Problem 1

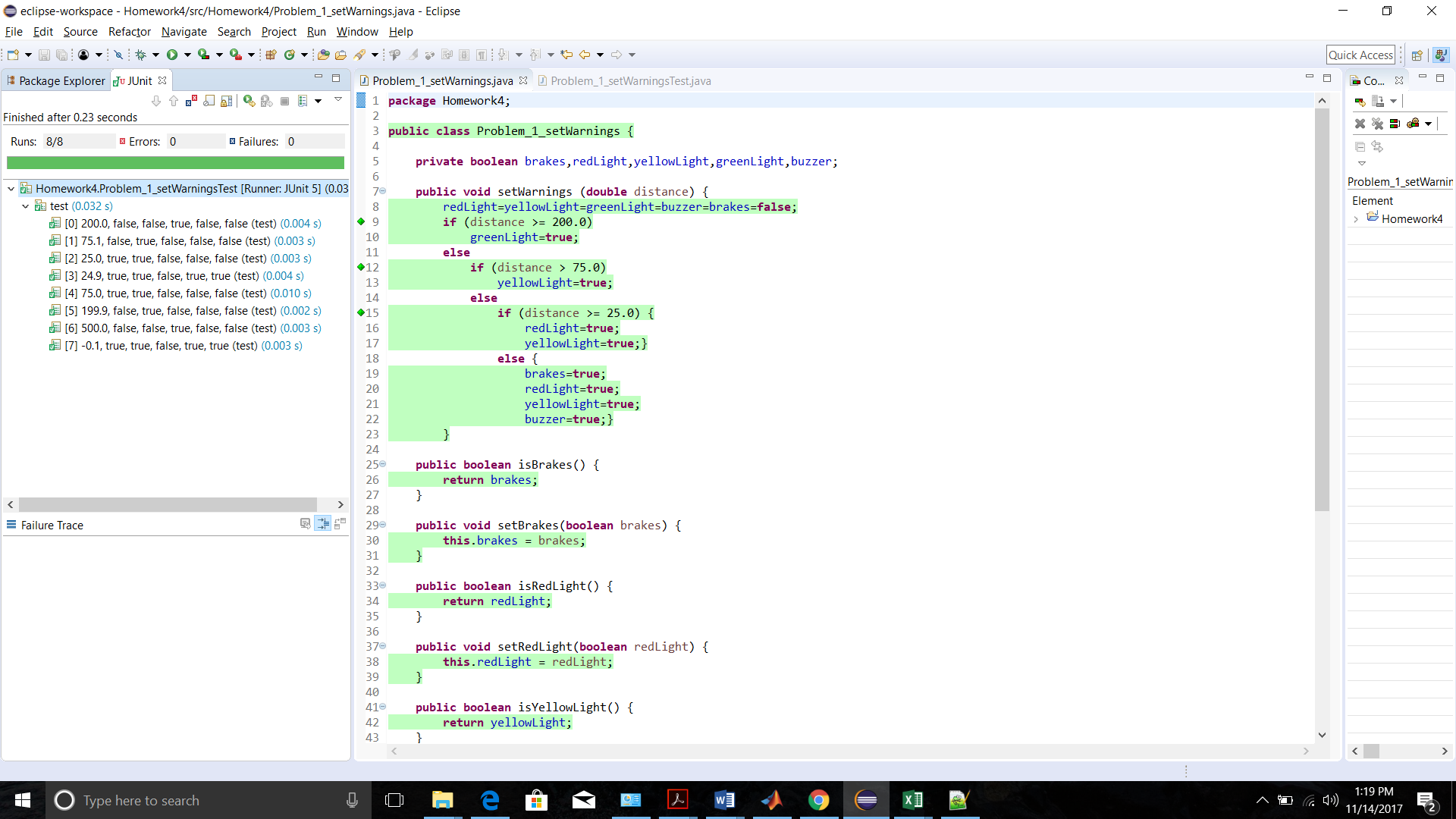
Use the code and the test cases we developed from Homework 3 for Problem 1. Implement this using the **JUnitParamsRunner**.

Solution1 Following was the table for homework 3 Problem 1



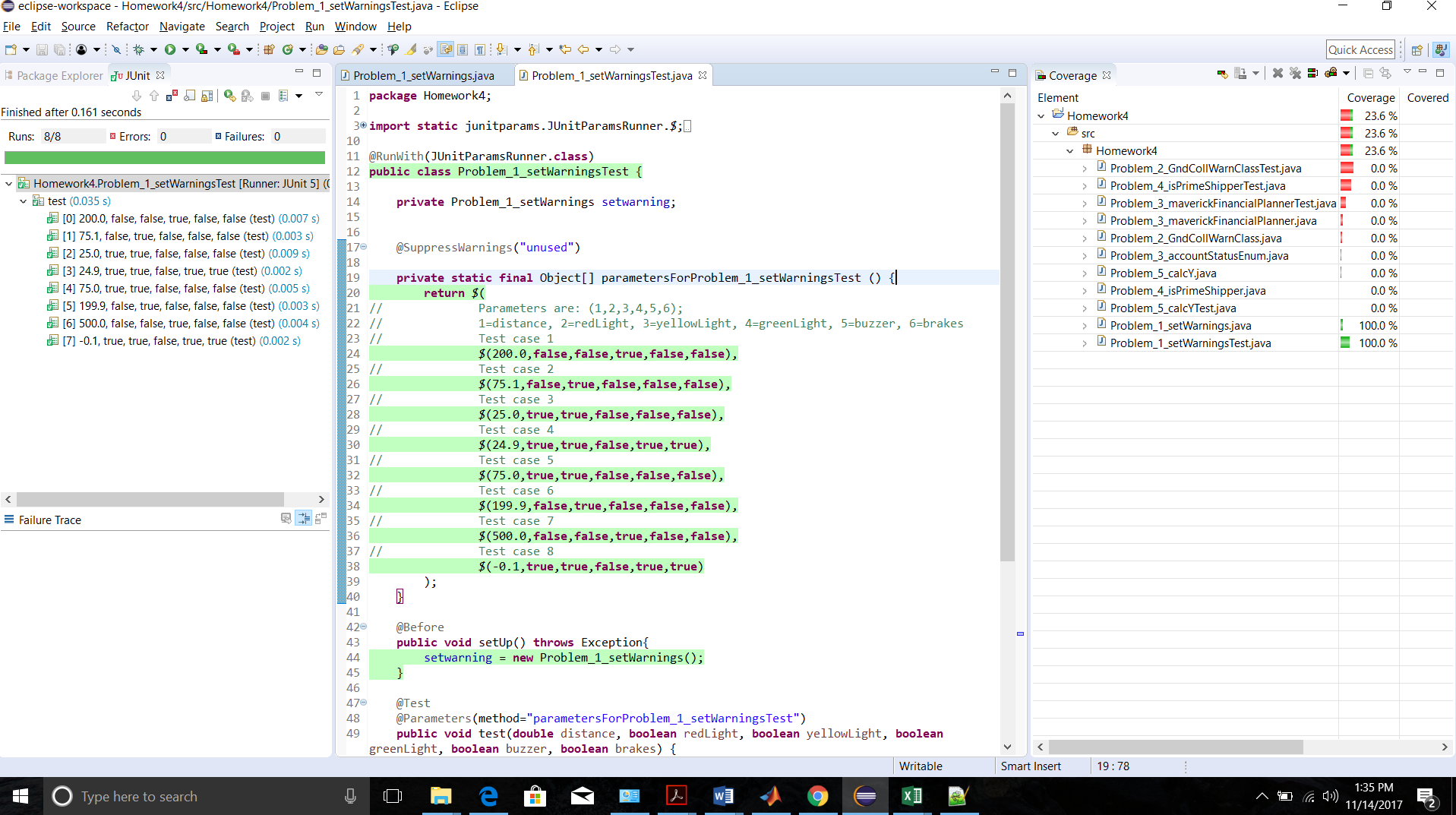
Screen shots for JUNIT and JACOCO coverage :-

