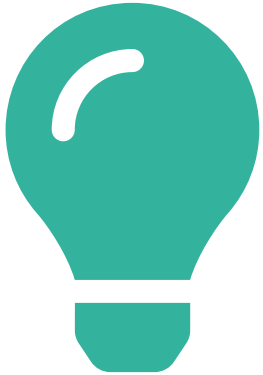
***Fonte**: https://refactoring.guru/pt-br/design-patterns/factory-method*

2 | Singleton

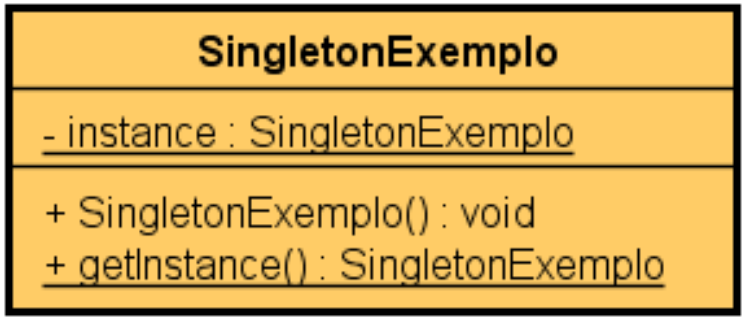
Criacional

Certified Tech Developer
The Ultimate Degree

O **Singleton** é um padrão criacional que garante que uma classe possua uma única instância e define um ponto de acesso global para ela.



UML – Singleton



Vantagens

Certified Tech Developer
The Ultimate Degree

- Você pode ter certeza que uma classe só terá uma única instância.
- Você ganha um ponto de acesso global para aquela instância.
- O objeto singleton é inicializado somente quando for pedido pela primeira vez.

Desvantagens

Certified Tech Developer
The Ultimate Degree

- O padrão Singleton pode mascarar um design ruim, por exemplo, quando os componentes do programa sabem muito sobre cada um.
- O padrão requer tratamento especial em um ambiente multithreaded para que múltiplas threads não possam criar um objeto singleton várias vezes.

