Lesson 01 Demo 03

Using Various Assertions in JUnit5

Objective: To create various assertions in JUnit5

Tool required: Eclipse IDE

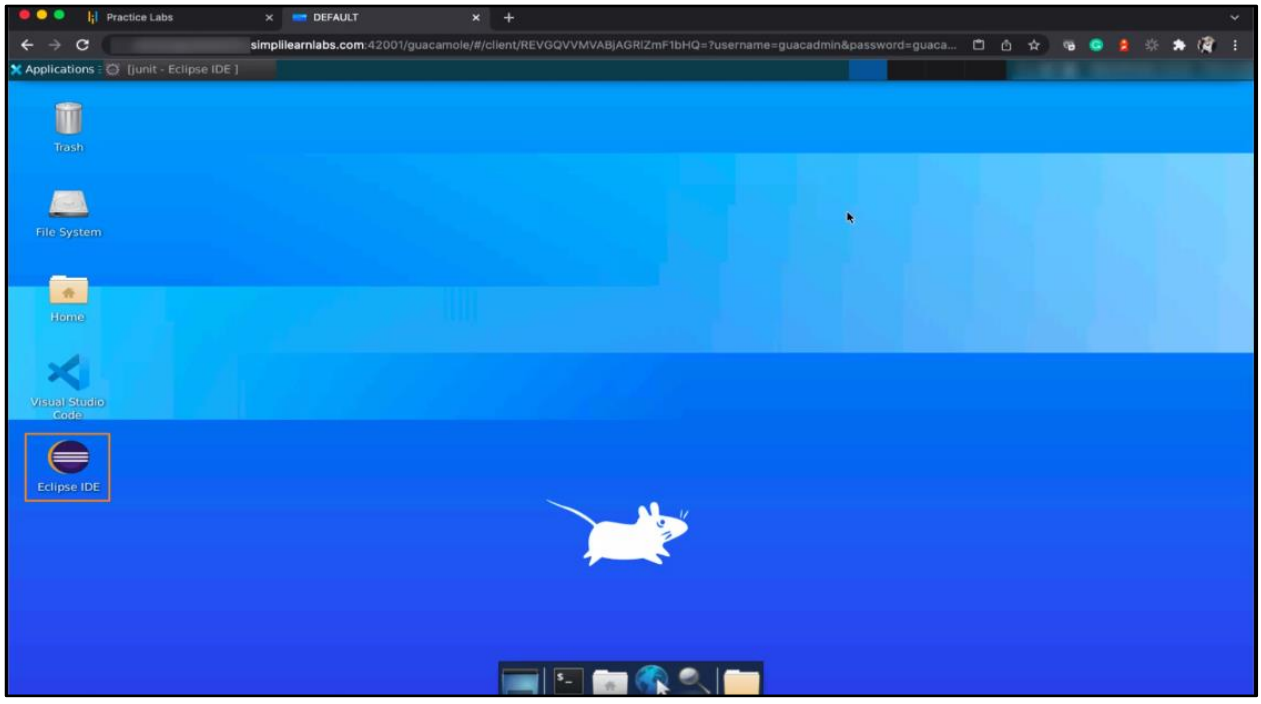
Prerequisites: None

Steps to be followed:

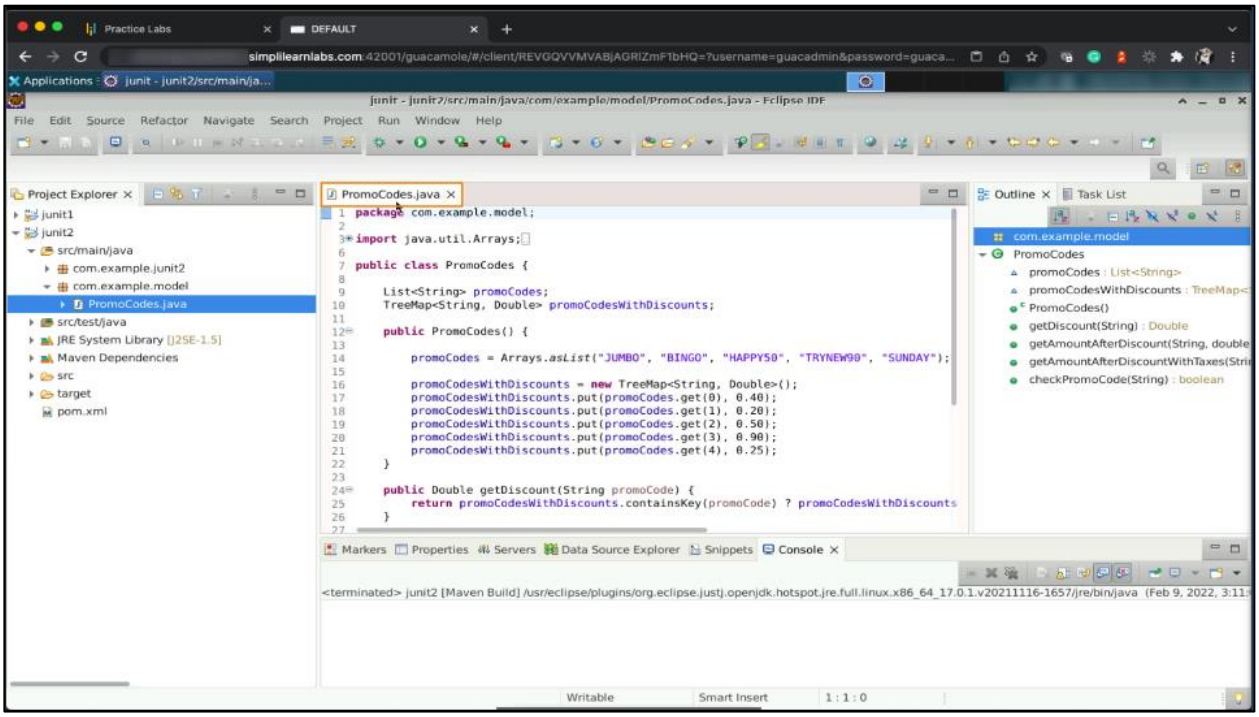
1. Using assertEquals()
2. Using assertAll()
3. Using assertThrows()
4. Using assertTimeout()

Step 1: Using assertEquals()