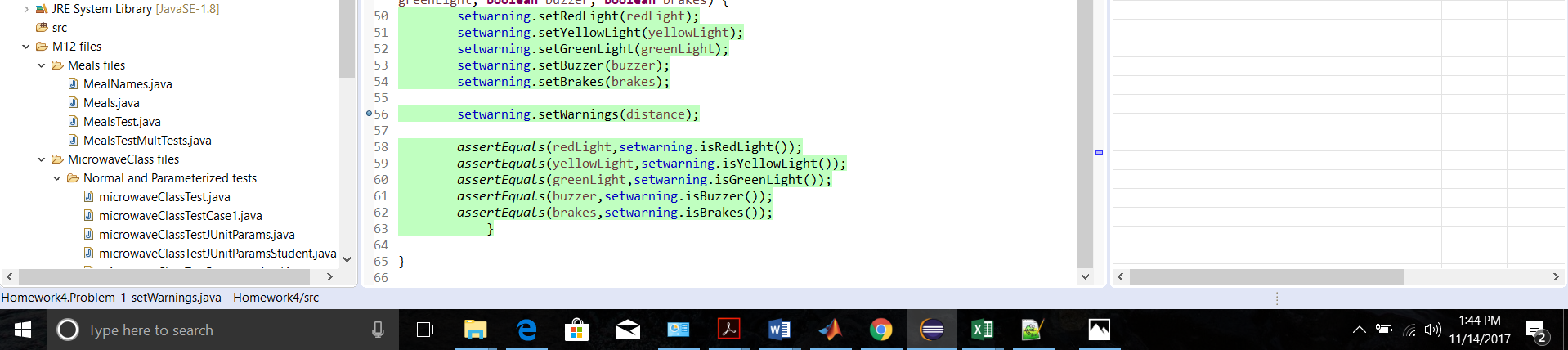
Main Java file with Junit green bar showing all the parameters executed and green highlighting showing full JACOCO coverage for the logic.





Junit implementation using **JUnitParamsRunner**





Problem 2

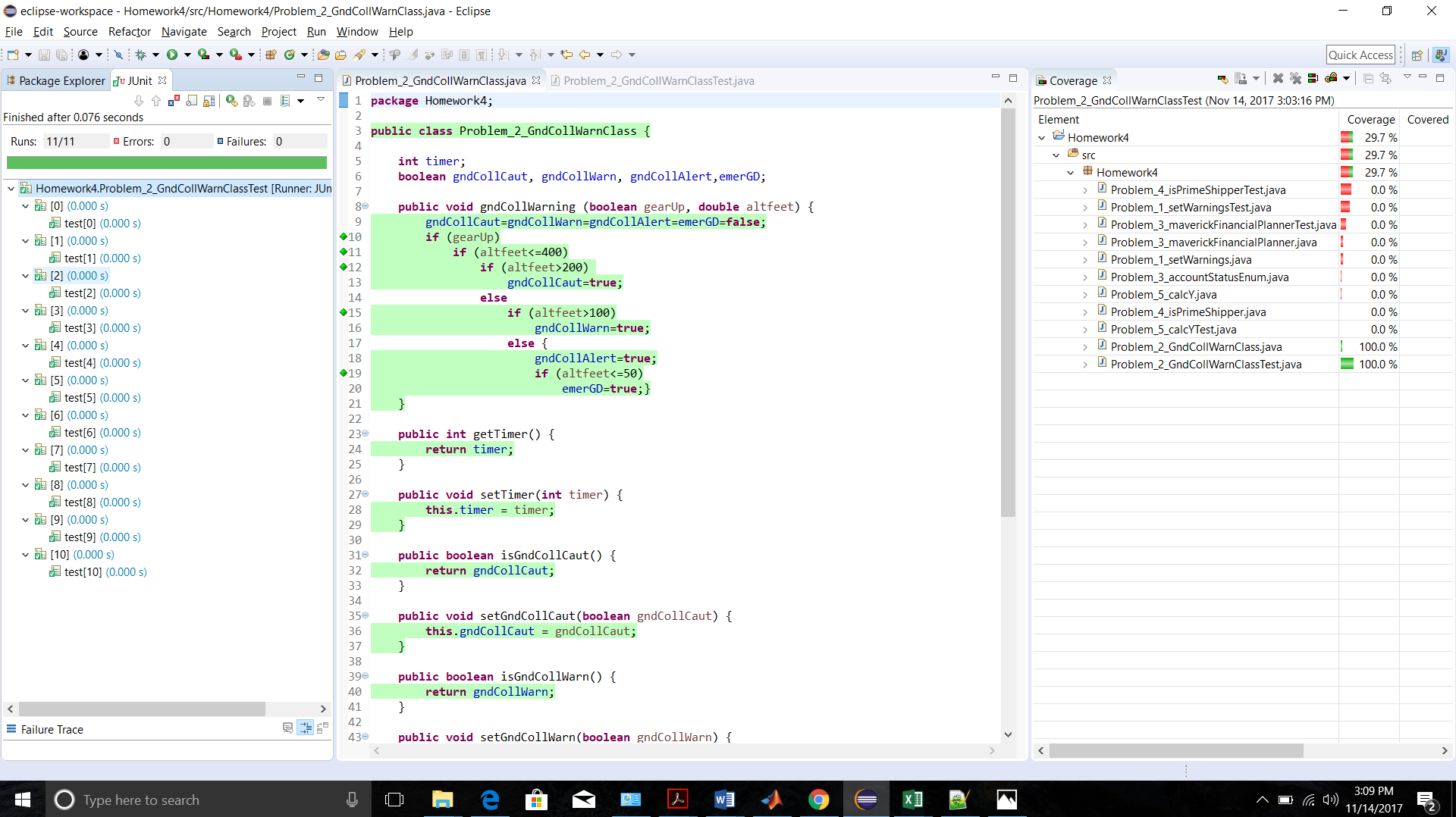
Use the code and the test cases we developed from Homework 3 for Problem 2. Implement this as a **Parameterized.class** test.