***Fonte**: https://refactoring.guru/pt-br/design-patterns/singleton*

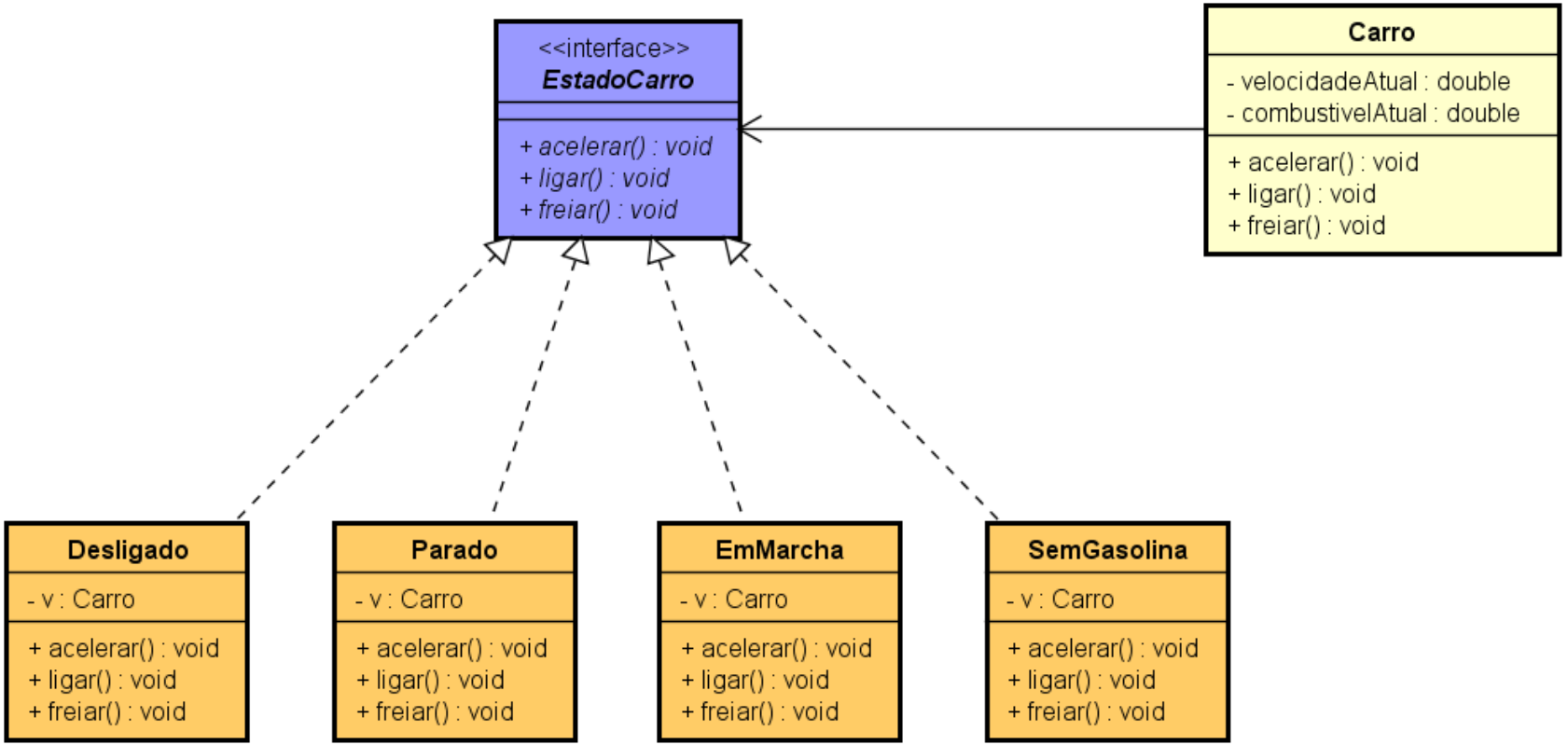
3 State

Comportamental

Certified Tech Developer
The Ultimate Degree

O padrão de projeto **State** permite que um objeto altere o seu comportamento quando o seu estado interno muda. O objeto parecerá ter mudado de classe.

UML – State


<<interface>>
EstadoCarro
+ acelerar() : void
+ ligar() : void
+ freiar() : void
Carro
- velocidadeAtual : double
- combustivelAtual : double
+ acelerar() : void
+ ligar() : void
+ freiar() : void
Desligado
- v : Carro
+ acelerar() : void
+ ligar() : void
+ freiar() : void
Parado
- v : Carro
+ acelerar() : void
+ ligar() : void
+ freiar() : void
EmMarcha
- v : Carro
+ acelerar() : void
+ ligar() : void
+ freiar() : void
SemGasolina
- v : Carro
+ acelerar() : void
+ ligar() : void
+ freiar() : void

Vantagens

Certified Tech Developer
The Ultimate Degree

- *Princípio de responsabilidade única*. Organiza o código relacionado a estados particulares em classes separadas.
- *Princípio aberto/fechado*. Introduz novos estados sem mudar classes de estado ou contexto existentes.
- Simplifica o código de contexto ao eliminar condicionais de máquinas de estado pesadas.

Desvantagens

Certified Tech Developer
The Ultimate Degree

- Aplicar o padrão pode ser um exagero se a máquina de estado só tem alguns estados ou raramente muda eles.

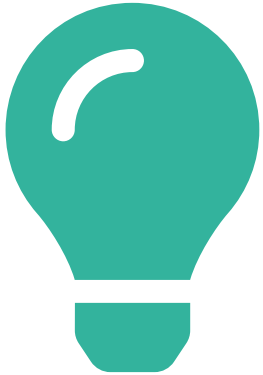
***Fonte**: https://refactoring.guru/pt-br/design-patterns/state*