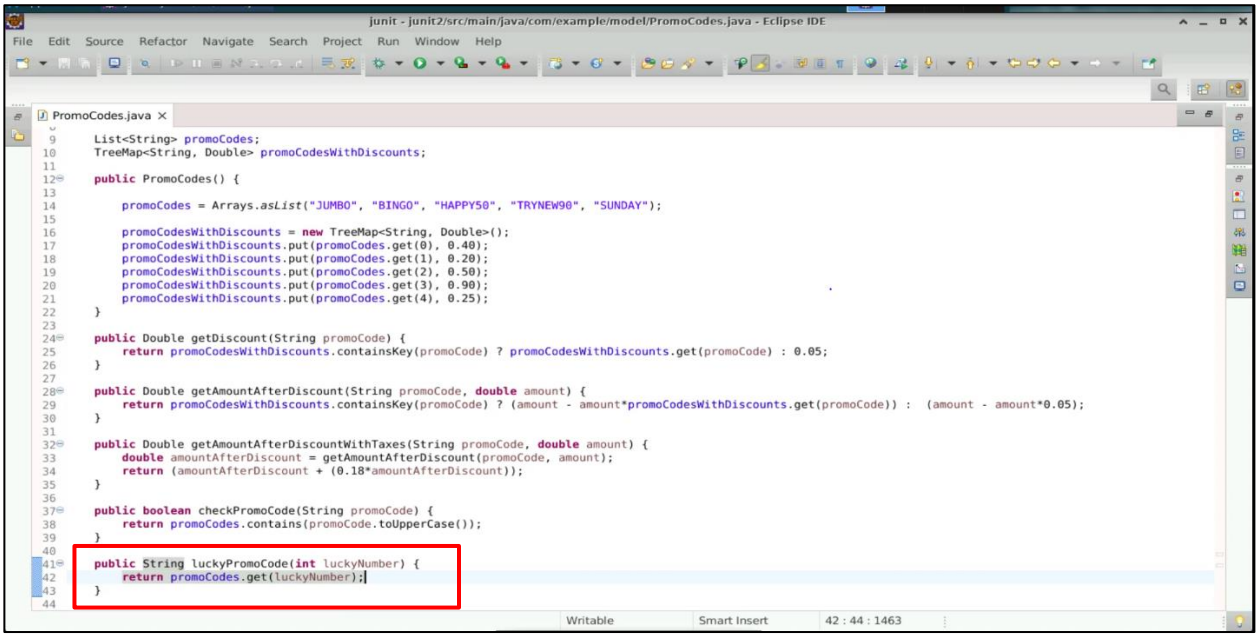
1.1 Open **Eclipse IDE**



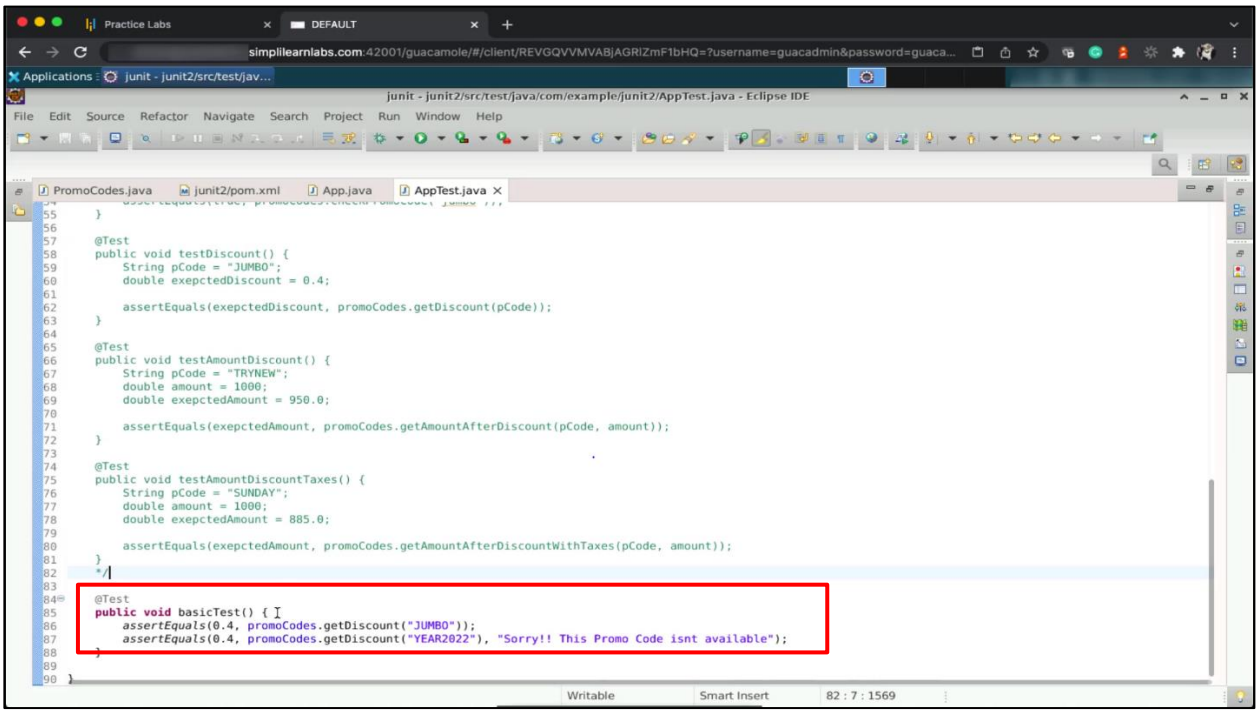
1.2 Go to **junit2** project. Then, go to **src/main/java > com.example.model** and open the **PromoCodes.java** file



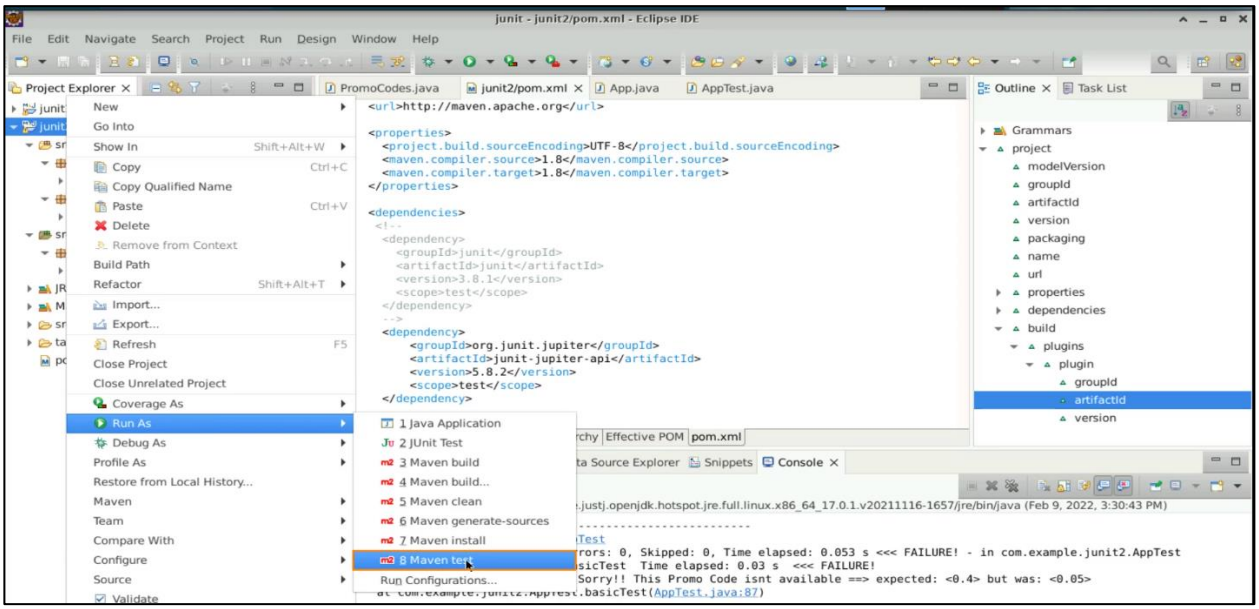
1.3 There are four methods available. Create one more method **luckyPromoCode()** and pass **luckyNumber** inside this method. Return the **promo code** from the promo code list