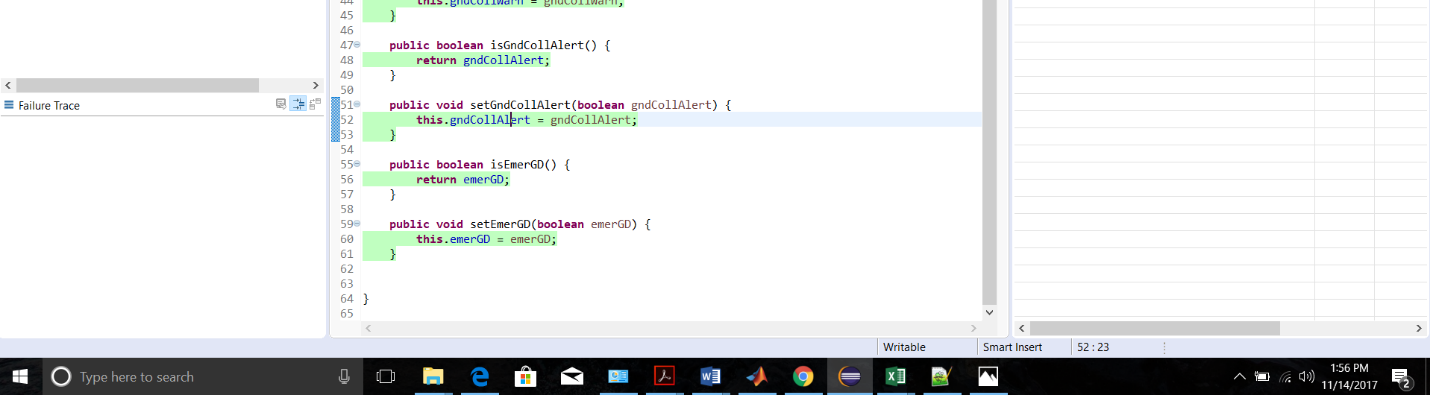
Solution2 Following was the table for homework 3 Problem 2



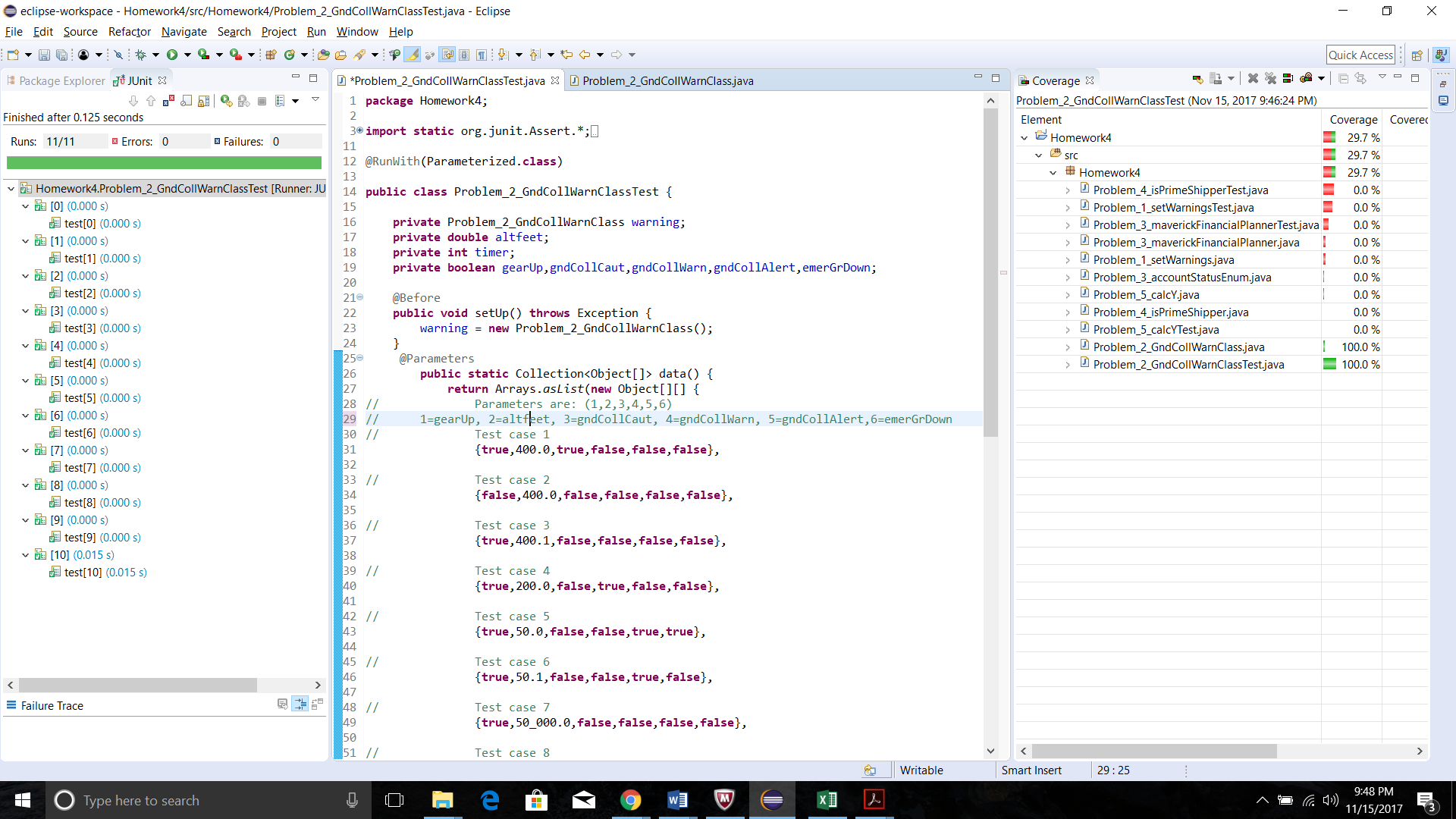
Screen shots for JUNIT and JACOCO coverage :-