4 Composite

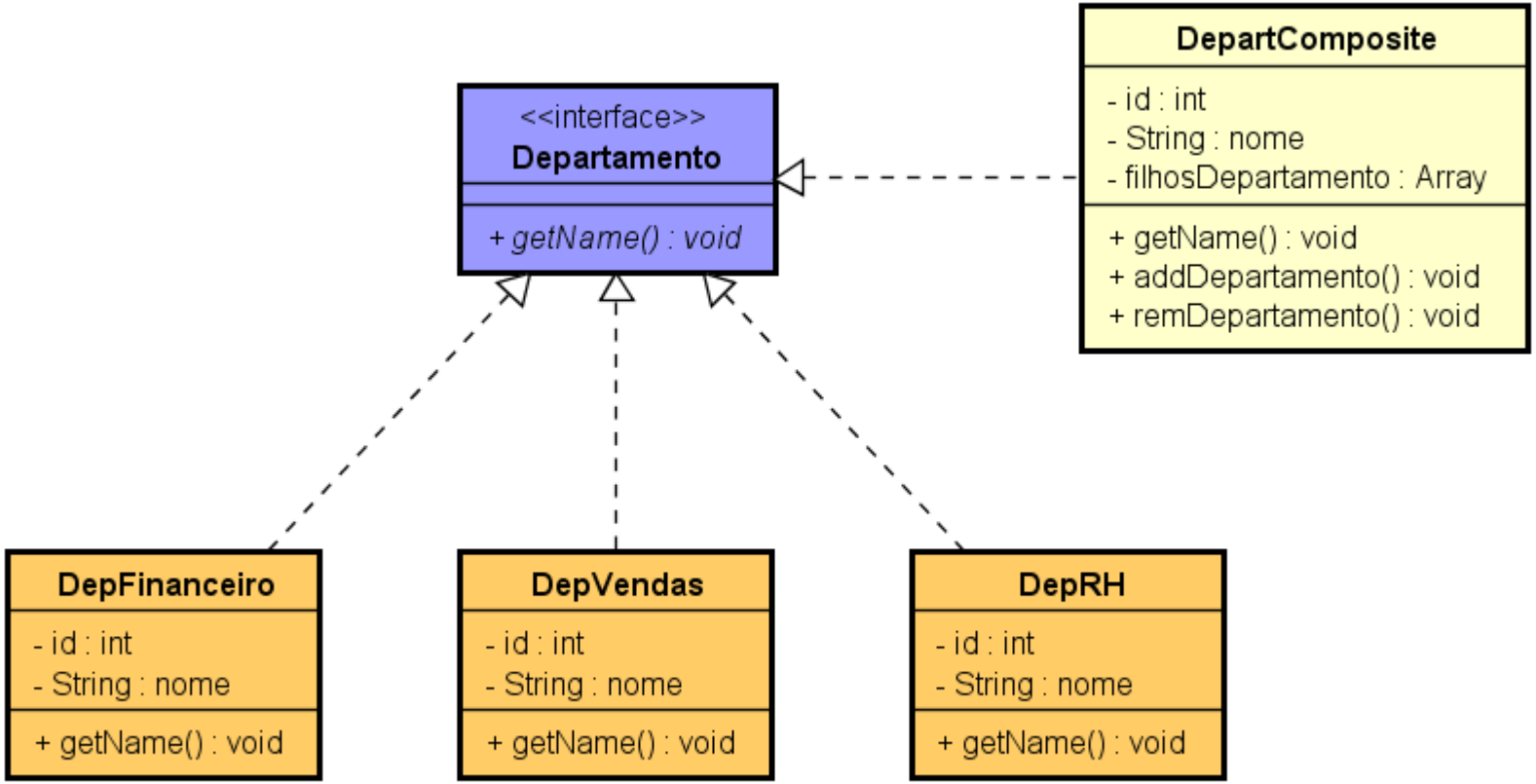
Estrutural

Certified Tech Developer
The Ultimate Degree

O padrão **composite** permite a composição de objetos em estruturas de árvore para representar hierarquias parte-todo. Com esse padrão, os clientes podem tratar objetos individuais ou composições de objetos de maneira transparente e uniforme.



UML – Composite



Vantagens

Certified Tech Developer
The Ultimate Degree

- Você pode trabalhar com estruturas de árvore complexas mais convenientemente: utilize o polimorfismo e a recursão a seu favor.
- *Princípio aberto/fechado*. Você pode introduzir novos tipos de elemento na aplicação sem quebrar o código existente, o que agora funciona com a árvore de objetos.

Desvantagens

Certified Tech Developer
The Ultimate Degree

- Pode ser difícil providenciar uma interface comum para classes cuja funcionalidade difere muito. Em certos cenários, você precisaria generalizar muito a interface componente, fazendo dela uma interface de difícil compreensão.

