CSE 5321 SOFTWARE TESTING

FALL 2015

HOMEWORK 4

Submitted by: Sneha Suhas Chitre

Student Id: 1001244953

**Problem 1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case | Current state | V1 | V2 | a | x | y | V1 | V2 | Next State |
| 1 | Start | -1 | -1 | F | 0 | 0 | 0 | 0 | s0 |
| 2 | s0 | 0 | 0 | T | 1 | 2 | 0 | 1 | s1 |
| 3 | s0 | 0 | 0 | F | 2 | 1 | 0 | 0 | s0 |
| 4 | s1 | 0 | 1 | T | 2 | 2 | 1 | 0 | s2 |
| 5 | s1 | 0 | 1 | F | 2 | 1 | 0 | 1 | s1 |
| 6 | s2 | 1 | 0 | T | 3 | 2 | 1 | 1 | s3 |
| 7 | s2 | 1 | 0 | F | 3 | 1 | 1 | 0 | s2 |
| 8 | s3 | 1 | 1 | T | 3 | 0 | 0 | 1 | s1 |
| 9 | s3 | 1 | 1 | F | 4 | 3 | 1 | 1 | s3 |

**Test Case Table**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Inputs | | | Outputs | | | |  |
| Test Case | Current state | V1 | V2 | a | x | y | V1 | V2 | Next State |
| 1 | Start | -1 | -1 | F | 0 | 0 | 0 | 0 | s0 |
| 2 | s0 | 0 | 0 | T | 1 | 2 | 0 | 1 | s1 |
| 3 | s0 | 0 | 0 | F | 2 | 1 | 0 | 0 | s0 |
| 4 | s1 | 0 | 1 | T | 2 | 2 | 1 | 0 | s2 |
| 5 | s1 | 0 | 1 | F | 2 | 1 | 0 | 1 | s1 |
| 6 | s2 | 1 | 0 | T | 3 | 2 | 1 | 1 | s3 |
| 7 | s2 | 1 | 0 | F | 3 | 1 | 1 | 0 | s2 |
| 8 | s3 | 1 | 1 | T | 3 | 0 | 0 | 1 | s1 |
| 9 | s3 | 1 | 1 | F | 4 | 3 | 1 | 1 | s3 |

**Java Code for StateTableClass**

**package** Problem\_1;

**public** **class** StateTableClass {

**public** **void** nextState (**int** v1,**int** v2, **boolean** a, StateTableOutputData outputs)

{

**switch** (v1\*2 + v2){

**case** 0: **if** (a)

{

outputs.setX(1);

outputs.setY(2);

outputs.setV1(0);

outputs.setV2(1);

System.***out***.println("State: S1");

}

**else**

{

outputs.setX(2);

outputs.setY(1);

outputs.setV1(0);

outputs.setV2(0);

System.***out***.println("State: S0");

}

**break**;

**case** 1: **if** (a)

{

outputs.setX(2);

outputs.setY(2);

outputs.setV1(1);

outputs.setV2(0);

System.***out***.println("State: S2");

}

**else**

{

outputs.setX(2);

outputs.setY(1);

outputs.setV1(0);

outputs.setV2(1);

System.***out***.println("State: S1");

}

**break**;

**case** 2: **if** (a)

{

outputs.setX(3);

outputs.setY(2);

outputs.setV1(1);

outputs.setV2(1);

System.***out***.println("State: S3");

}

**else**

{

outputs.setX(3);

outputs.setY(1);

outputs.setV1(1);

outputs.setV2(0);

System.***out***.println("State: S2");

}

**break**;

**case** 3: **if** (a)

{

outputs.setX(3);

outputs.setY(0);

outputs.setV1(0);

outputs.setV2(1);

System.***out***.println("State: S1");

}

**else**

{

outputs.setX(4);

outputs.setY(3);

outputs.setV1(1);

outputs.setV2(1);

System.***out***.println("State: S3");

}

**break**;

**default**:

System.***out***.println("State: Start State");**break**;

