**Experiment Report - 55 - test2\_ATMValidation**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 10 | Logic | √ | × | × |
| E02 | line 31 | Syntax | √ | √ | √ |
| E03 | line 44 | Syntax | √ | √ | √ |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 100%

Peer 1 Detection Rate: 67%

Peer 2 Detection Rate: 67%

1. **Source Code**
2. package a;
3. import com.code\_intelligence.jazzer.api.FuzzedDataProvider;
4. public class c02\_ATMValidation {
6. public static boolean isValidWithdrawal(double amount, double accountBalance) {
7. boolean flag;
8. if(amount > 0 && accountBalance > 0 && amount < accountBalance) {
9. flag = true;
10. }
11. else flag = false;
12. return flag;
13. }
14. public static void main(String[] args) {
15. // testcase-VT:
16. double[][] testCases = {
17. {2598.16, 1784.78}, //amount > 0 && accountBalance > 0 && amount > accountBalance
18. {197.64, 102.47}, //amount > 0 && accountBalance > 0 && amount > accountBalance
19. {1015.07, 4908.75}, //amount > 0 && accountBalance > 0 && amount <= accountBalance
20. {-1939.66, 2509.62}, //amount <= 0 && accountBalance > 0 && amount <= accountBalance
21. {1707.62, -753.75}, //amount > 0 && accountBalance <= 0 && amount > accountBalance
22. };
24. for (double[] testCase : testCases) {
25. boolean result = isValidWithdrawal(testCase[0], testCase[1]);
26. System.out.printf("isValidWithdrawal(%.1f, %.1f) = %b%n",
27. testCase[0], testCase[1]);
28. }
29. }
30. }
31. class t02 {
32. public static void fuzzerTestOneInput(FuzzedDataProvider data) {
33. // generate 2 random double value
34. double a = data.consumeDouble();
35. double b = data.consumeDouble();
36. // call target method
37. boolean result = c02\_ATMValidation.isValidWithdrawal(a);
38. // System.out.println("Testing with: a=" + a + ", b=" + b + ", c=" + c + " => ");
39. System.out.println("Testing with: a=" + a + ", b=" + b + ", => Result: " + result);
40. }
41. }