Fonte: https://refactoring.guru/pt-br/design-patterns/composite

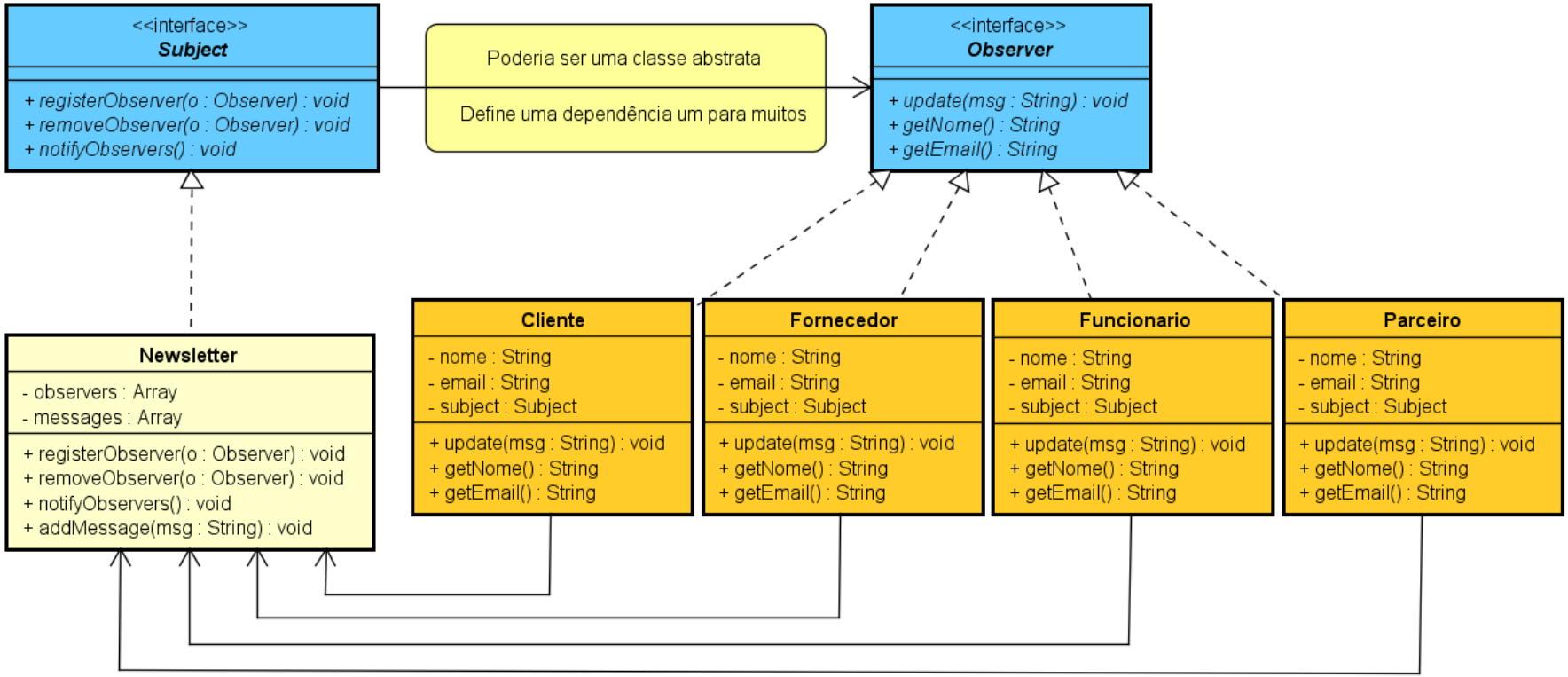
5 Observer

Comportamental

Certified Tech Developer
The Ultimate Degree

O **Observer** é um padrão de projeto de software que define uma dependência um-para-muitos entre objetos, de modo que quando um objeto muda seu estado, todos seus dependentes são notificados e atualizados automaticamente.

UML – Observer



Vantagens

Certified Tech Developer
The Ultimate Degree

- *Princípio aberto/fechado*. Você pode introduzir novas classes assinantes sem ter que mudar o código da publicadora (e vice versa se existe uma interface publicadora).
- Você pode estabelecer relações entre objetos durante a execução.

Desvantagens

Certified Tech Developer
The Ultimate Degree

- Assinantes são notificados em ordem aleatória.