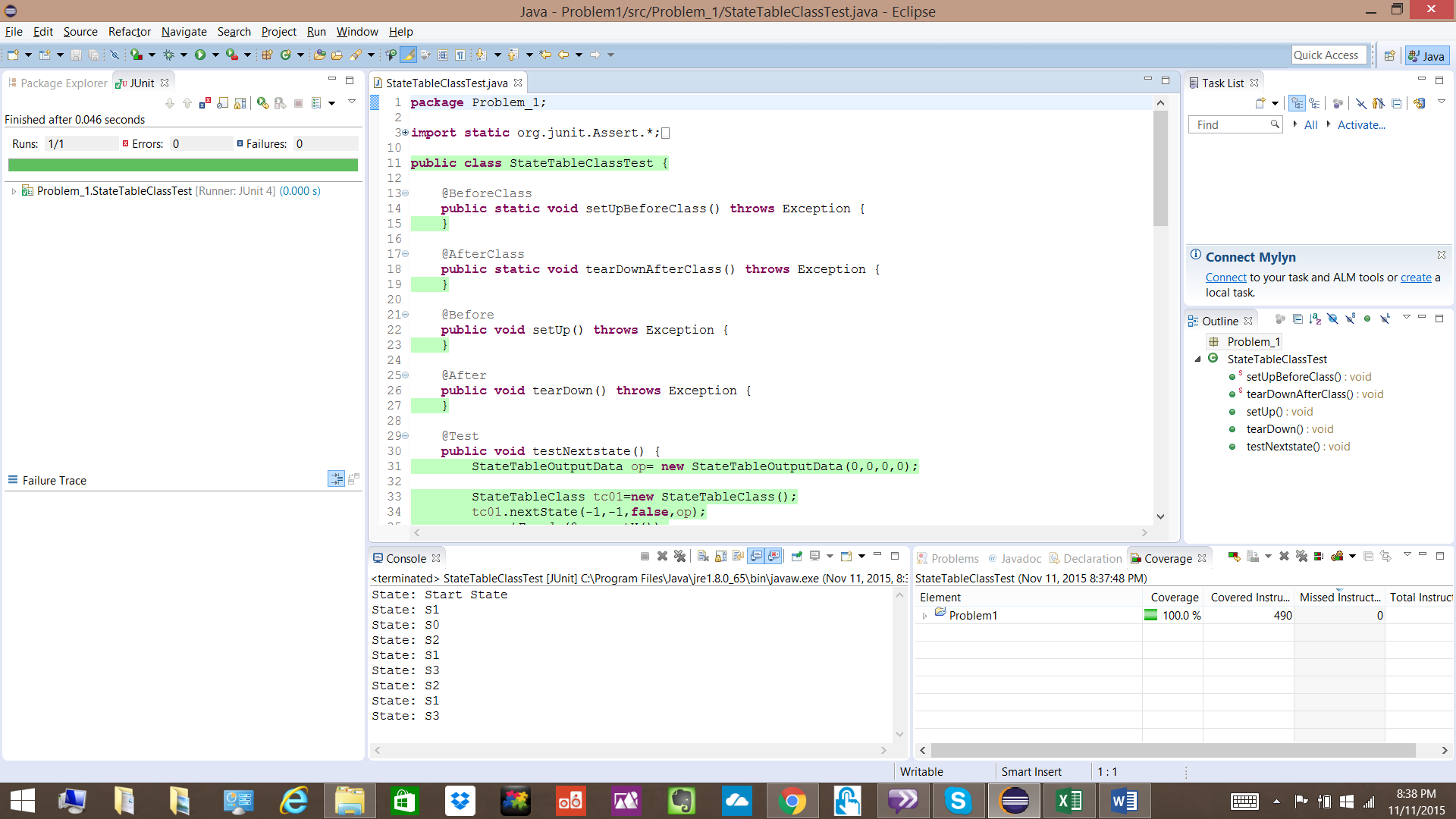
}

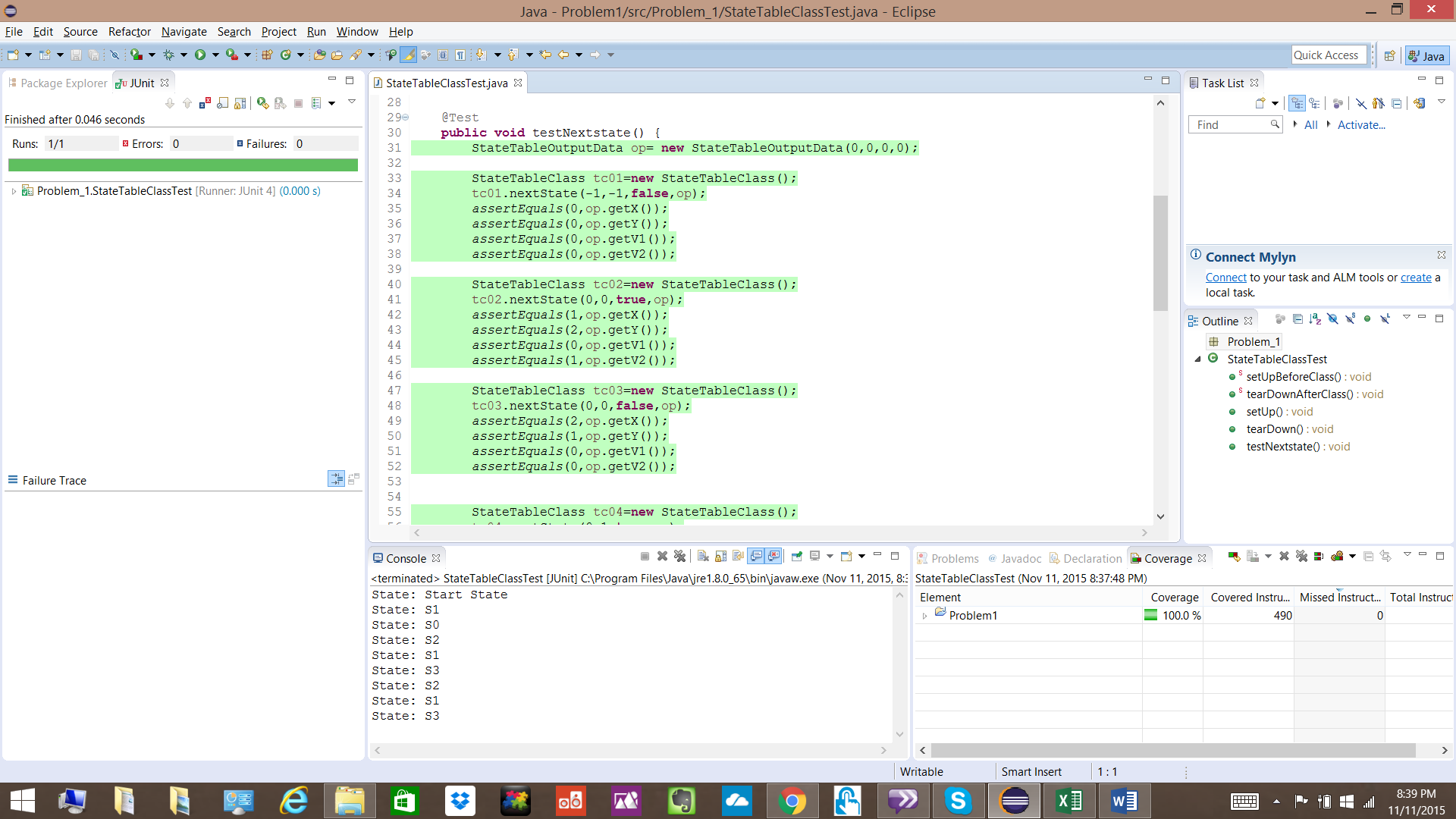
}

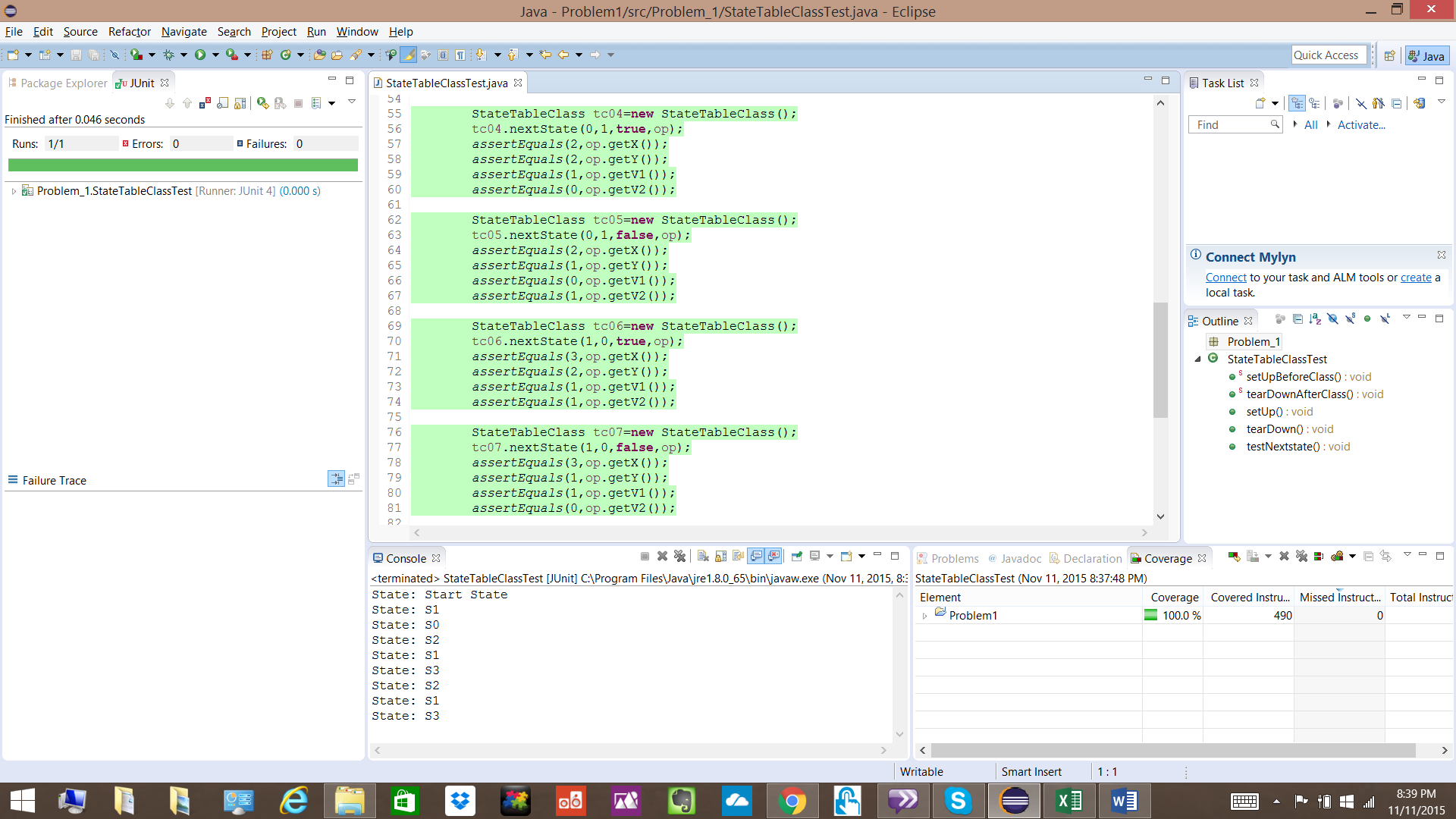