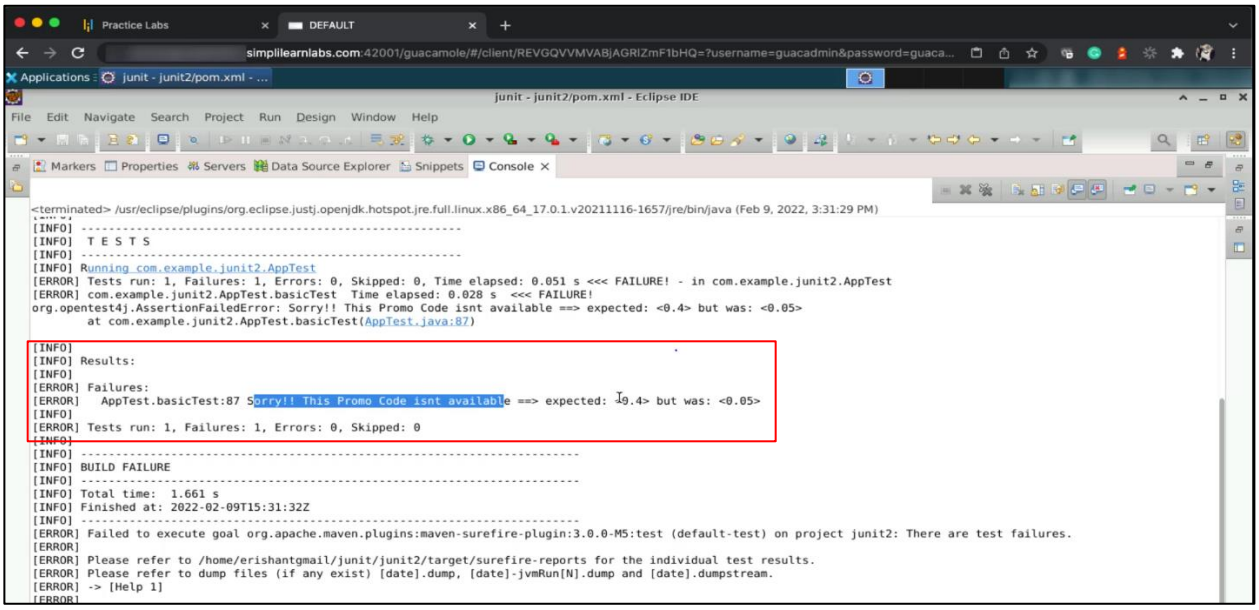
1.4 Go to **src/test/java > com.example.junit2** and open the **AppTest.java** file. Create a new test case **basicTest()** to test **assertEquals()** by passing the **expected** and **actual** values. Also, give a failure message to handle errors. Comment on all the other test cases



1.5 Now, right-click on project **junit2.** Click **Run As** option and build this project as a **Maven test**

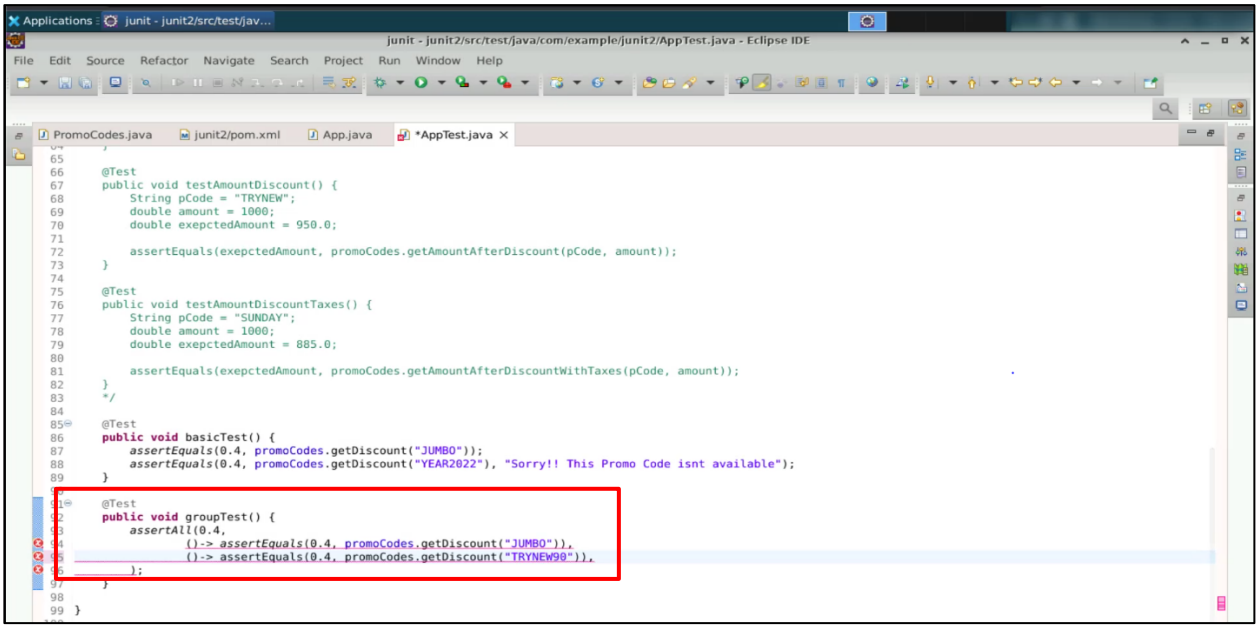


In the console output, you can see 1 test case failed with the same message.

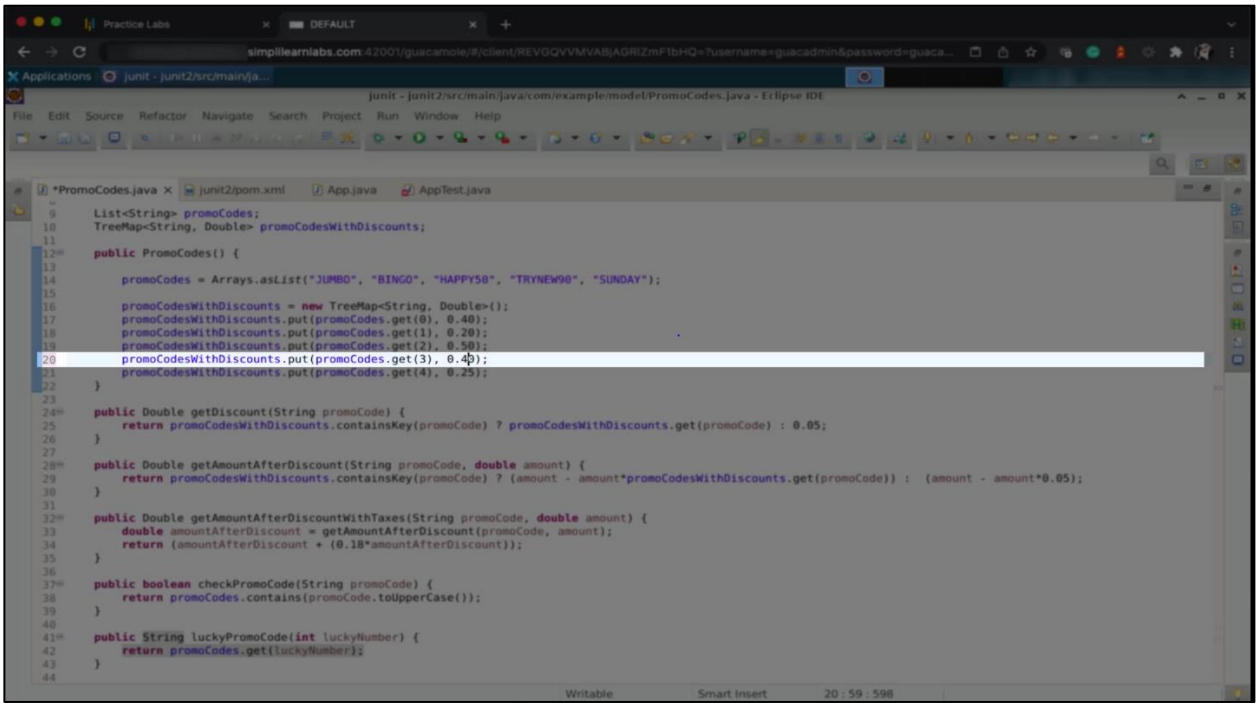


Step 2: Using assertAll()