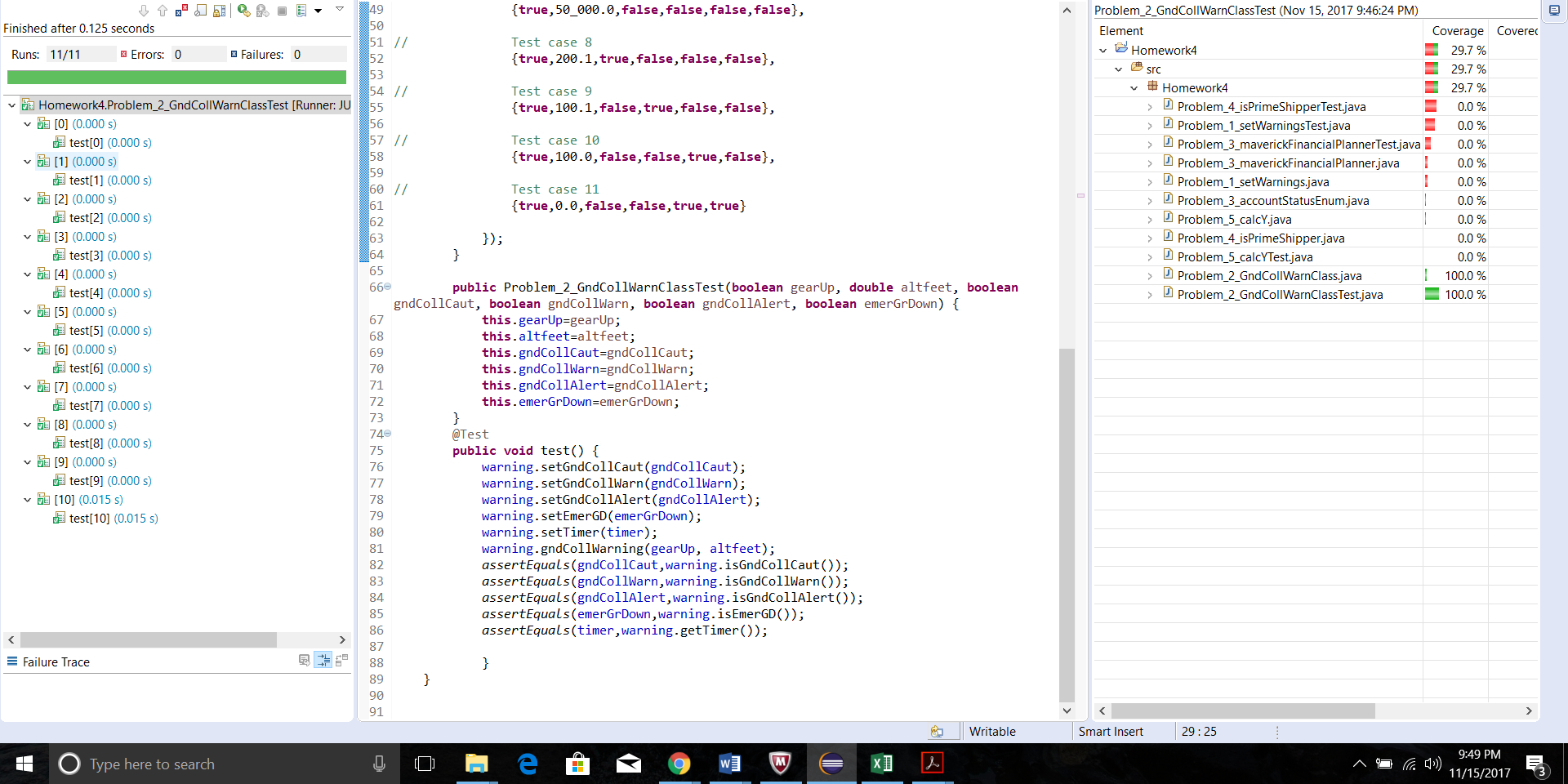
Main Java file with Junit green bar showing all the parameters executed and green highlighting showing full JACOCO coverage for the logic.