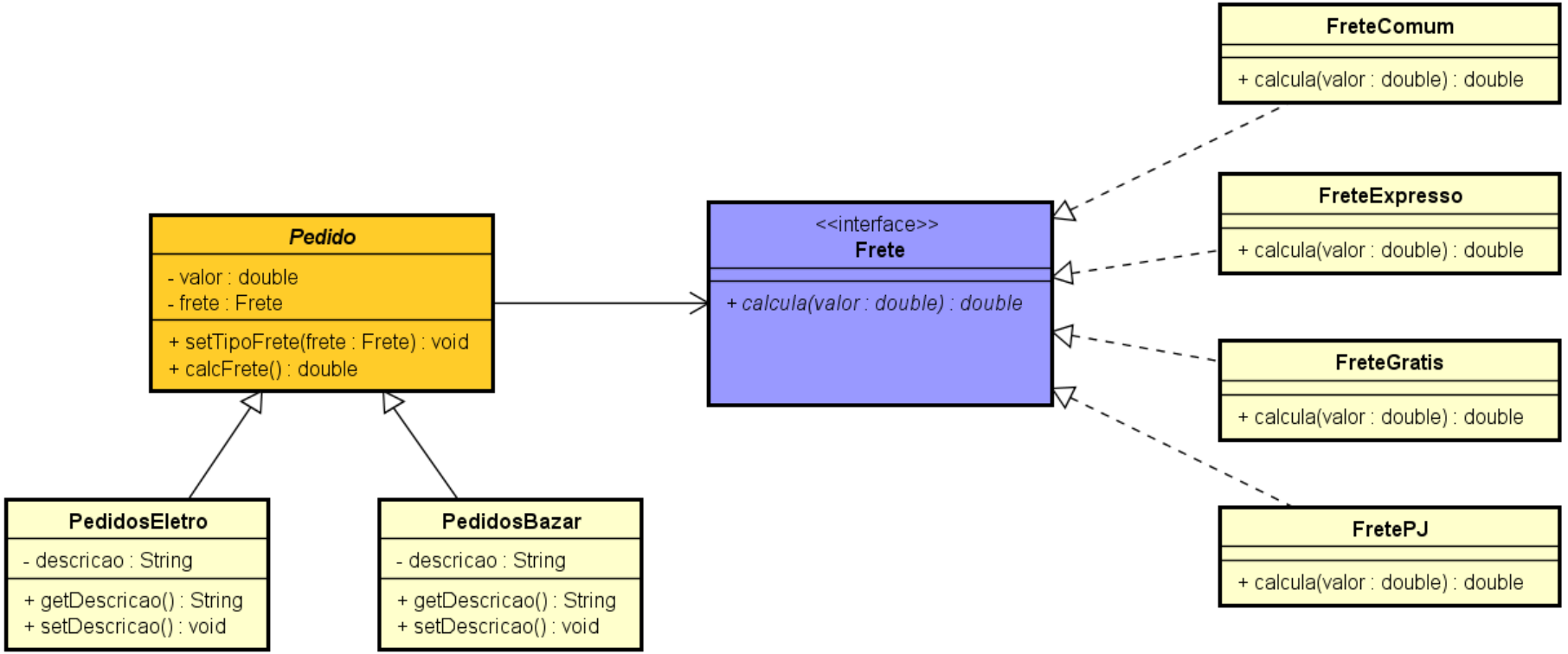
Fonte: https://refactoring.guru/pt-br/design-patterns/observer

6 Strategy
Comportamental

Certified Tech Developer
The Ultimate Degree

O padrão de projeto **Strategy** define uma família de algoritmos, encapsula cada um deles e os torna intercambiáveis. Permite que o algoritmo varie independentemente dos clientes que o utilizam.

UML – Strategy



Vantagens

Certified Tech Developer
The Ultimate Degree

- Você pode trocar algoritmos usados dentro de um objeto durante a execução.
- Você pode isolar os detalhes de implementação de um algoritmo do código que usa ele.
- Você pode substituir a herança por composição.
- *Princípio aberto/fechado*. Você pode introduzir novas estratégias sem mudar o contexto.

Desvantagens

Certified Tech Developer
The Ultimate Degree

- Os Clientes devem estar cientes das diferenças entre as estratégias para serem capazes de selecionar a adequada.
- Muitas linguagens de programação modernas tem suporte do tipo funcional que permite que você implemente diferentes versões de um algoritmo dentro de um conjunto de funções anônimas.

***Fonte**: https://refactoring.guru/pt-br/design-patterns/strategy*

Certified Tech Developer
The Ultimate Degree

Lembre-se! Este material é acessório, não deixe de consumir o seu PG.

DigitalHouse>
Coding School