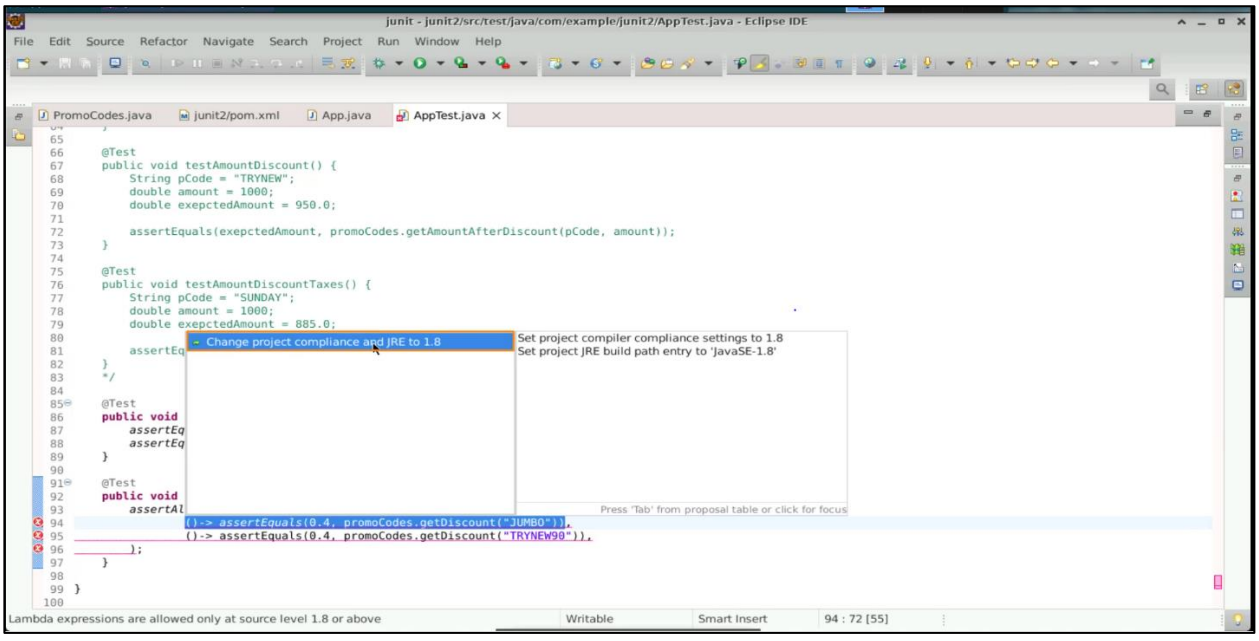
2.1 To test multiple cases in one go, create a new test case **groupTest().** Create **assertAll()** assertion, pass single value, and two lambda expressions



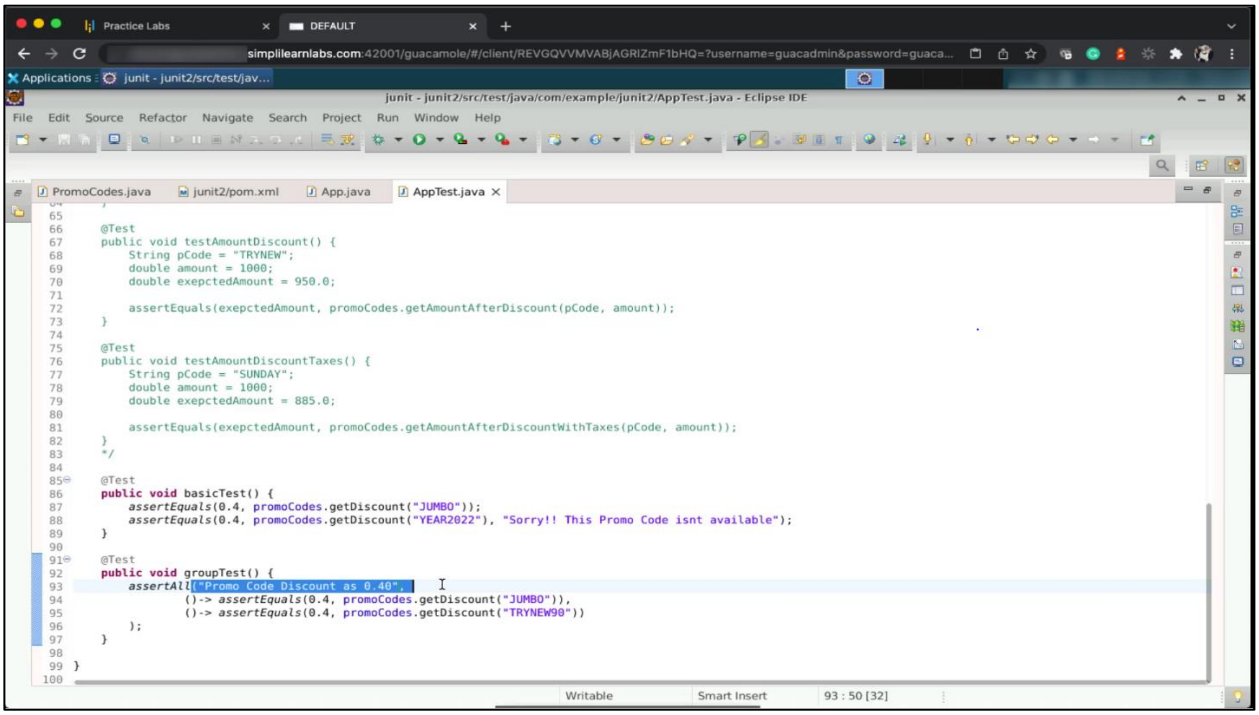
2.2 Go to the **PromoCodes.java** file and change the third promo code discount value from **0.90** to **0.40**



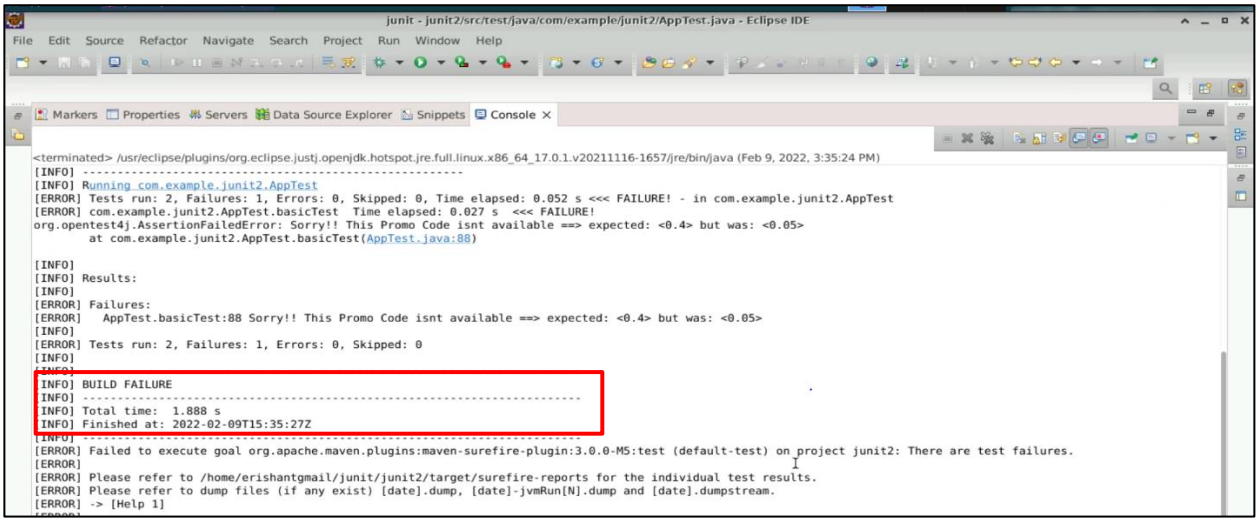
2.3 Go back to the **AppTest.java** file, and you will get an error. To remove this error, click on the red cross and choose **change project compliance and JRE to 1.8**



