

Tutorial 8: Coding the Stock Trader

Task

Implement the Stock Trader program using the client-server architecture. Create two classes: `StockClient` and `StockServer` and other classes to represent other entities in the program as necessary (such as `User`, `Stock`, `StockInformation`).

- Create a package named `stocktrader`, then two sub-packages `stocktrader.server` and `stocktrader.client` to separate server codes from client codes.
- Do not expose server classes and objects to the client.
 - You cannot import classes from the `server` sub-package in any of the classes of the `client` sub-package, except for the class `StockServer`.
 - The communication between server and client should be done using `String` and primitive data types only.
- In the class `stocktrader.server.StockServer`, provide these methods:
 - `public boolean login(String username, String password)`
To make it simple, this method verify the provided username and password against an list of users stored on the server. On server, there is a boolean attribute indicating whether the user has logged in or not. If it is `false`, other operations on the server will not work (e.g. throw an `AccessDeniedException` or something). When the user successfully logs in, set this attribute to `true`.
 - `public String listAllStocks()`
Show a list of all stocks with a stock no. (1, 2, 3) associated to each stock. Client can use the stock no. in the `purchase()` method.
 - `public boolean purchase(int stockNo, int quantity)`
This method uses the stock no. listed in the `listAllStocks()` method.
 - `public String listOwnStocks()`
Show a list of all stocks owned by the user with a stock no. (1, 2, 3) associated to each stock. Client can use the stock no. in the `sellStock()` method.
 - `public boolean sellStock(int stockNo, int quantity)`
This method uses the stock no. listed in the `listOwnedStocks()` method.
 - `public void nextDay()`
Randomly change all stock prices as if a day has passed.
 - `public Double checkBalance()`
Show the user's current account balance (the sum of all the multiple of price and stock, then added by user money).
 - `public String getPrice()`
return the current price of all stocks to user.
 - `private ArrayList<Stock> getCurrentPrice()`
Load the current price of all stocks from file text.
 - `private boolean storeInformaton(StockInformation info)`

Append information to file text.

- The class `stocktrader.client.StockClient` is a console-based, menu-driven program which allows the user to perform all the features provided by the server.