Junit implementation using **Parameterized.class**



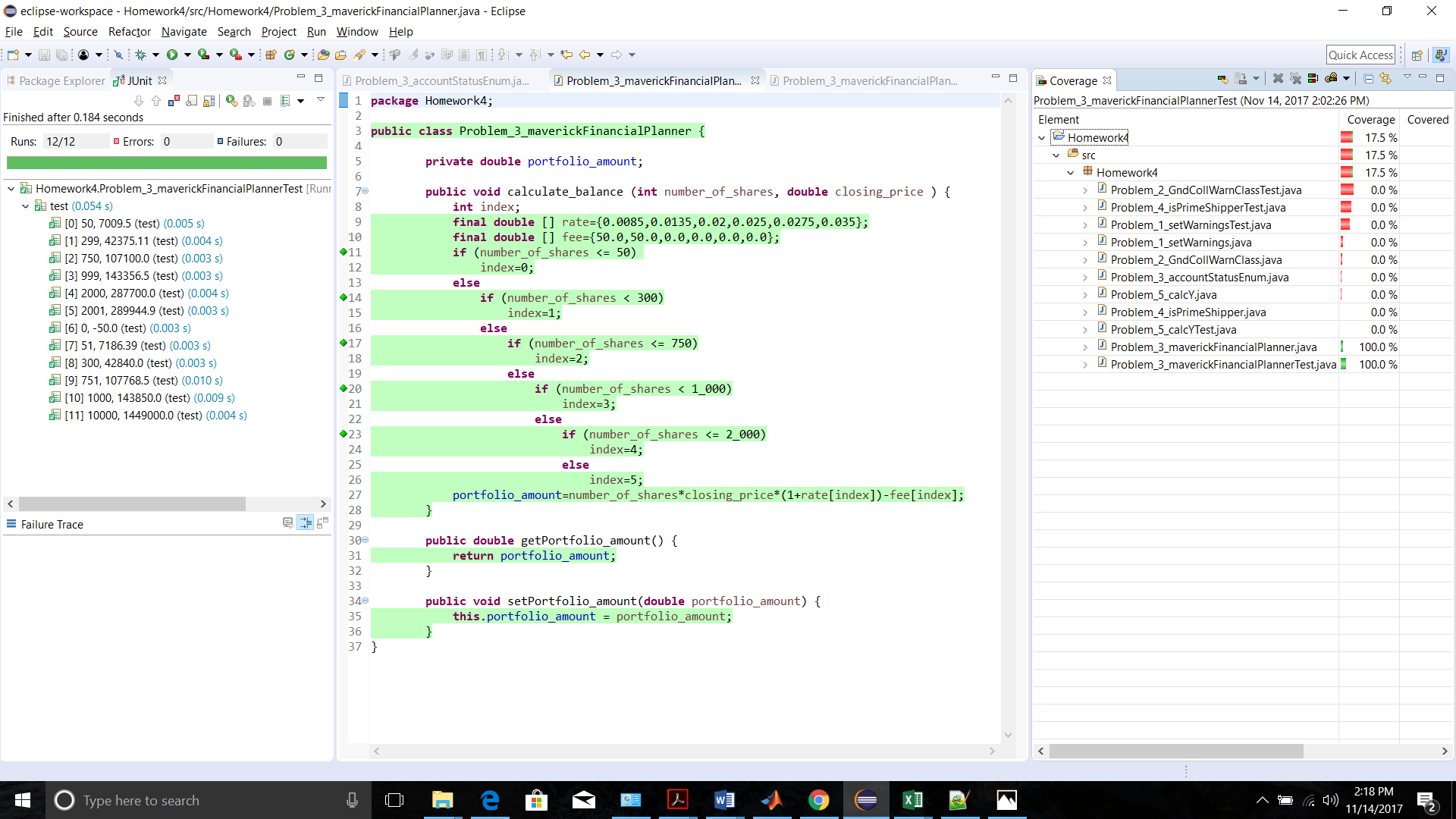


Problem 3 Use the code and the test cases we developed from Homework 3 for Problem 3. Implement this using the **JUnitParamsRunner**. Test portfolio\_amount to a threshold of 0.01

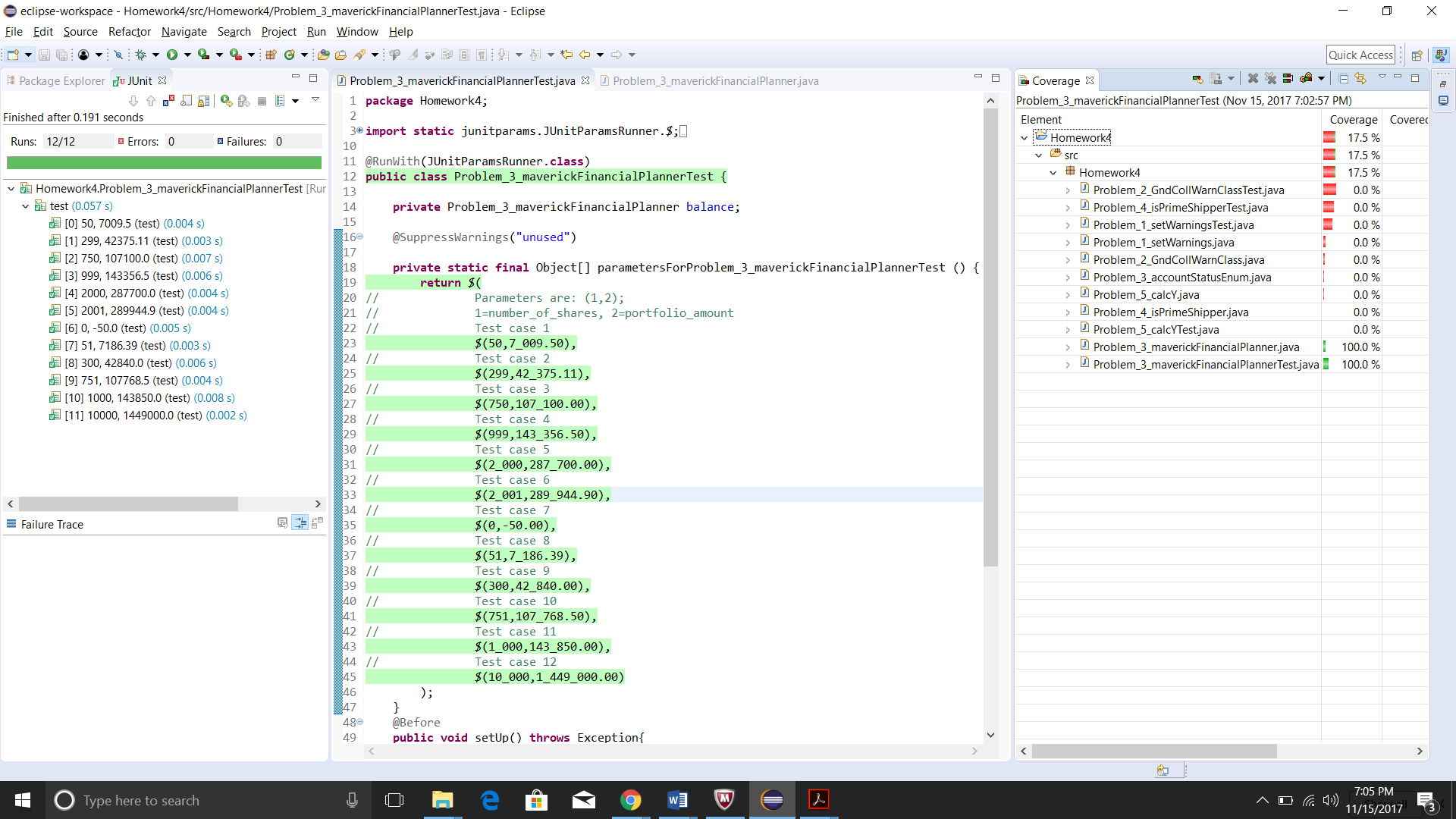
Solution3 Following was the table for homework 3 Problem 3 :-