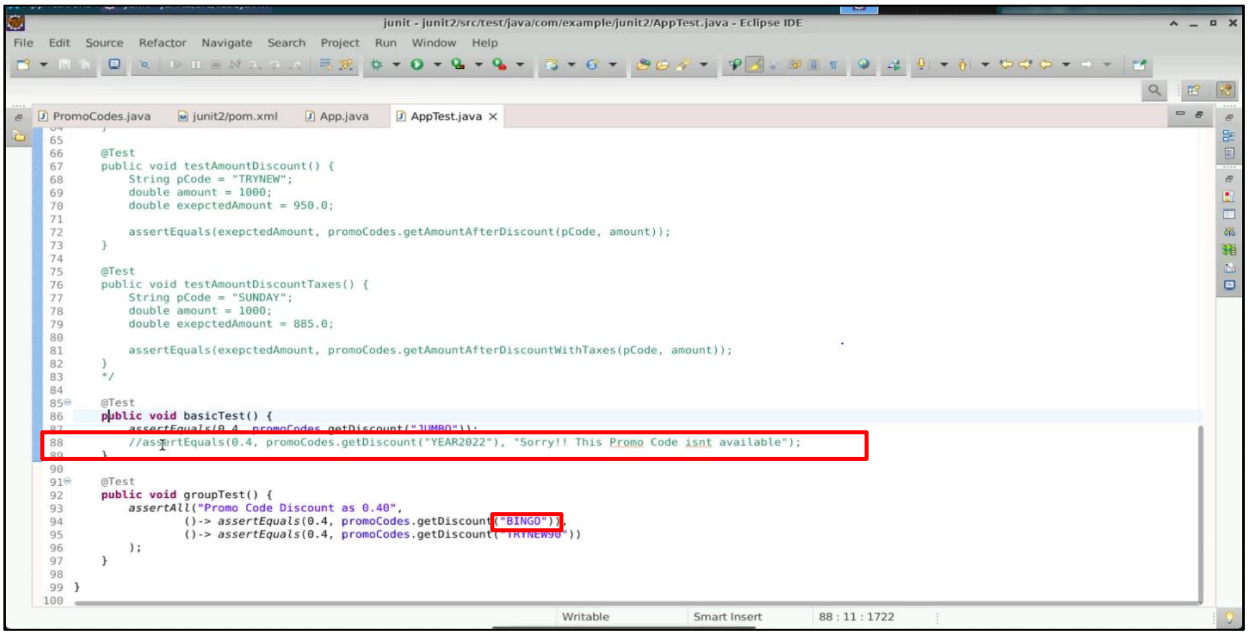
2.4 The expected single value we pass in **assterAll()** should be a string. So, pass **"Promo Code Discount as 0.40"**



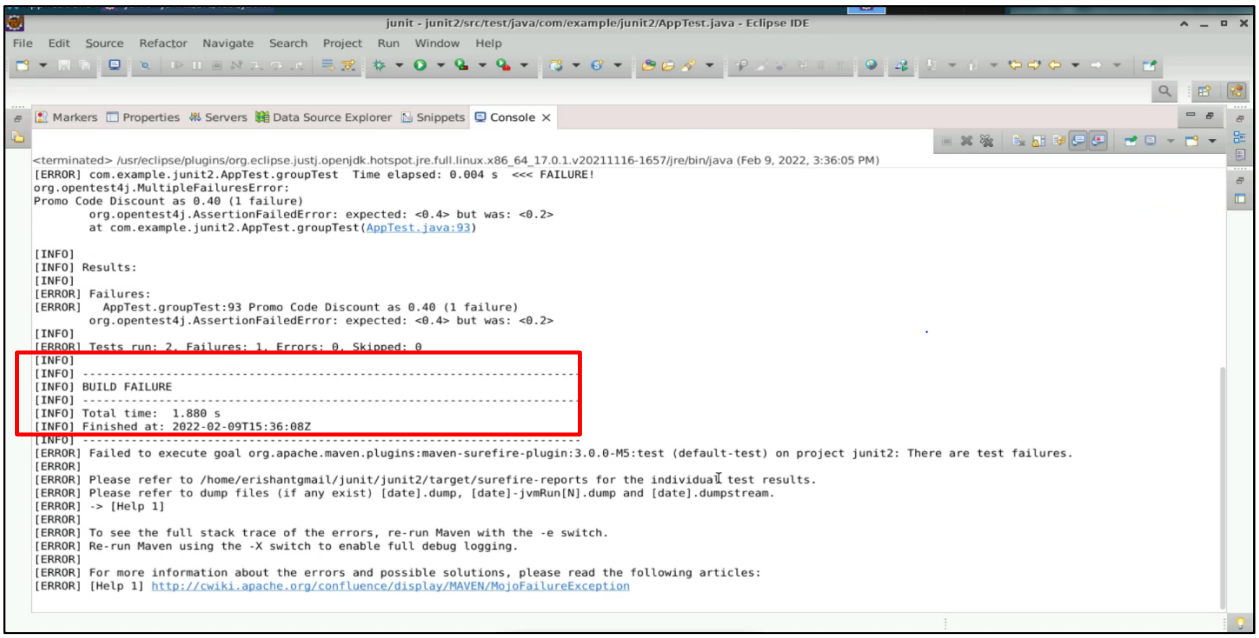
2.5 Now run the code as a **Maven test** again. In the output, you can see 2 test cases running and 1 failing, which is the previous test case. The new groupTest case will pass.



2.6 Now, go back to the **AppTest.java** file. Change the promo code to **BINGO** which has a **0.20** discount value. Comment the error **assertEquals()** from **basicTest()** test case

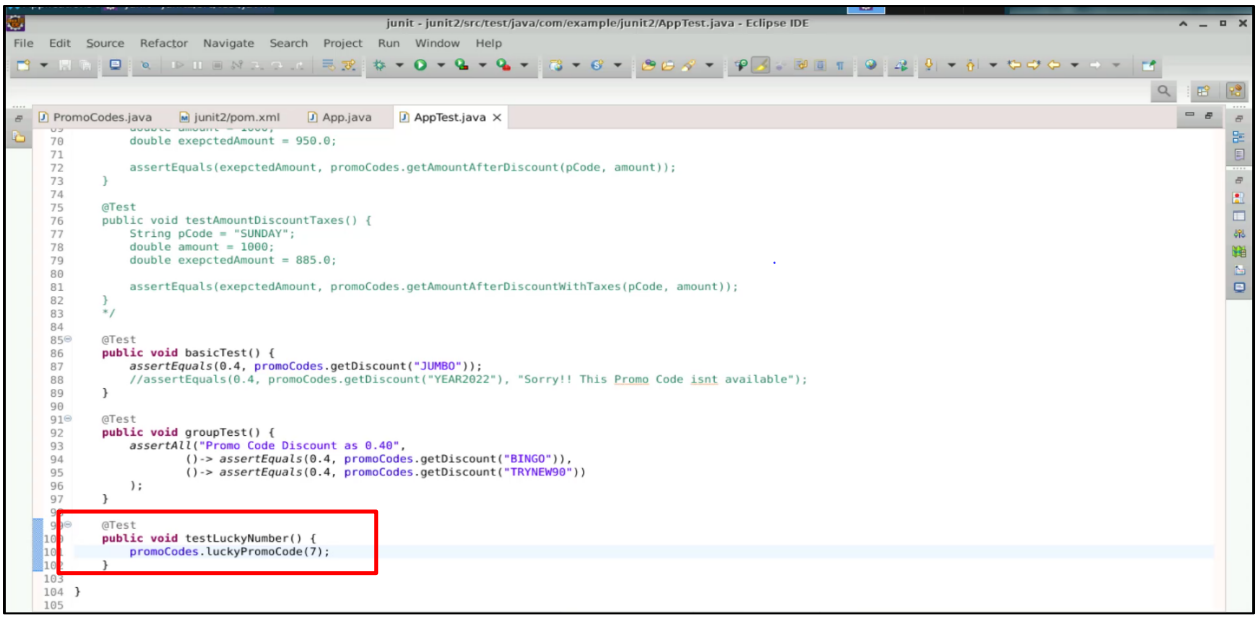


2.7 Run the code again as a **Maven test.** You can see both test cases run again in the output, and 1 fails. This time the second test case failed as the **BINGO** promo code has an actual value of **0.20**. The expected value is **0.40.**