}

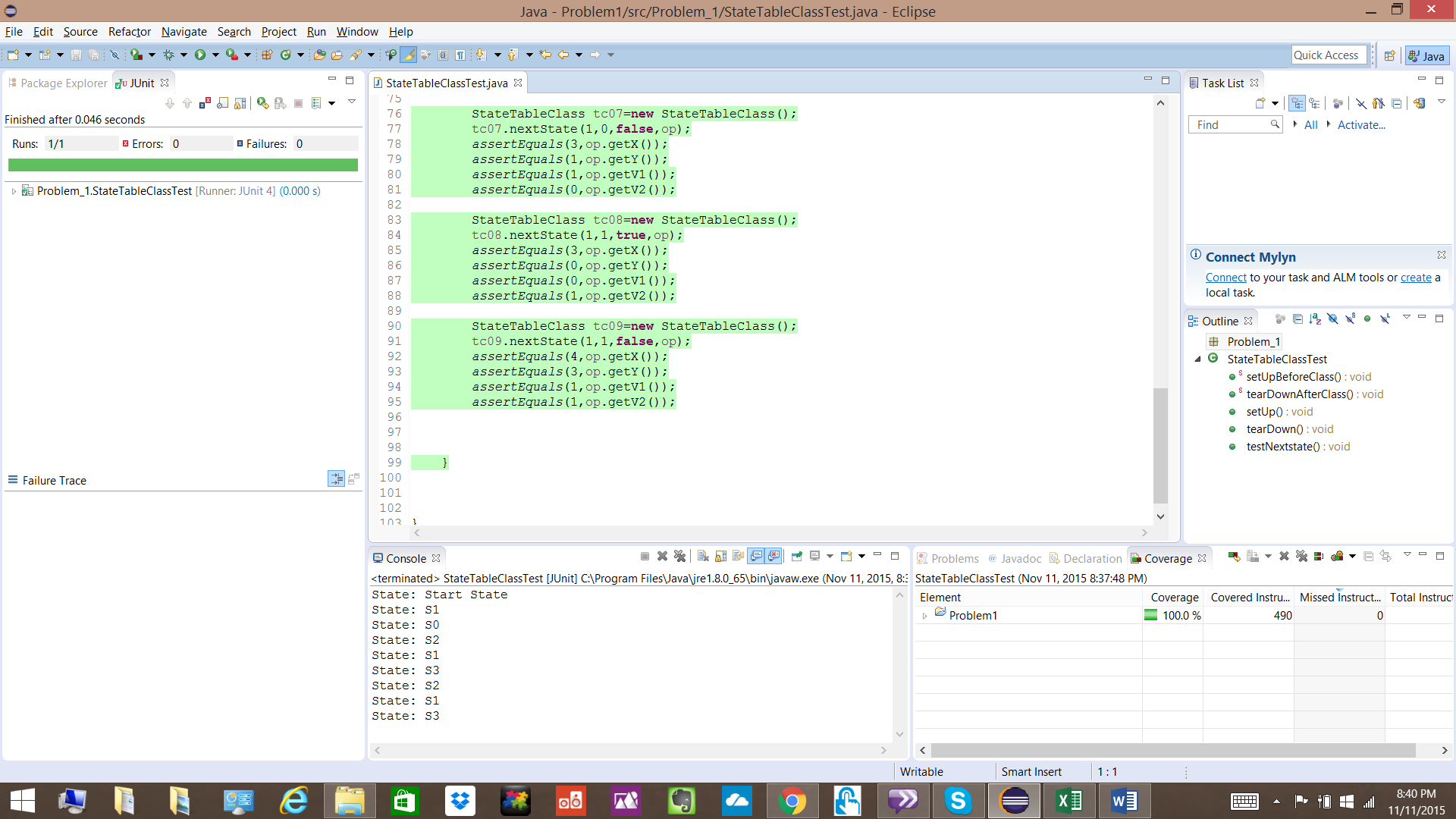
**Junit Test Cases and JACOCO coverage snapshots are as given below.**

