

Bank Account Handler Module

Summary

This solution is written as a module. So, the output is basically a jar file. Instead sufficient test cases are added to test functionalities. It is a maven project, which can be run using command prompt from inside the project directory provided that maven is installed in local computer. **IBankAccountHandler** interface is implemented to provide requested features for that solution. JUnit testing framework is used to write unit tests. Apache Log4j framework is used to write logs in log file.

Technical information

All commands should be run from inside the project directory. **Change the directory (value of log) in log4j.properties to your convenient directory before running maven command. Otherwise it will show error.** Make sure that your user has the privilege to create file in that directory.

Following are some maven commands to operate this module.

- **maven-compiler-plugin** is used to compile the sources of your project.
- Make jar file using command: **mvn clean package**. The file will be generated in target folder named **bankaccount-0.0.1-SNAPSHOT.jar**
- Run tests using command: **mvn test**
- **maven-javadoc-plugin** is used to generate javadoc for the module. Run command: **mvn javadoc:javadoc** that will generate java documentation inside the directory: `../target/site/apidocs`. Classes and methods are commented so that user can read the descriptions from both java files and java documentations.

Bullet points' solution description

Following is the short description that says how the requirements given in the bullet points are solved.

Keep data in memory (no database or other storage)

Mainly **DataStore** class is responsible to keep data in memory. No database is used.

Add support for retrieving account information

The method **getAccount** inside **DataStore** class is used to retrieve account information. The account id is needed to pass as a parameter to get the information of that account.

Add support for retrieving account history - 10 latest transactions shall be available

Please check **getLatestTransactions** method inside **DataStore** class. Unit test is added for that. You can also check **bankaccount.log** file to see whether 10 latest transactions is saved or not. Example file is given in the project directory for your convenience.

Add support for retrieving accounts, in account balance order, for those accounts with balance exceeding specified balance limit

Please see `getAccountsExceedingBalance` method inside `DataStore` class. Also see test method `testGetAccountsExceedingBalance` in `DataStoreTest` class

Negative account balance is allowed

The method `getAccountBalance` inside `DataStore` class supports negative balance. Test case is added for this purpose. Please see `testGetBalance` in `DataStoreTest` class.

Add support for registering `IBankAccountEventListener`'s that will be notified when an account balance becomes negative, calls to registered listeners should be done in separate thread

Implemented the interface `IBankAccountEventListener` where the method used to notify the users if the balance becomes negative after any transaction. Added `addEventListener` method for this purpose. When an account balance becomes negative, calls to registered listeners should be done in separate thread. `BankAccountHandlerImpl` is doing this. Please also see `testEventListener` method in `BankAccountHandlerImplTest` class where test is added for this.

Error handling shall exist

Errors and exceptions related to this module are handled with error handling class `BankAccountException`.

Log relevant information to file (using Log4j) or other log framework)

Apache Log4j is added as maven dependency. This framework is used to write log in file. Log history is saved in `bankaccount.log` file. The directory of the log file is configurable from `log4j.properties` file which is inside directory: `../src/main/resources`.

Simultaneous requests shall be supported (thread safe)

`synchronized` block is used in `addAccount`, `addTransaction` method inside `DataStore` class to make account adding and account transactions thread safe.

Unit tests shall exist

Tests are added in directory: `../src/test/java`. Sufficient test cases are written to test all the functionalities and requirements given in the problem document. `BankAccountHandlerImplTest` class is added with tests bank operations. `DataStoreTest` class is added to check the memory data store is working properly. Comments are written above the test methods, which shortly explains the purpose of that particular test.

Conclusion

The module is written keeping simplicity, modularity and enterprise development concept in mind. It is also written in a way so that it can be tested easily. All the requirements given in the bullet points are taken care of in this solution.