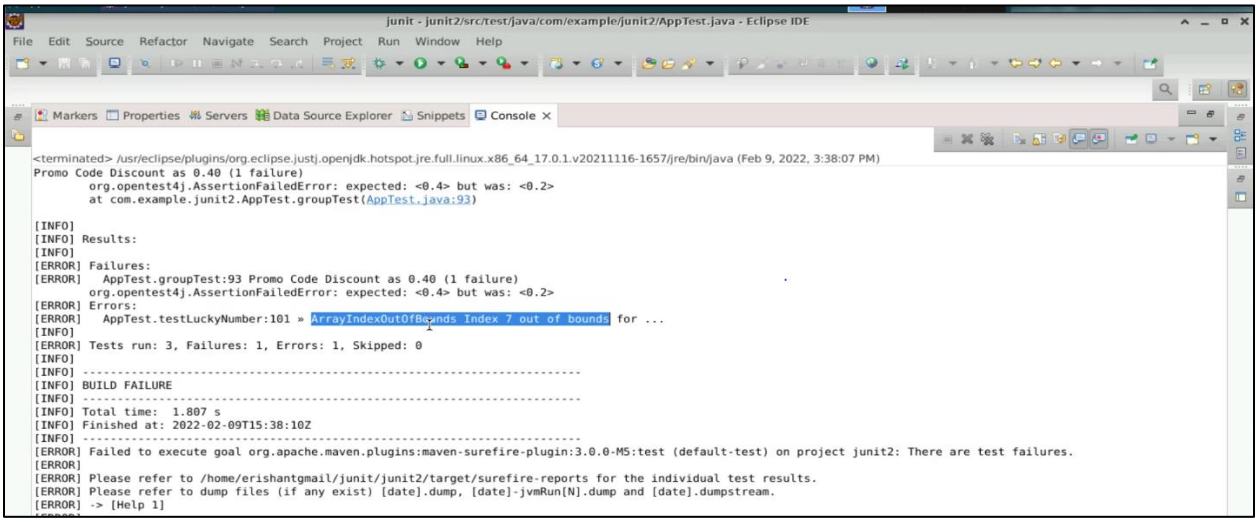


Step 3: Using assertThrows()

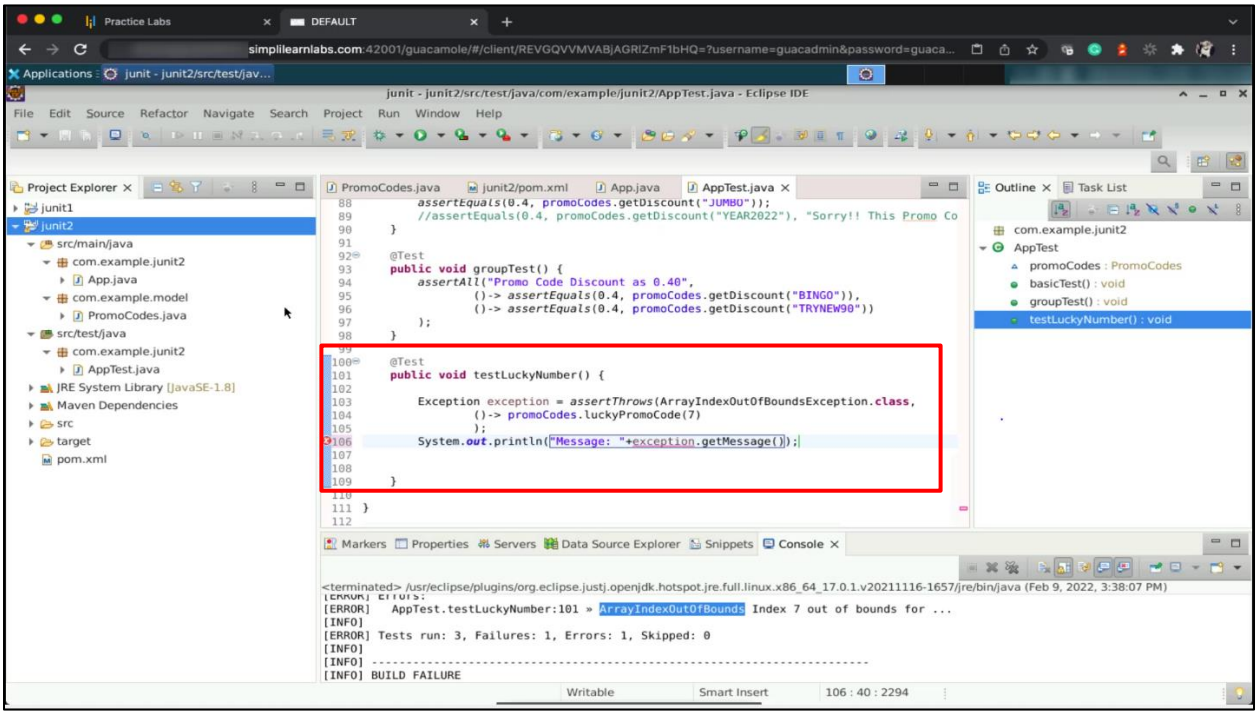
3.1 Now, go back to the **AppTest.java** file. Create a new test case name **testLuckyNumber()** and execute this method for the lucky number **7,** which is not available in the list



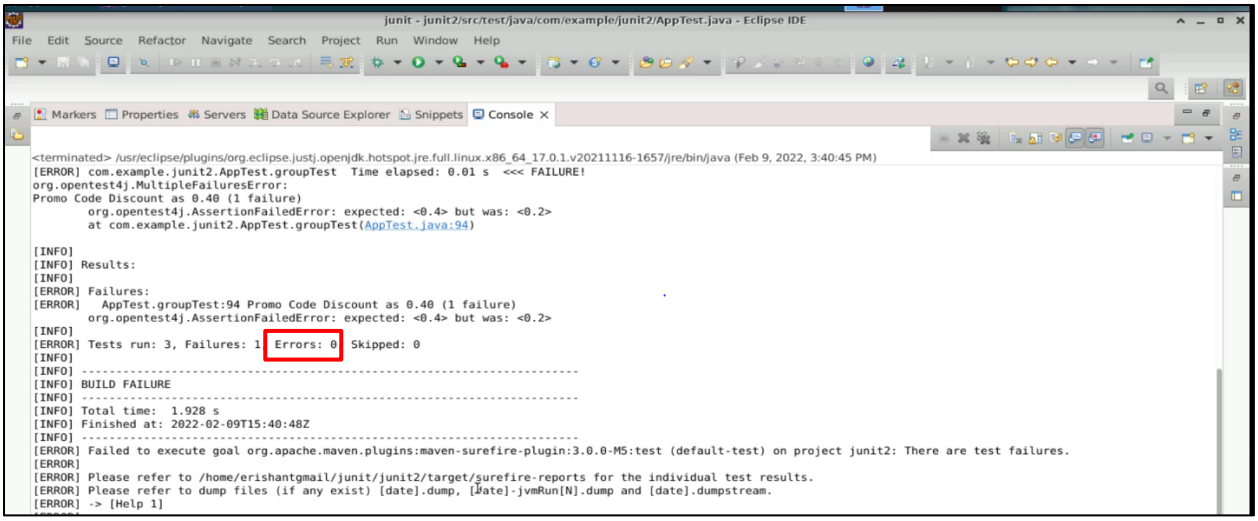
3.2 With the changes made, run the project again as a **Maven test**. You will see 3 test cases running, and this time you will get 1 error with the error message. **ArrayIndexOutOfBounds Index 7 out of bounds**