**StateTableClassTest.java**

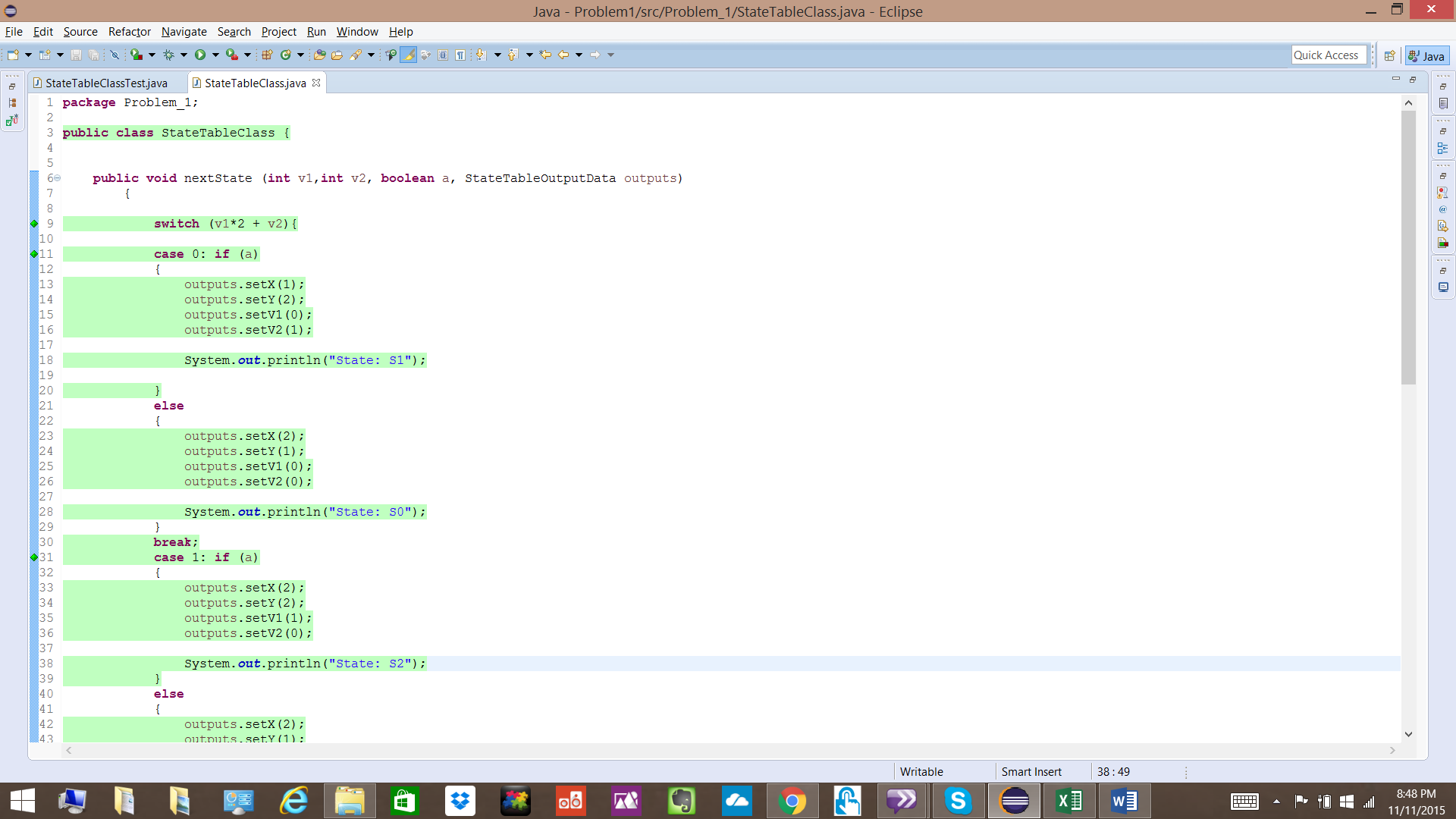


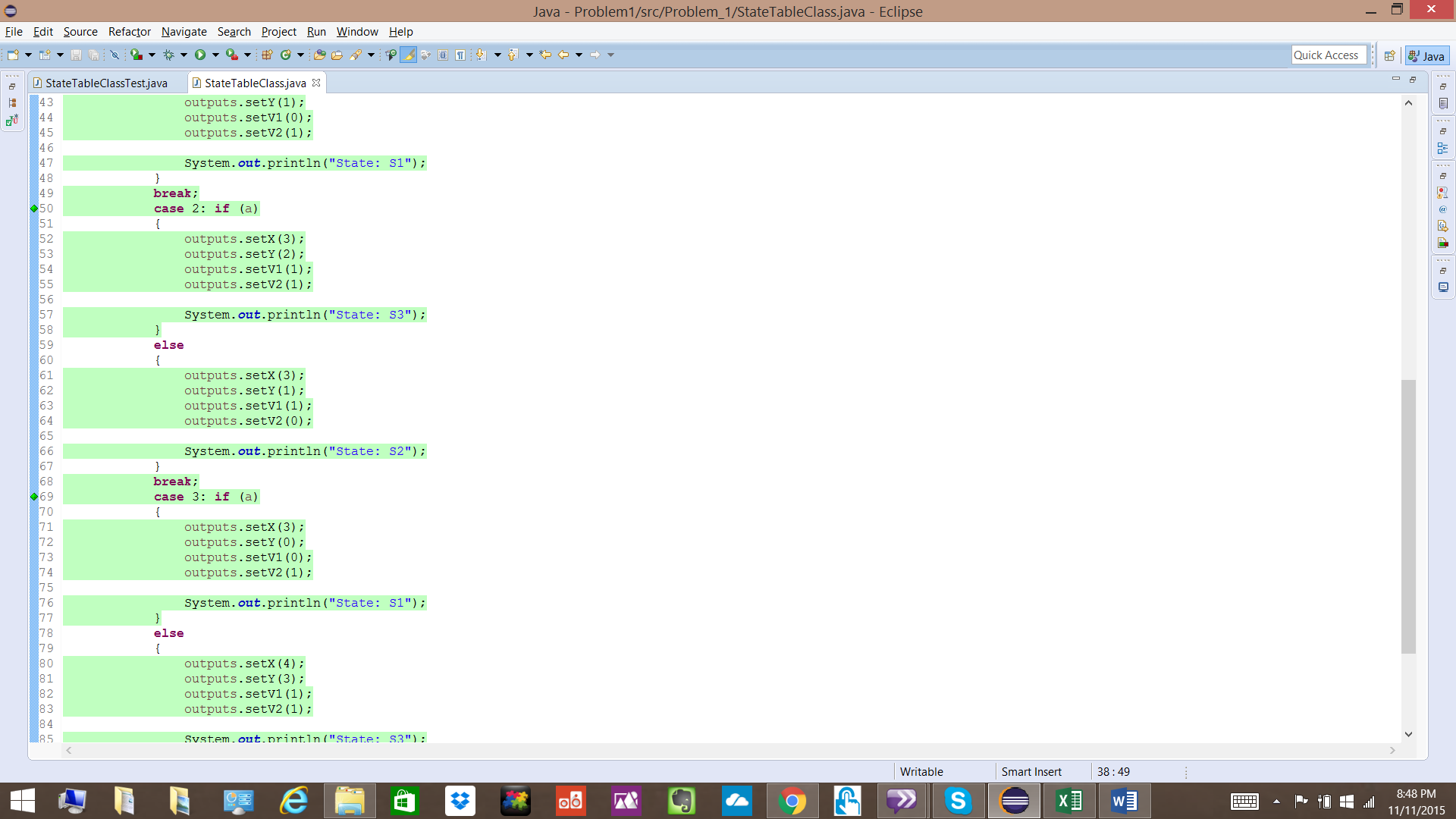


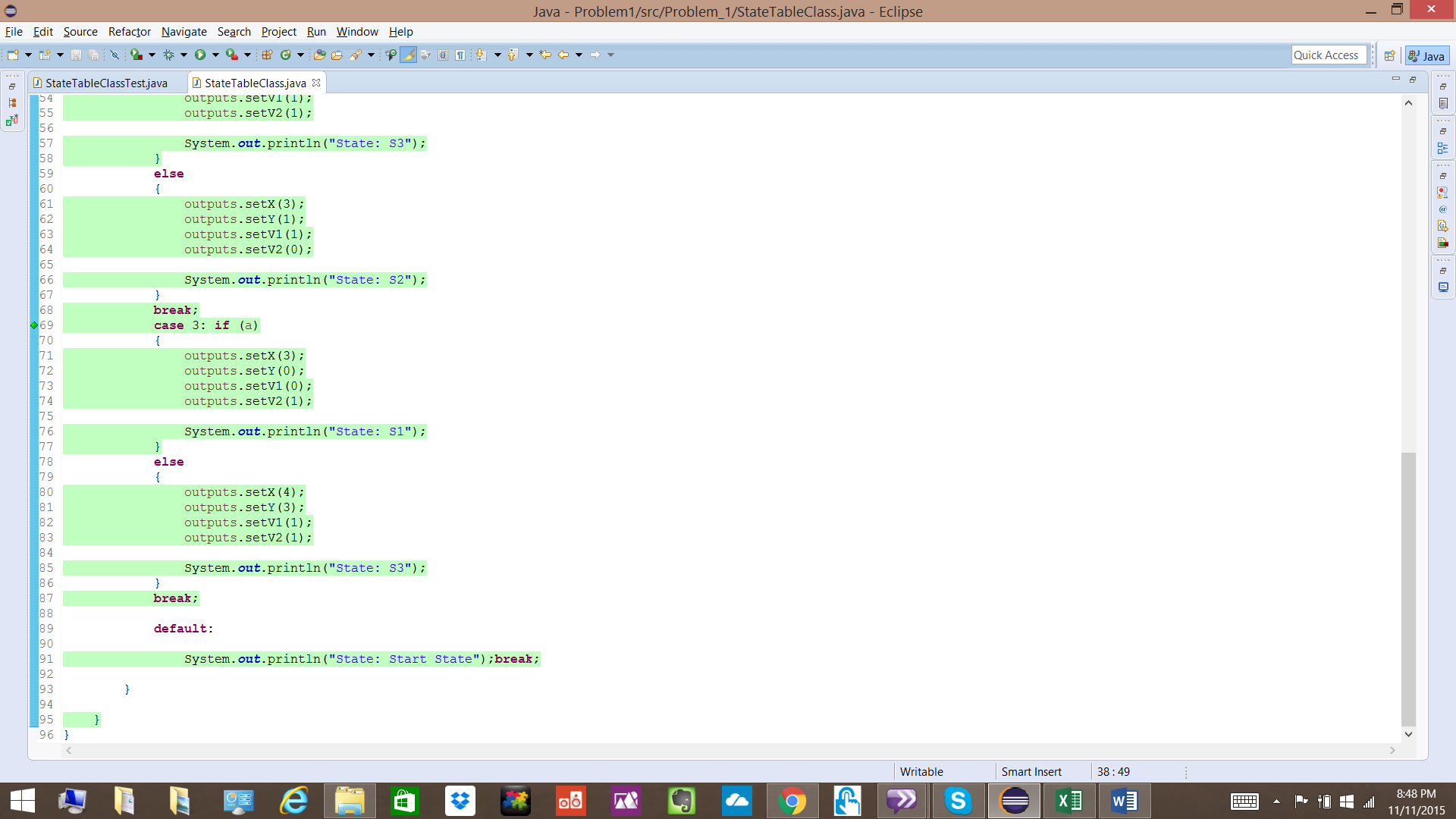




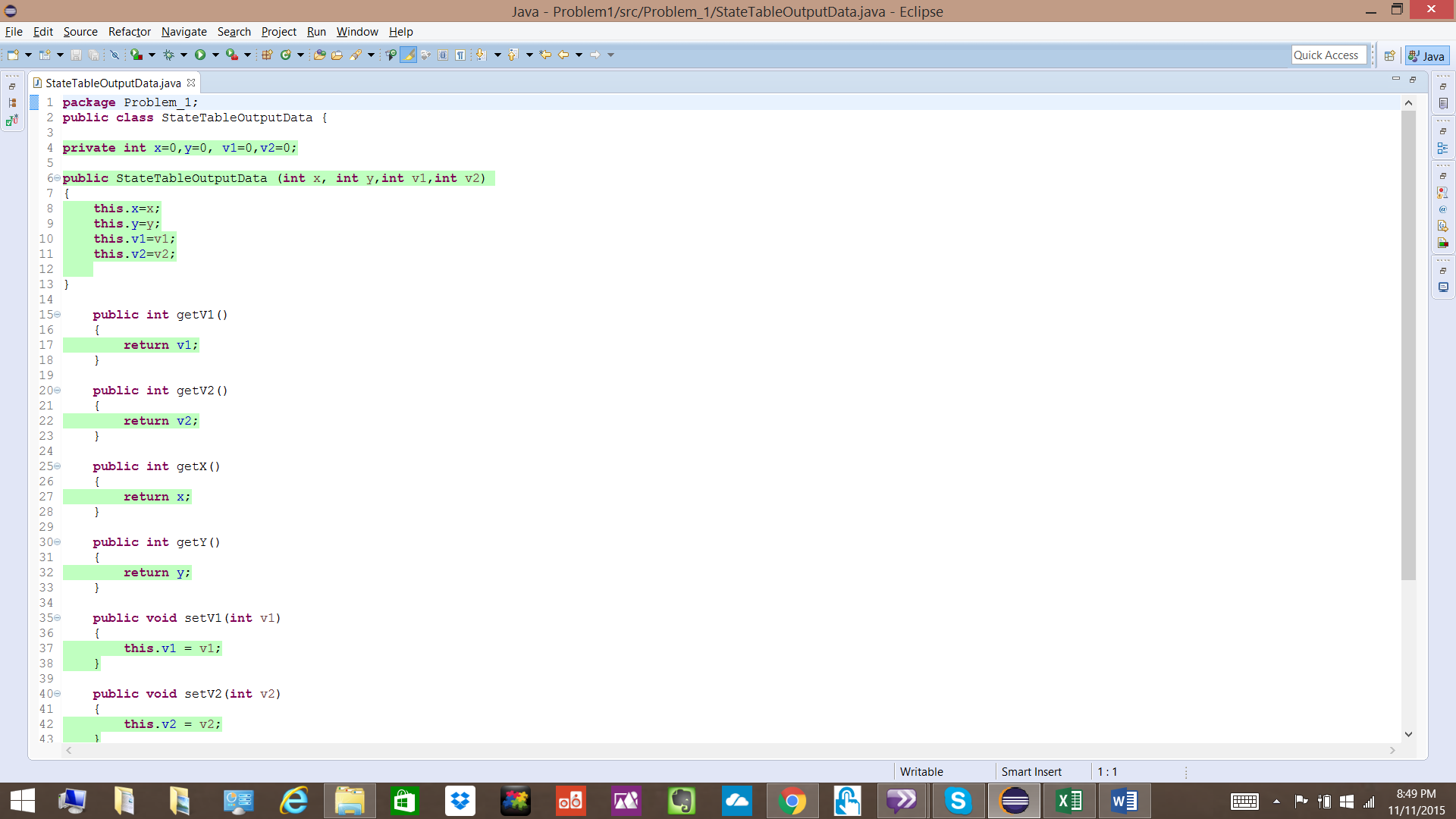
**StateTableClass.java**

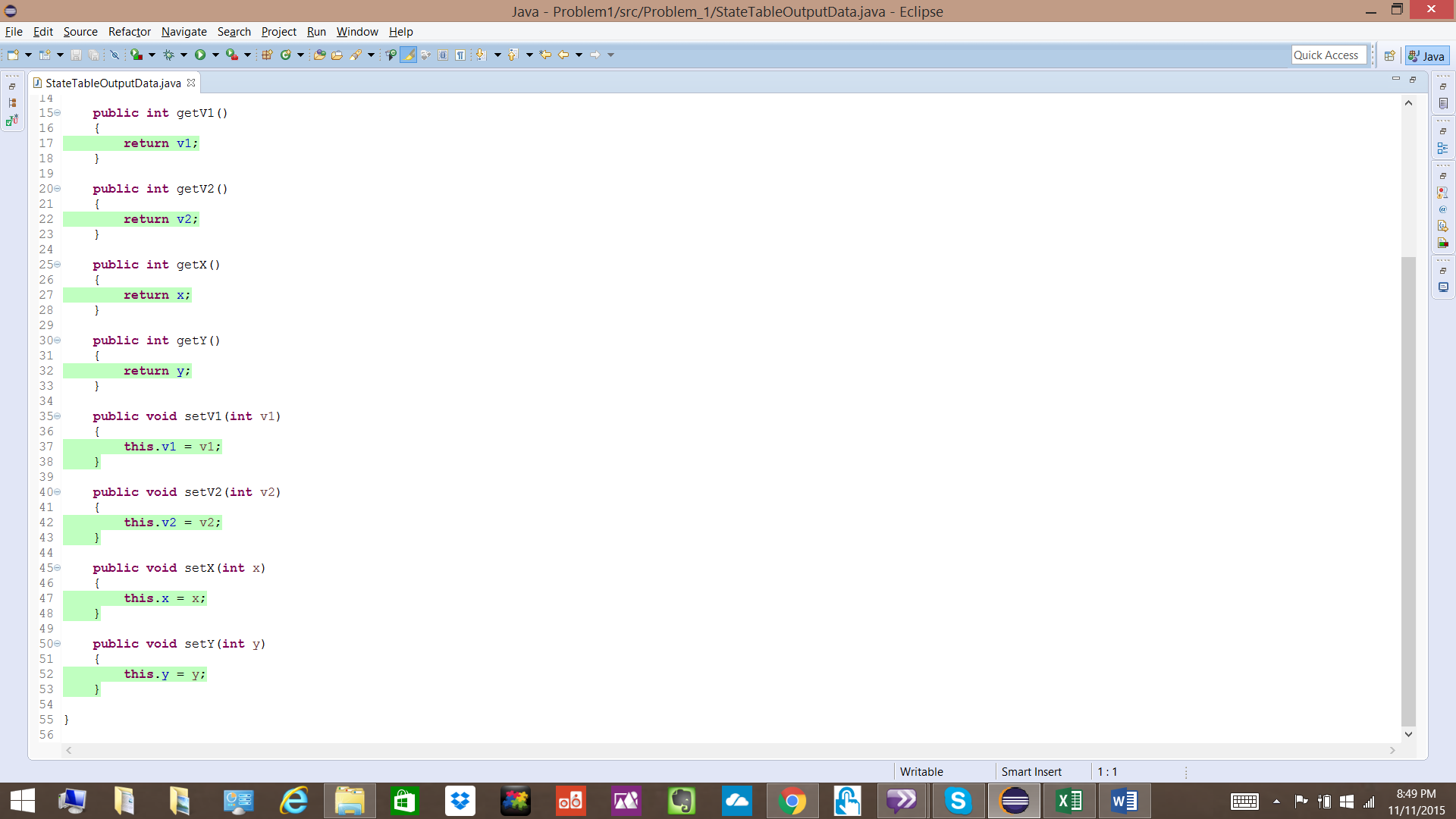






**StateTableOutputData.java**





**Problem 2**

**Given Code:**

package Problem\_2;

public class Problem2Class {