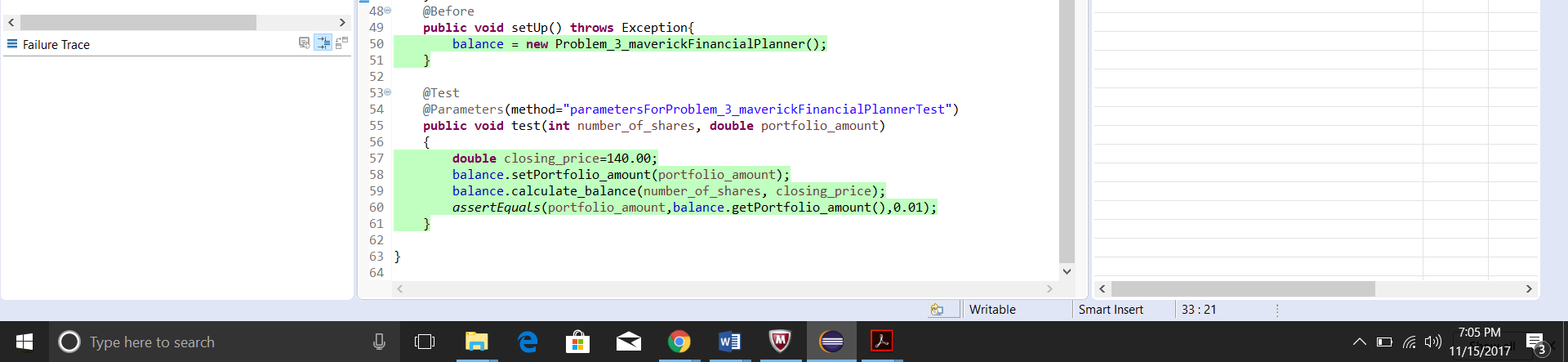


Main Java file with Junit green bar showing all the parameters executed and green highlighting showing full JACOCO coverage for the logic.



Junit implementation using **JUnitParamsRunner**





Problem 4

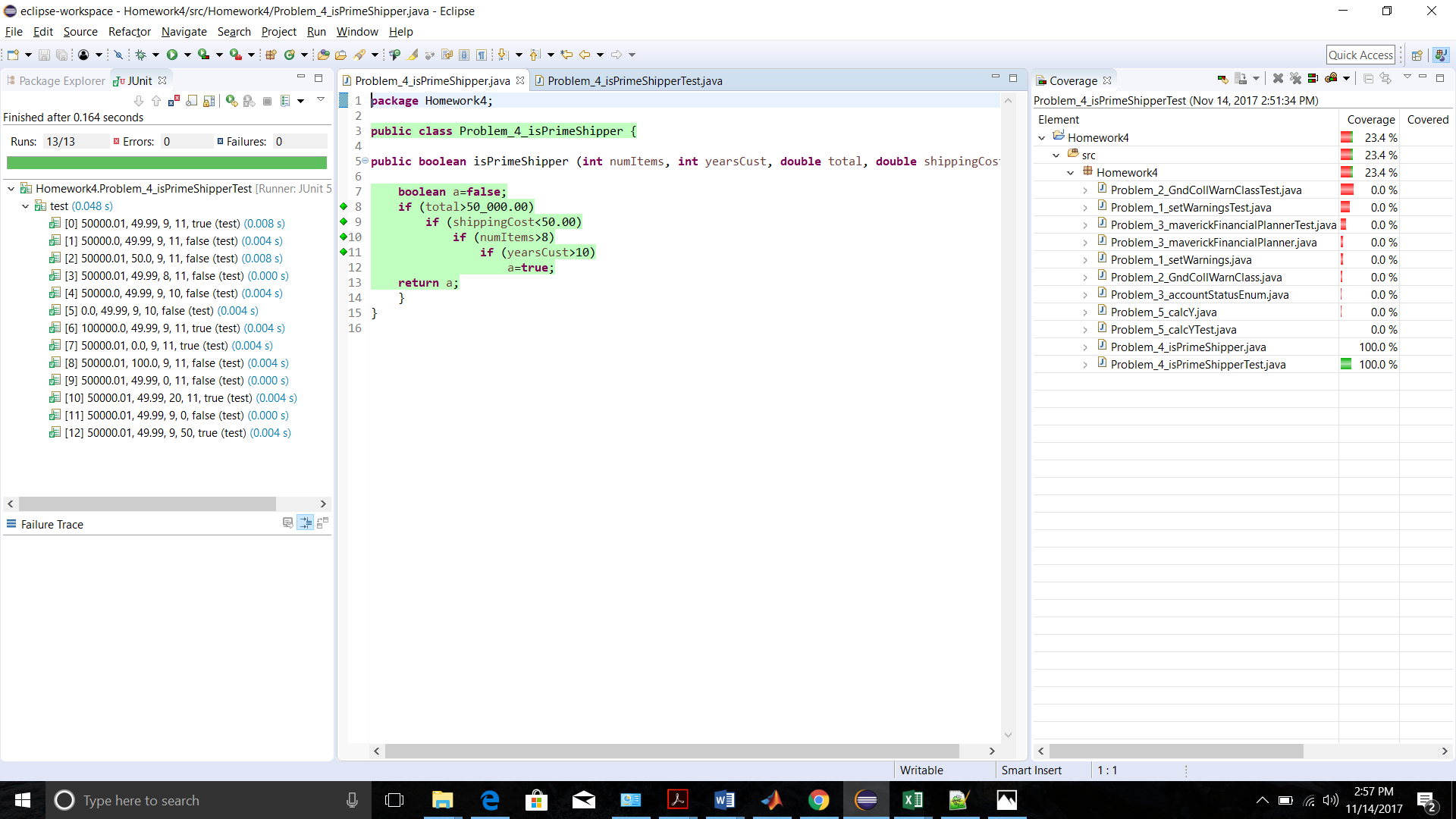
Use the code and the test cases we developed from Homework 3 for Problem 4. Implement this using the **JUnitParamsRunner**. Use the posted class test cases from Homework 3 and test to a threshold of 0.01