3.3 To handle such scenarios, go back to the **AppTest.java** file and use the **assertThrows()** assertion to handle this type of exception. In **assertThrows()**, first, specify the type of class as **ArrayIndexOutOfBounds.class,** then pass the Lambda expression with **luckyPromoCode(7).** Also, add a message to identify in the console

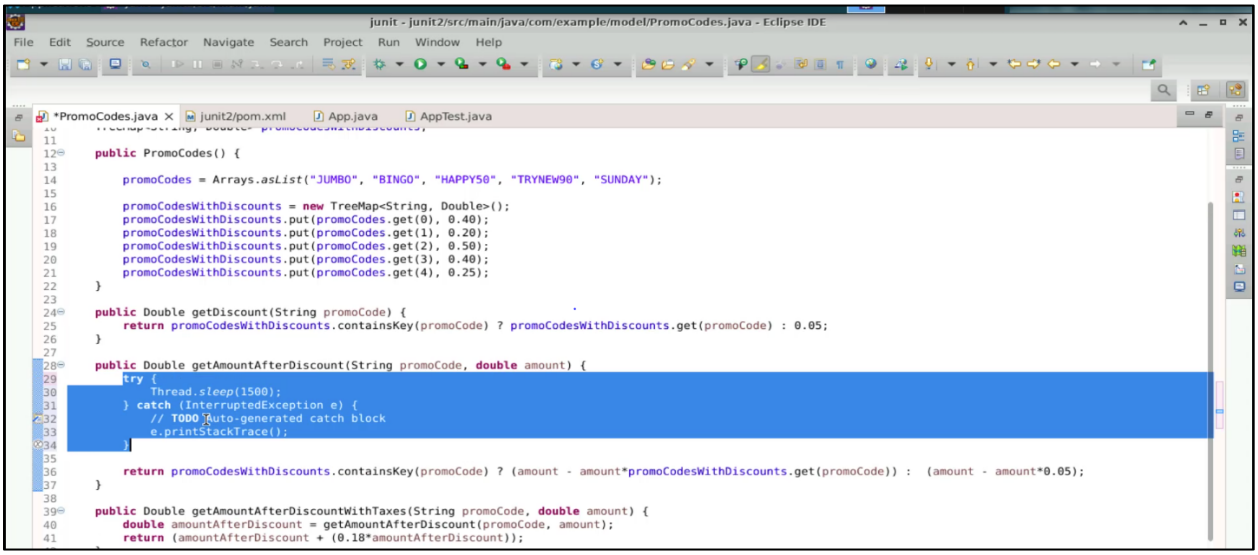


3.4 Again, run the project as a **Maven test.** You can see the error was handled by the **assertThrows()** assertion



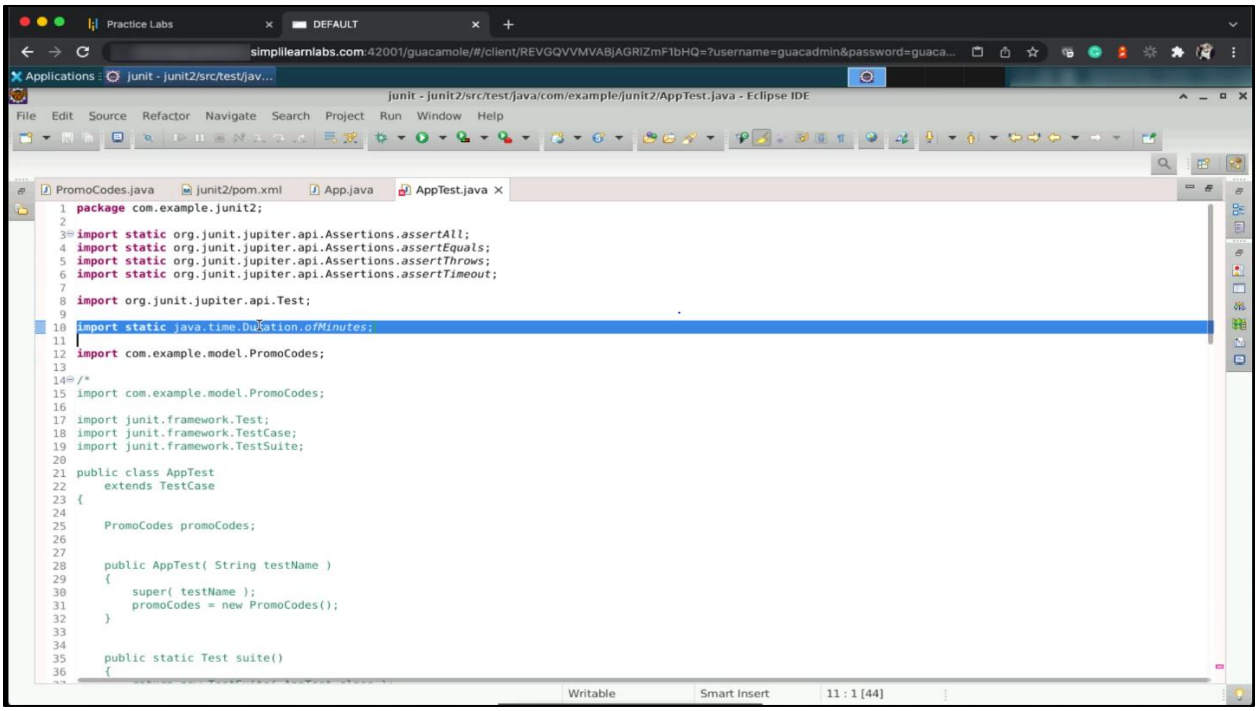
Step 4: Using assertTimeout()

4.1 Go to the **PromoCodes.java** file. Add sleep time of 1.5 sec in the method **getAmountAfterDiscount(),** which is calculating the discount. Surround this API with a **try-catch** block