public int numPos (int[] nums) {

int Count = 0;

for (int i = nums.length-1; i > 0; i--)

if (nums[i] > 0)

Count++;

return Count;

}

}

Above program counts the number of integers greater than zero (positive integers) in any given input array.

There is a problem in for loop of above code. I > 0 condition does not iterate through every element of the array missing the element array index 0 element. Hence the number of positive integers is calculated one less.

**Corrected Code is given here as well as attached.**

**package** Problem\_2;

**public** **class** Problem2Class {

**public** **int** numPos (**int**[] nums) {

**int** Count = 0;

**for** (**int** i = nums.length-1; i >= 0; i--)

**if** (nums[i] > 0)

Count++;

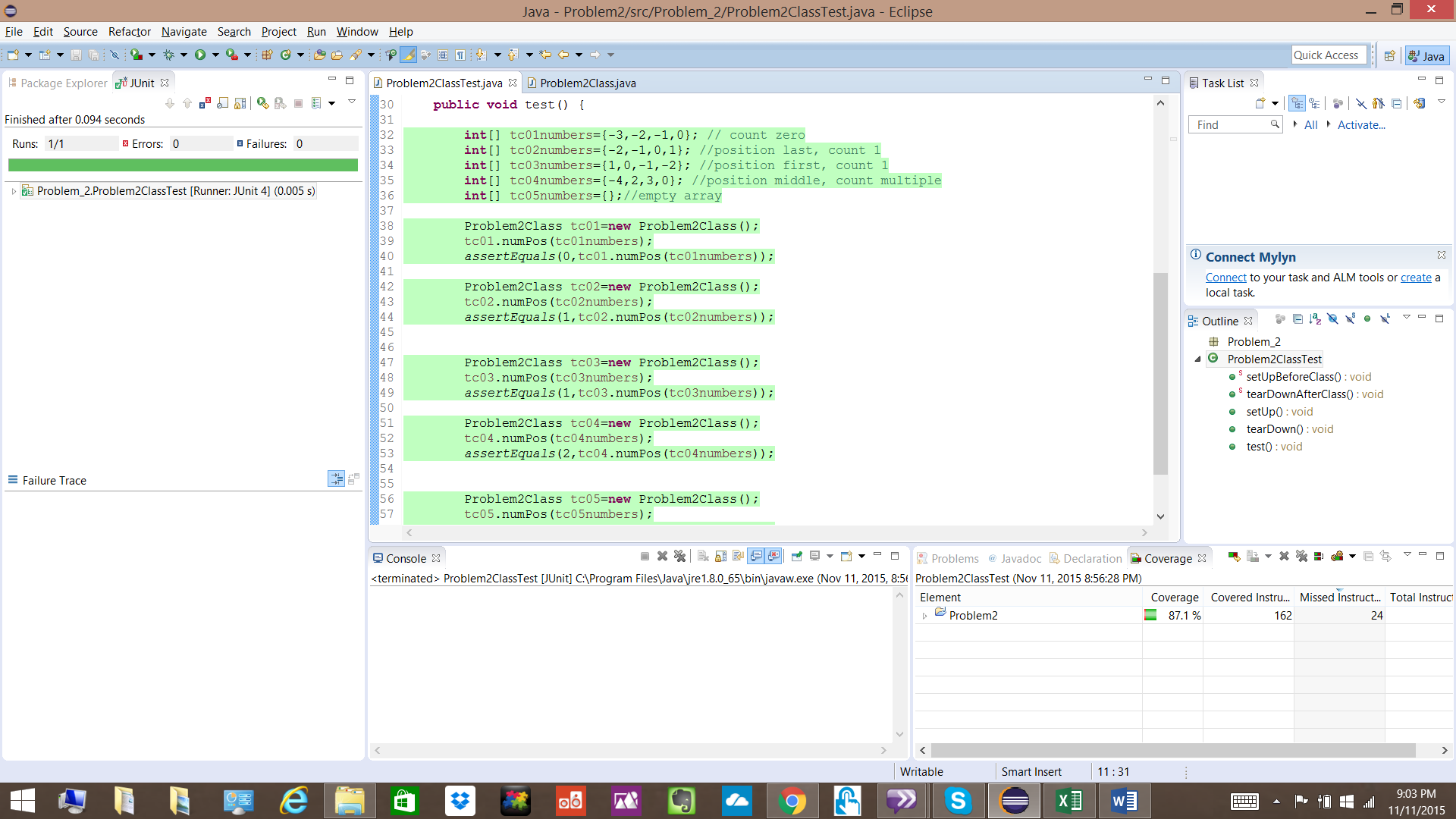
**return** Count;