Solution4 Following was the table for homework 3 Problem 4 :-

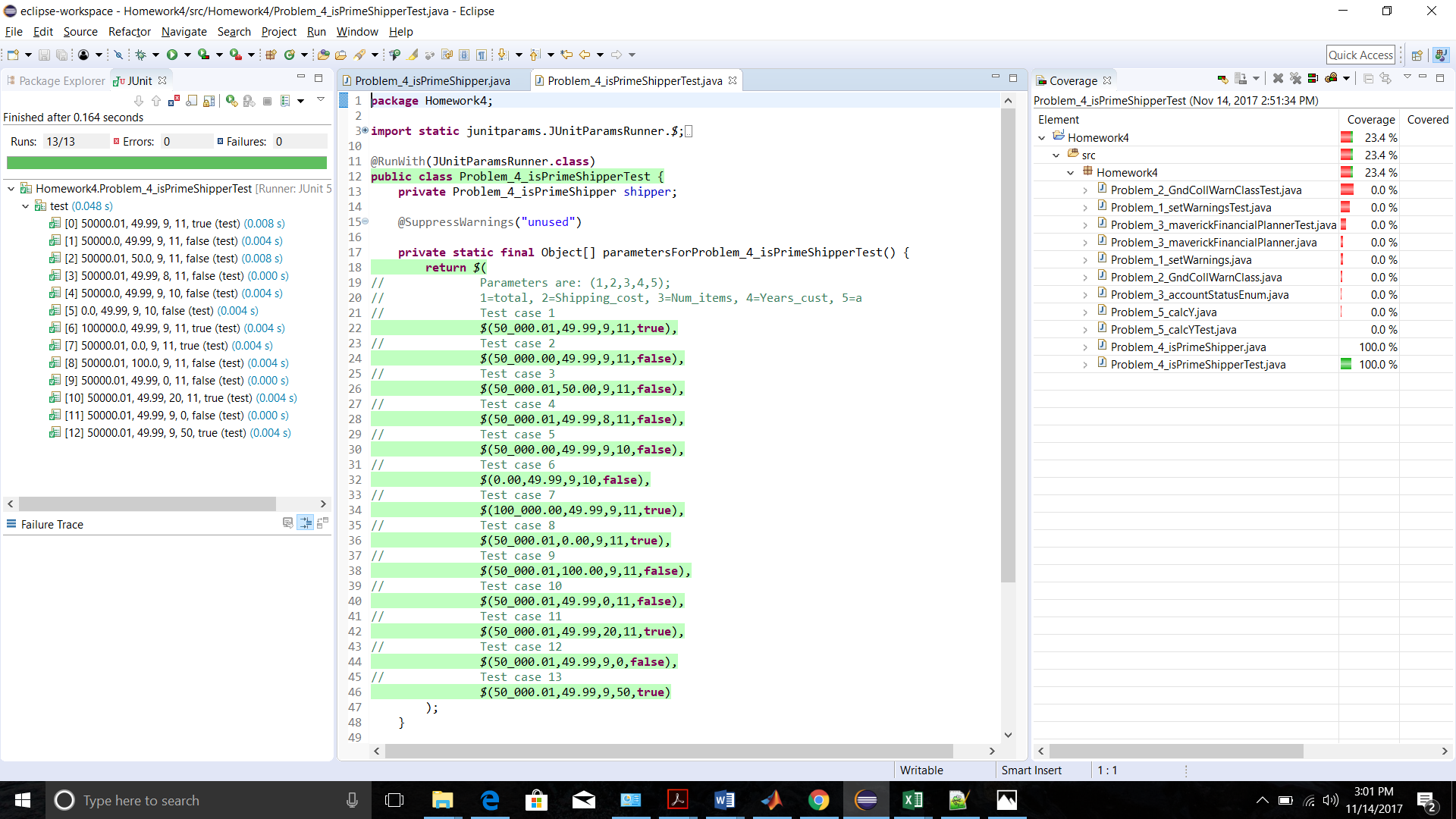


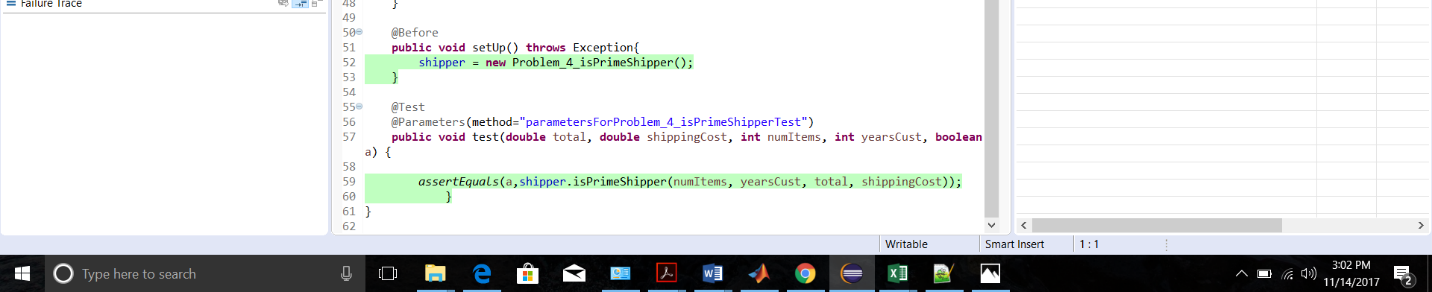
Screen shots for JUNIT and JACOCO coverage :-

Main Java file with Junit green bar showing all the parameters executed and green highlighting showing full JACOCO coverage for the logic.



