

Comunicação Indireta | Estrutura do Projeto

Marlon Henrique da Silva

Pastas e Arquivos

- **bin:** contém os arquivos binários (.class) da aplicação.
- **lib:** contém as dependências externas (.jar) da aplicação.
- **src:** contém os pacotes, classes (.java) e todo o código da aplicação.
- **acoes_bovespa:** é uma lista (.csv) de ações da bovespa contendo o código, nome de pregão e atividade principal de cada ativo.
- **Broker:** é um arquivo compilado (.jar) que abre e executa a interface gráfica para corretoras.
- **StockExchange:** é um arquivo compilado (.jar) que abre e executa a interface gráfica da bolsa de valores.
- **README:** é um arquivo de texto (.txt) com as instruções de execução e uso da aplicação.

Pacotes

- **broker:** possui todas as classes referentes às corretoras (brokers);
- **stock_exchange:** possui todas as classes referentes à bolsa de valores (stock exchange);
- **storage:** possui todas as classes de arquivo e armazenamento em memória do sistema.

Classes

- **broker:**
 - **BrokerEmit:** é uma thread responsável pela publicação de mensagens no canal "BROKER".
 - **BrokerReceive:** é uma thread responsável por receber as mensagens do canal "BOLSADEVALORES" com base em um tópico específico.
 - **BrokerGUI:** é uma interface gráfica para brokers que possibilita a negociação e o acompanhamento de ativos (tópicos).
- **stock_exchange:**
 - **StockEmit:** é uma thread responsável pela publicação de mensagens no canal "BOLSADEVALORES".

- **StockReceive:** é uma thread responsável por receber todas as mensagens do canal “BROKER”.
- **StockGUI:** é uma interface gráfica que permite iniciar e acompanhar as negociações na bolsa.
- **storage:**
 - **AssetList:** é uma classe utilitária que realiza a leitura do arquivo com as ações da bovespa.
 - **OfferBook:** é uma classe utilitária que armazena as ofertas e realiza correspondências de compra-venda.
 - **Transaction:** é uma classe utilitária que gera e armazena as transações.

Operações

- **BrokerReceive.run():** gera um JLabel para cada mensagem recebida (com base em um tópico) e o insere em um JPanel (actionsP) no BrokerGUI para ser exibido na interface.
- **StockReceive.run():** gera um JLabel para cada mensagem recebida (de todos os tópicos) e o insere em um JPanel (actionsP) no StockGUI para ser exibido na interface, além de emitir a mesma mensagem para a “BOLSADEVALORES” e armazená-la em OfferBook.
- **BrokerGUI.viewer():** gera e exibe o visualizador de negociações dos ativos que são acompanhados pelo broker.
- **AssetList.load():** realiza a leitura do arquivo “acoes_bovespa.csv” e retorna um Vector<String> com os códigos de todos os ativos.
- **OfferBook.store():** recebe uma oferta e a salva em sua respectiva lista (compra ou venda).
- **OfferBook.matchOffers():** recebe a última oferta salva e busca em uma das listas uma oferta correspondente para gerar uma transação.
- **Transaction.store():** recebe duas ofertas (compra e venda) correspondentes e gera a transação entre elas; armazena e emite a transação.

Diagrama de Classes

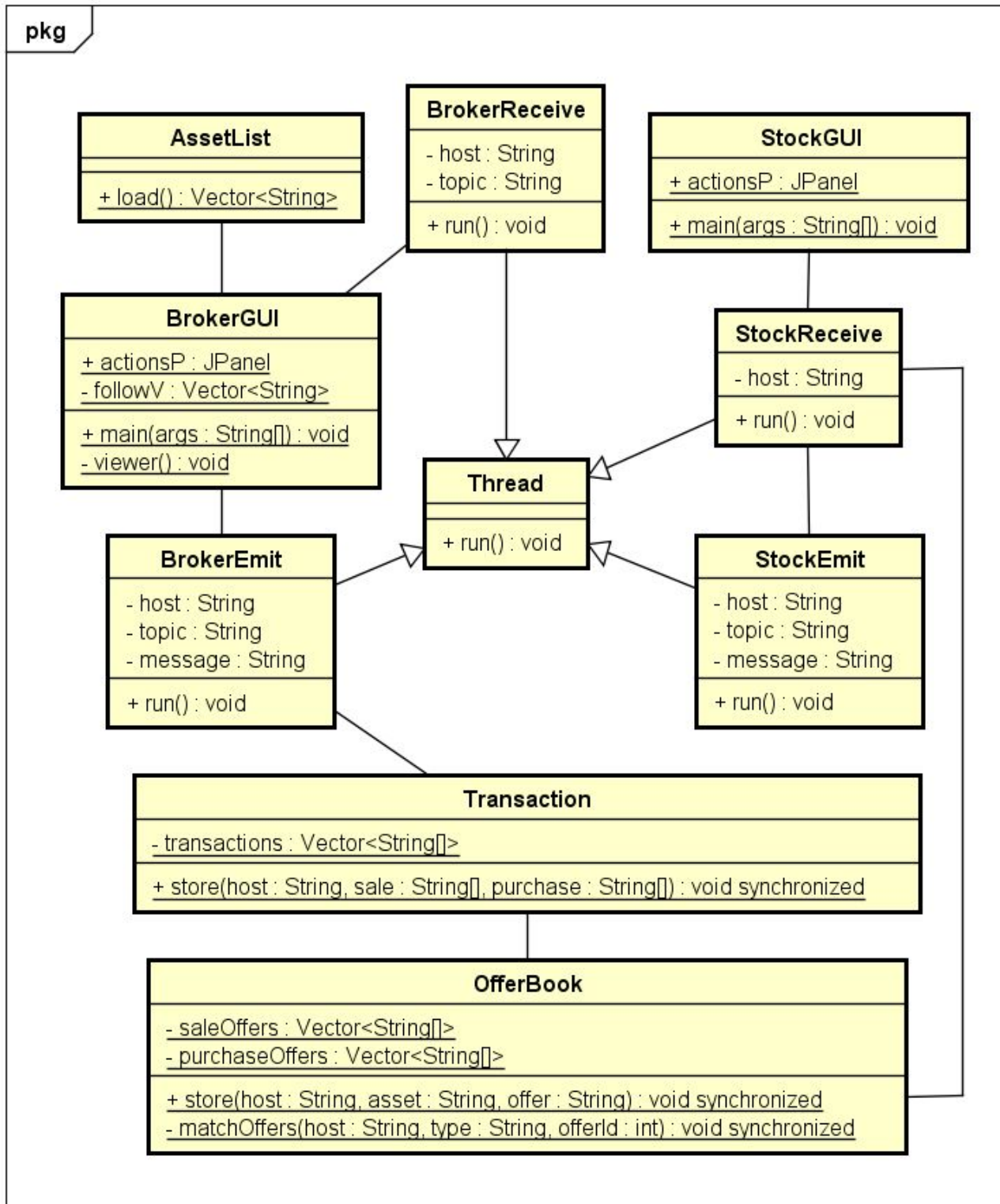


Diagrama de Componentes

