Scala Project Part 1 – Threads

TDT4165 – Programming Languages 2015

1 Exercise Description

Traditional online banking applications are currently experiencing great competition from new players in the market who are offering direct transactions with a few seconds of response time. Banks are therefore looking at possibilities of changing their traditional method which involves batch transactions at given times of day with hours in between. They must now update their software systems to adapt to the current demand, which means transactions must be handled in real-time.

Your overall task for this project is to implement features of a real-time banking transaction system.

In the zipped folder `Exercise` is the source code for Part 1 of the project. Unzip the folder and import its contents to the Scala IDE of your choice. The file structure is presented below.

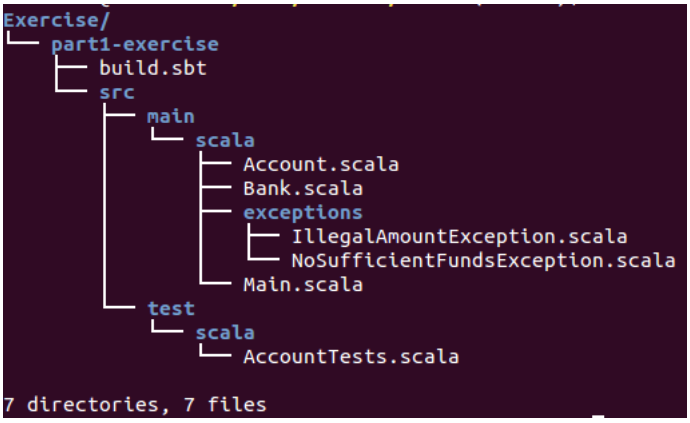

Figure 1: File structure

For this part, you will be working in `Account.scala`, `Bank.scala`, and `Main.scala`. The main objective is to implement missing functionality and pass all of the tests.

- **Account**
  In the `Account` class, one should be able to withdraw (take out) or deposit (insert) a certain amount of money, which will change the total balance in the account. Throw suitable exceptions for invalid states.
- **Bank**
  In `Bank`, a method for transactions between accounts should be implemented. Throw suitable exceptions for invalid states.

- **Main**
  In `Main`, we have provided an implementation of a thread that you can play around with to familiarize with the concept, and test your code. If you are having issues running the `Main`-file, make sure the `src/main/scala`-directory is marked as a source folder, or use `sbt run` in the command line.

2 Running the Tests

2.1 IDE

To run the tests within a Scala IDE, simply run `AccountTests.scala`. If you do not have the option of running this file, make sure the `src/test/scala`-directory is marked as a source folder (Eclipse) or a test source root (IntelliJ).

2.2 Command Line

To run the tests from the command line, `cd` into the `part1-exercise`-directory, and run the `sbt test` command.

If you have not yet installed sbt, visit `www.scala-sbt.org/release/tutorial/Setup.html` and follow the installation instructions for your OS.

3 Deliverables and Deadline

To submit your solution, upload the files `Account.scala`, `Bank.scala`, `Main.scala`, and any other modified/added files to itslearning before **October 30th**.

Your code should be presented to a TA during lab hours before **November 6th**. For your submission to be approved, 70% test coverage is required; 7 of the 10 tests need to pass, where these 7 passed tests *must* include either “Correct balance amount after several withdrawals and deposits” or “Correct balance amounts after several transfers”.