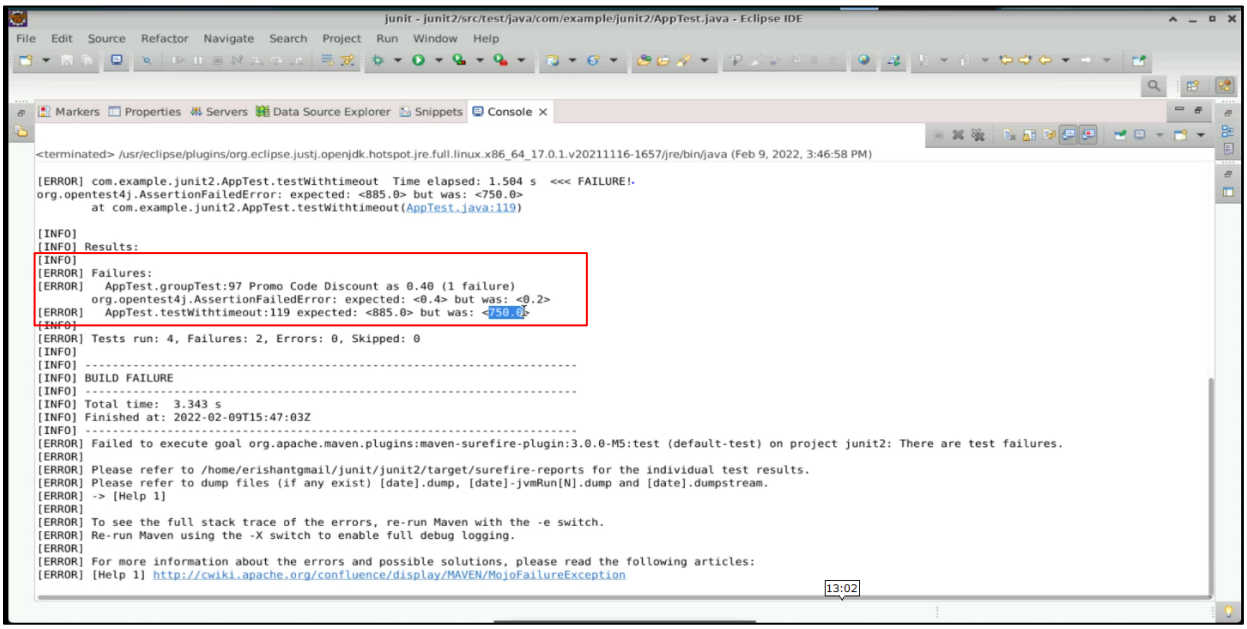


4.2 Go back to the **AppTest.java** file. Create a new test named **testWithtimeout().** Use **assertTimeout()** to fix the time at which you need the results. Use the method **ofMinutes()** and give the expected return value through the Lambda function. Also, add the **assertEquals()** to compare the expected and actual amount

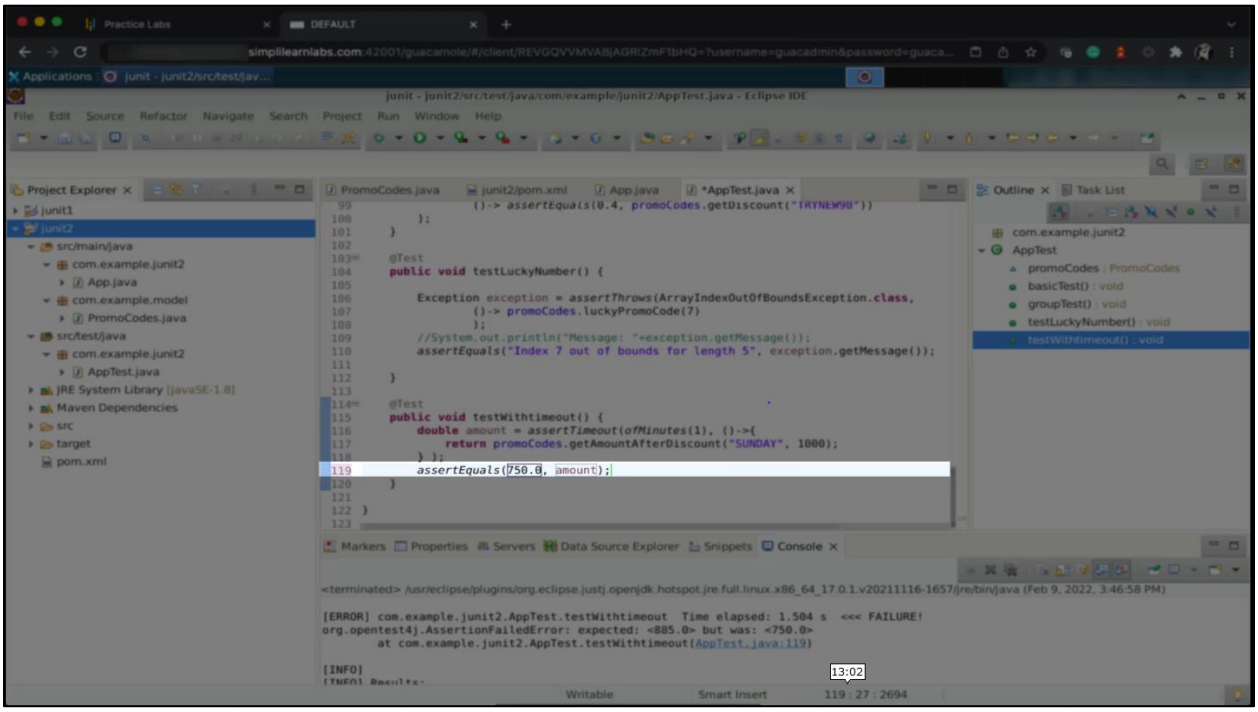
4.3 Scroll up and check **ofMinutes()** method was imported or not, if not import manually



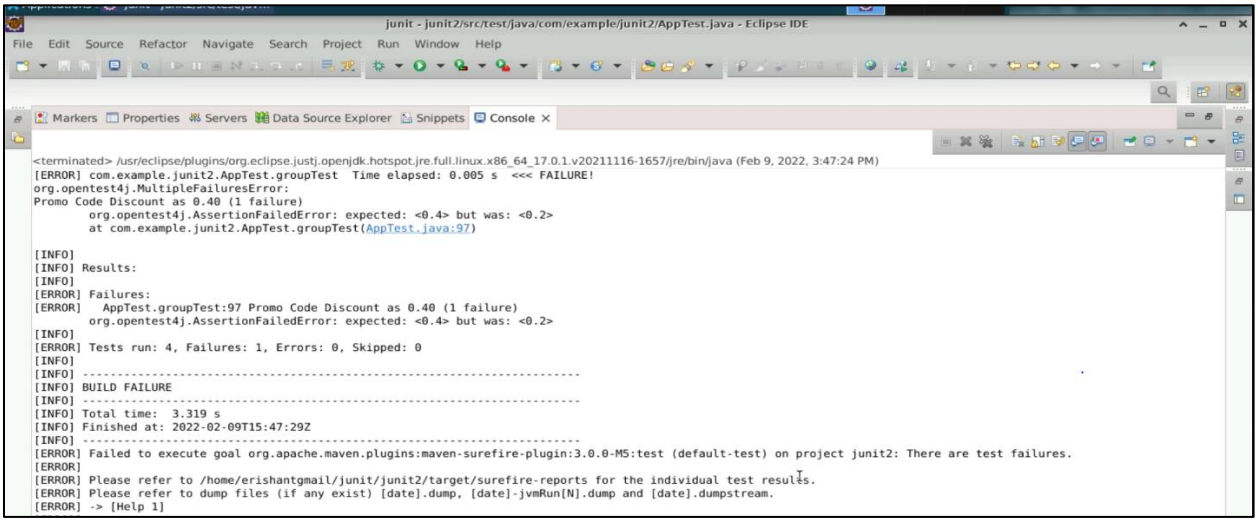
4.4 Run the code as a **Maven test**. You can see the test has failed because the expected discount is **885** and we got **750.**



4.5 Go back to the **AppTest.java** file and change the expected amount to **750**



4.6 Rerun the project. You will see that the **testWithTimeout()** test case passed.



This is how we can perform different assertion methods in the Java-JUnit project using the **assertEquals(), assertAll(), assertThrows()**, and **assertTimeout().**