

# Trabalho Prático sobre Princípios SOLID e Padrões de Projeto

## PUC Minas - Engenharia de Software - Programação Modular

Luca Ferrari Azalim

---

### 1. Artefatos do Projeto

Os artefatos do projeto estão listados abaixo:

Artefato	Caminho
Arquivo README.md	README.md
Diagrama UML	/docs/uml-diagram.png
Javadoc	/docs/javadoc/index.html
Código Fonte	/src/main/java
Testes Unitários	/src/test/java
Repositório do GitHub	<a href="https://github.com/lucaazalim/stock-market-simulation">github.com/lucaazalim/stock-market-simulation</a>

### 2. Descrição do Sistema

Com base no enunciado fornecido, desenvolvi uma aplicação que simula a operação de uma bolsa de valores, inspirada na B3. Tomei a liberdade de abordar mais conceitos do que o solicitado pelo enunciado, como, por exemplo, a divisão das negociações entre mercado comum e mercado fracionário e a identificação dos ativos como ordinários, preferenciais ou units.

Quando executada, a aplicação inicia a simulação e realiza as seguintes tarefas:

1. Vincula um observador de transações (*TransactionObserver*) à bolsa (*StockMarket*), para reportar quando novas transações (*Transaction*) são realizadas;
2. Faz com que todas as corretoras (*Broker*) observem um conjunto aleatório de ativos (*Asset*) e reporta no console quando novas ofertas (*OfferOperation*) são registradas;
3. Inicia uma thread para cada corretora, que registra operações aleatórias de compra, venda (*OfferOperation*) e informação (*InfoOperation*) na bolsa. Isso ocorre com uma frequência aleatória e é reportado no console.

### 3. Padrões de Projeto

Os seguintes padrões de projeto foram utilizados:

Padrão de Projeto	Onde foi utilizado
Decorator	Classe <b>Asset</b> , por meio do atributo <b>parentAsset</b> .
Observer	Classes <b>Observable</b> , <b>Observer</b> e suas implementações.
Singleton	Classe <b>StockMarket</b> .
Strategy	Método <b>process(...)</b> da classe <b>Operation</b> , que é sobrescrito pelas classes <b>InfoOperation</b> e <b>OfferOperation</b> .
Factory	Classe <b>OperationFactory</b> .

### 4. Descrição das Classes

Abaixo, estão descritas todas as classes da aplicação:

Classe	Descrição
<b>Pacote principal</b>	
App	Classe principal da aplicação, responsável por instanciar o <b>StockMarket</b> e iniciar a simulação por meio da classe <b>Simulation</b> .
Simulation	Responsável por executar a simulação, que inclui, por exemplo, o envio de operações aleatórias com frequência aleatória.
StockMarket	Representa a bolsa de valores e é responsável por armazenar os livros de operação e as corretoras.
Utils	Classe utilitária estática que inclui, por enquanto, apenas um método para identificar a thread atual está sendo executada pelo Junit ou não.
<b>Pacote "asset"</b>	
Asset	Representa um ativo da bolsa de valores, como PETR4, PETR4F ou XPLG11.
AssetType	Representa o tipo de um ativo da bolsa de valores, podendo ser ordinário, preferencial ou unit.
MarketType	Representa o tipo de mercado em que um ativo da bolsa de valores é negociado, podendo ser comum (COMMON) ou fracionário (FRACTIONAL).
<b>Pacote "broker"</b>	

B3Broker	Enumeração de corretoras que negociam ativos na B3.
Broker	Interface que representa uma corretora que negocia ativos na bolsa de valores.
<b>Pacote "company"</b>	
B3Company	Enumeração de empresas de capital aberto que negociam seus ativos na B3.
Company	Interface que representa uma empresa de capital aberto que negocia seus ativos na bolsa de valores.
<b>Pacote "observer"</b>	
Observable	Interface que representa um objeto que pode ser acompanhado (observado) por objetos do tipo Observer.
Observer	Interface que representa um objeto observador, que pode acompanhar (observar) objetos do tipo Observable.
OperationBookObserver	Interface que representa um observador de livros de operação. Ele é notificado quando novas operações do tipo oferta são registradas na bolsa de valores.
TransactionObserver	Interface que representa um observador de transações.
<b>Pacote "operation"</b>	
Operation	Representa uma operação que pode ser registrada em um livro de operações da bolsa de valores. Pode ser de <b>venda</b> , <b>compra</b> ou de <b>informação</b> .
OperationBook	Representa um livro de operações de um ativo da bolsa de valores e pode conter operações de todos os tipos.
OperationFactory	Representa uma fábrica de operações.
InfoOperation	Representa uma operação do tipo <b>informação</b> , que é registrada por uma corretora quando esta deseja saber o preço de um ativo em um ponto específico do tempo.
OfferOperation	Representa uma operação do tipo <b>oferta</b> , que pode ser tanto de compra quanto de venda.
OfferOperationStatus	Representa o status de uma operação de oferta, podendo ser <b>aberta</b> , <b>executada parcialmente</b> ou <b>executada</b> .
OfferOperationType	Representa o tipo de uma operação de oferta, podendo ser <b>compra</b> ou <b>venda</b> .
<b>Pacote "wallet"</b>	
Transaction	Representa uma transação presente na carteira de uma corretora.

Wallet	Representa a carteira de uma corretora, contendo transações realizadas por ela.
--------	---