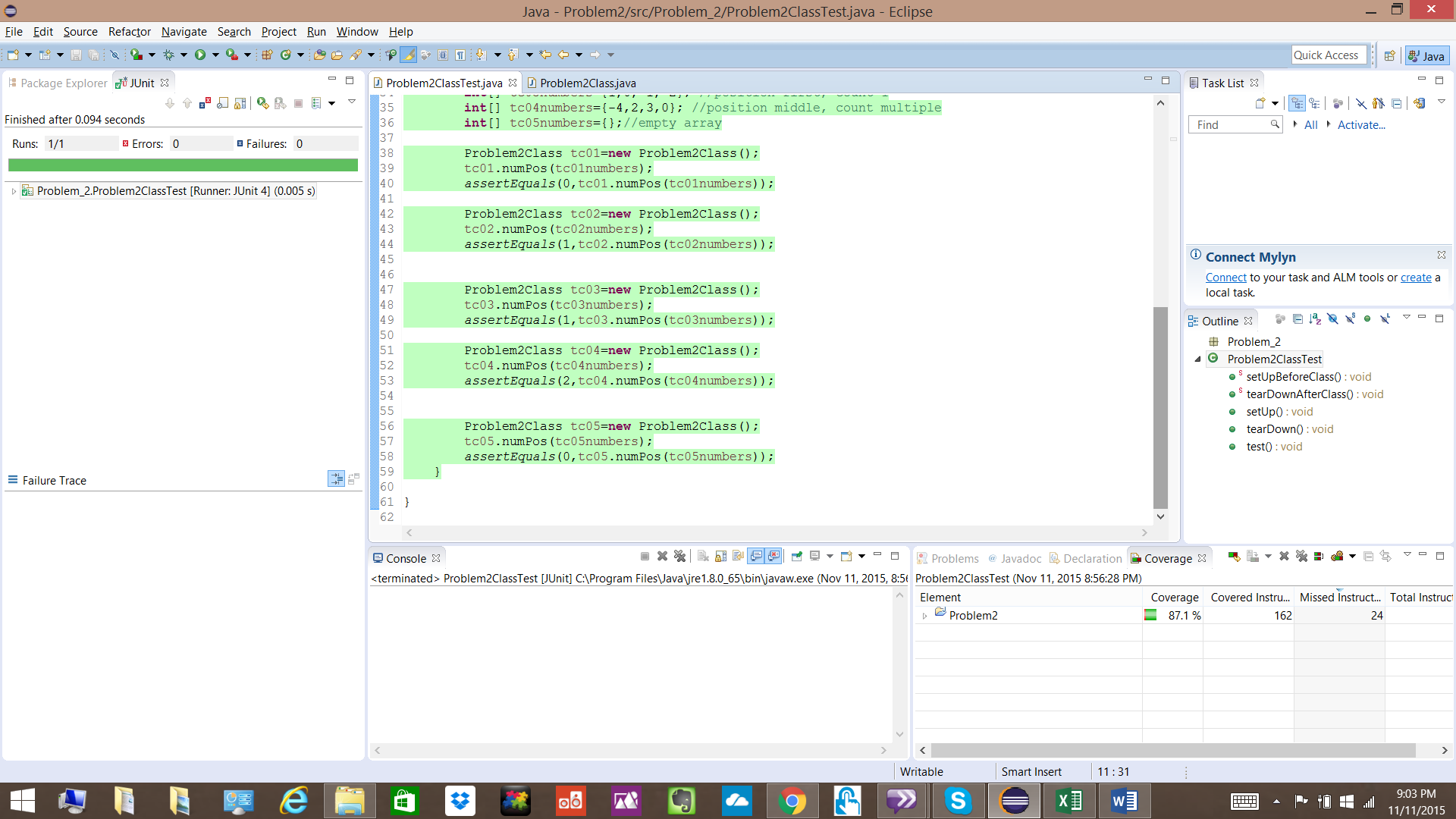
}

}

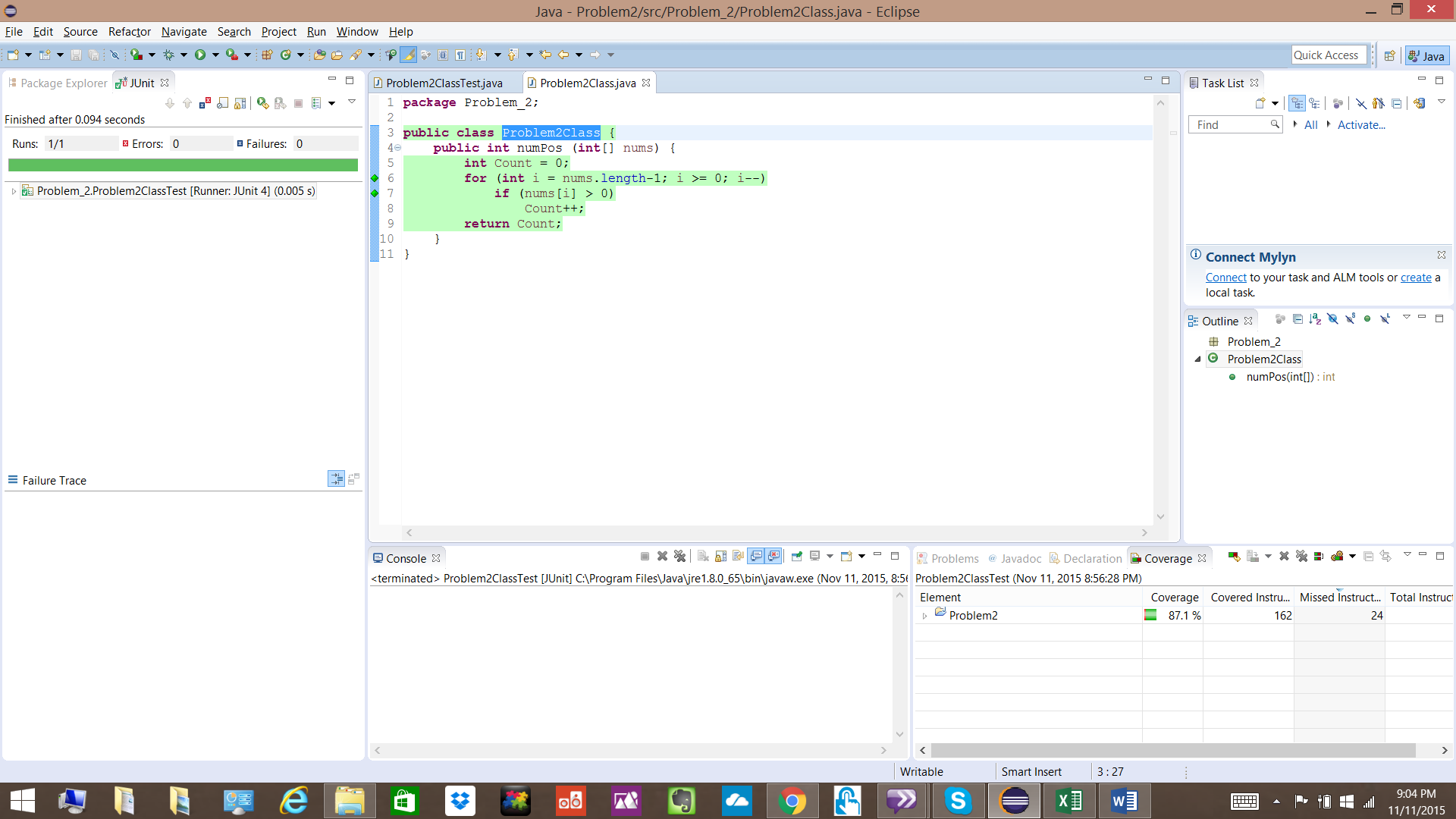
**Junit and JACOCO coverage snapshots as below.**

