note: for the exercise below, please create a Github repository (private or public) where to store all artifacts related to the resolution of the exercise (code, documents, readme files, etc). Having a descriptive git commit history will be useful for us to understand how you solved the exercise.

Code Quality exercise

Considering the code in `src/smelly_code/trade_execution_service.rb:`

1. How would you improve the code quality (testability, readability, maintainability...) of the previous Service object? Please enumerate any code smells or bad practices that are present in the code above.

2. Refactor the code in order to amend the issues described above. Feel free to decompose current `TradeExecutionService` into a set of smaller objects or services if needed. Push the new implementation into the `src/clean_code/` folder.

3. Write tests (in `Rspec`) to verify that the current code implementation satisfies the following order routing rules:

Order amount	routed to
`Less than 10k USD`	`LIQUIDITY_PROVIDER_C`
`Equal or bigger than 10k USD but less than 100k USD`	`LIQUIDITY_PROVIDER_B`
`Bigger than 100k USD`	`LIQUIDITY_PROVIDER_A`

Bonus:

4. Make sure all ruby code compiles and tests pass. Add `require` statements when needed, use a `Gemfile` to specify versions and `Bundler` to install any required libraries (gems).

5. Provide a Dockerfile so that tests can run in a docker container. Provide instructions on how to run the container.