Exercise 2

1) Get the test projects from the BitBucket repository (GIT) if you do not have it yet:

<https://bitbucket.org/leo91araujo/sample-bank>

<https://bitbucket.org/leo91araujo/sample-bank-be>

2) Sample-Bank: The **Withdrawal Feature** needs to be automated! Create a **new story and write two withdrawal scenarios**. Validate the happy path (INF001 - Operation completed with success) and a negative path, the user cannot withdraw if he has no balance in his account (ERR008 – Insufficient Balance). In the first one you should use a **normal scenario** and in the second one you should use an **examples table**. Use the story “account\_transactions.story” as example.

Extra Tips:

- To run the **Sample Bank Application** open the class Application.java that is inside the project sample-bank (if you`re using IntelliJ, run it through the Maven Spring Boot Plugin goal: spring-boot:run)

- (At this time, please do not worry about the service technical details, we are going to see it further soon! The class **AccountServiceTestClient** already has the necessary method to perform this operation, just call the method ***makeWithdrawal (ownerAccount, amount)***)

- The textual stories are inside the **folder src/main/resources/stories**

- To run your story, use the class **AllStories.java** (it is inside the package com.wipro.tutorial.ta), just open it, click with the mouse right button and select “Run” as Junit.

- Your report will be generated inside the project’s target folder. You can find this information in the test execution logs (eclipse console), it is HTML, just open using it your browser

- If the JBehave report is unformatted (without any style) execute the maven command to unpack the JBehave styles, inside the JBehave project execute: **mvn jbehave:unpack-view-resources**