Junit implementation using **JUnitParamsRunner**





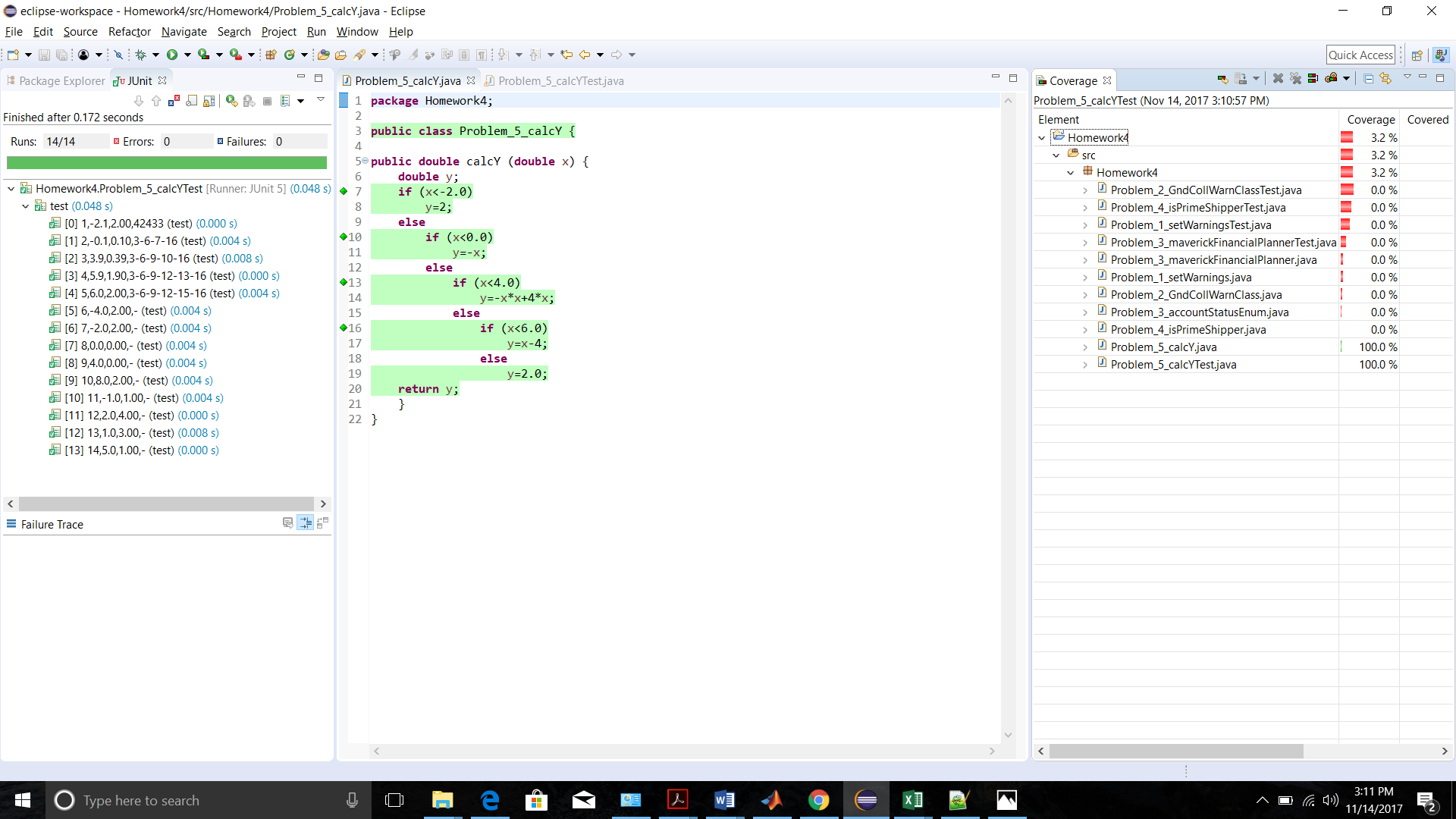
Problem 5 Use the code and the test cases we developed from Homework 3 for Problem 5. Implement this using the **FileParameters** (read the values from a file). Use the posted class test cases from Homework 3 and test to a threshold of 0.01

Solution5 Following was the table for homework 3 Problem 5 :-

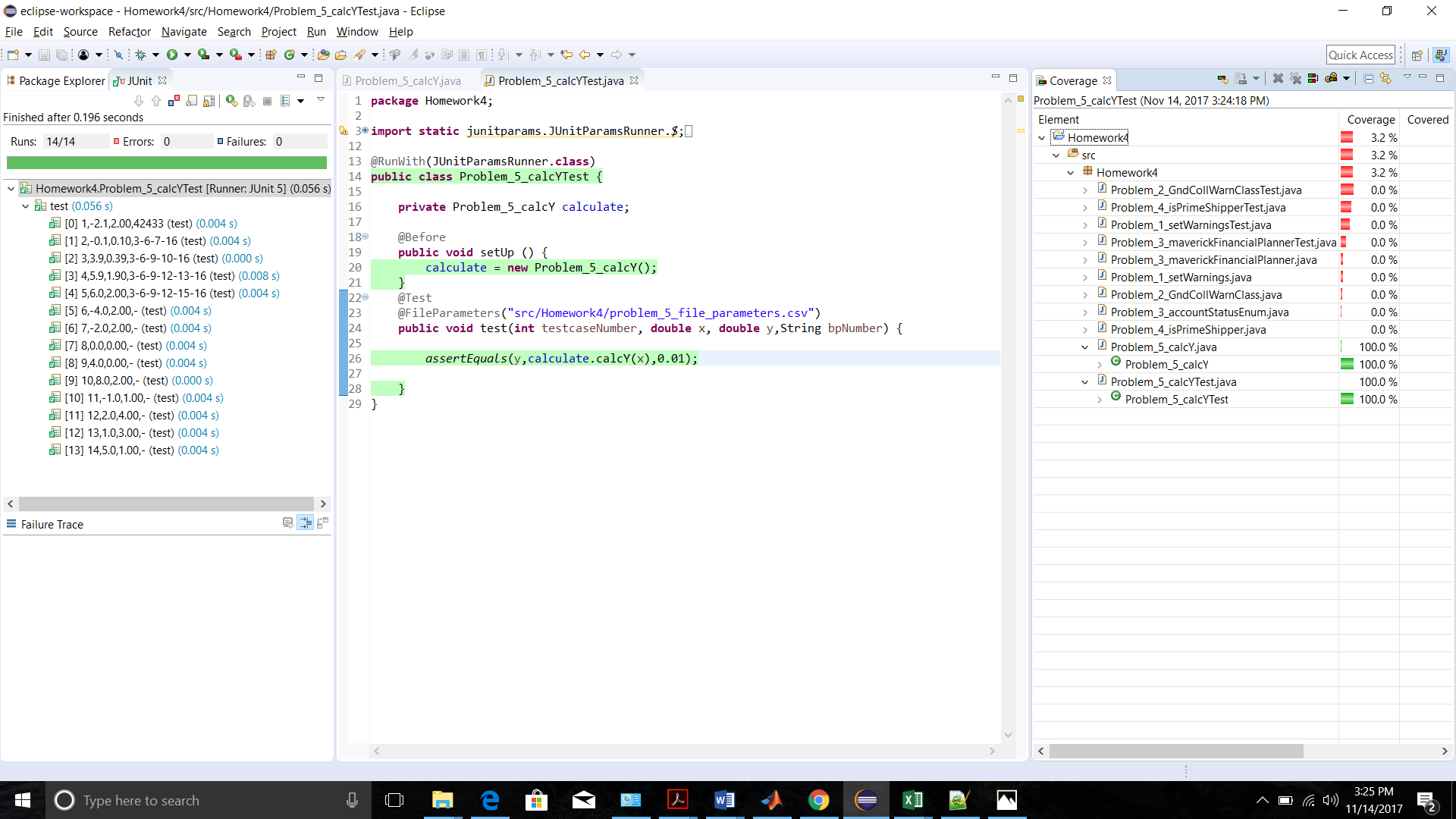


Screen shots for JUNIT and JACOCO coverage :-

Main Java file with Junit green bar showing all the parameters executed and green highlighting showing full JACOCO coverage for the logic.



Junit implementation using **FileParameters**



**CSV table used**