**Problem2ClassTest.java**

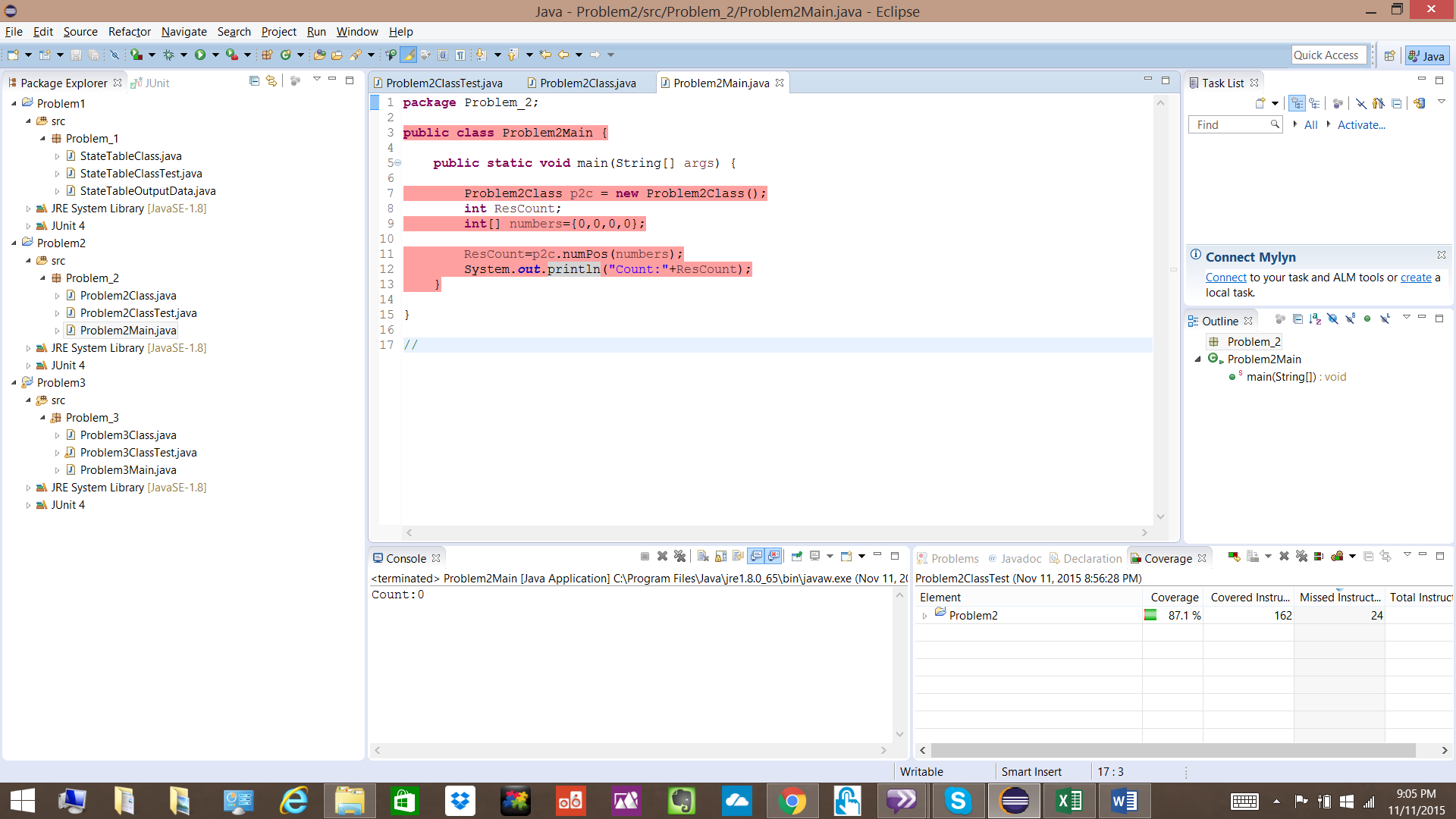




**Problem2Class.java**



**Problem2Main.java**



**Problem 3**

**Given Code:**

package Problem\_3;

public class Problem3Class {

public boolean isSorted (int[] nums) {

// this method determines if an integer array of

// any length is sorted in an ascending manner

boolean sorted=nums.length>0;

for (int i = nums.length-1; i > 0 & sorted; i--)

if ((nums[i] - nums[i-1]) < 0)

sorted=false;

return sorted;

}

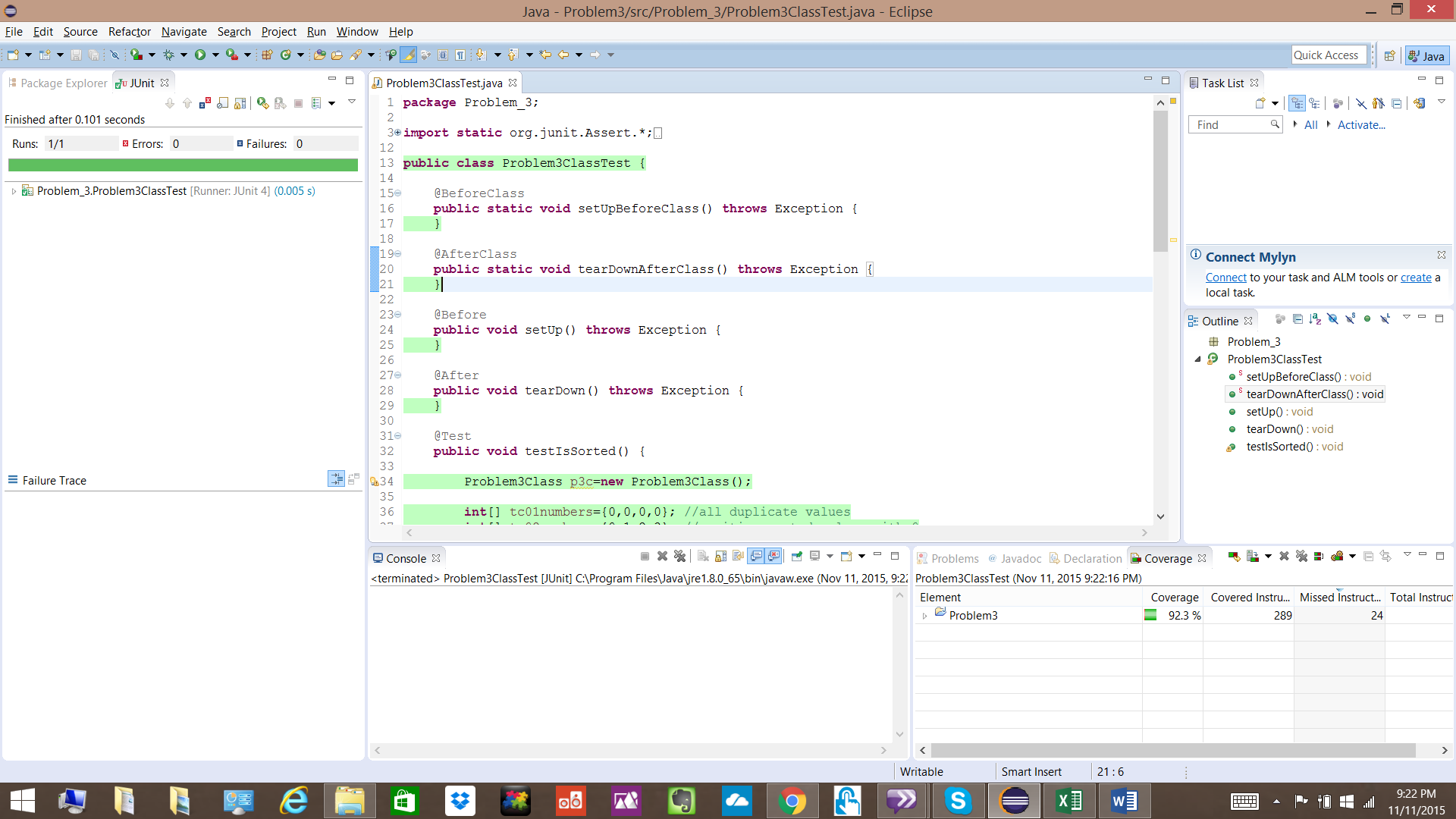
}

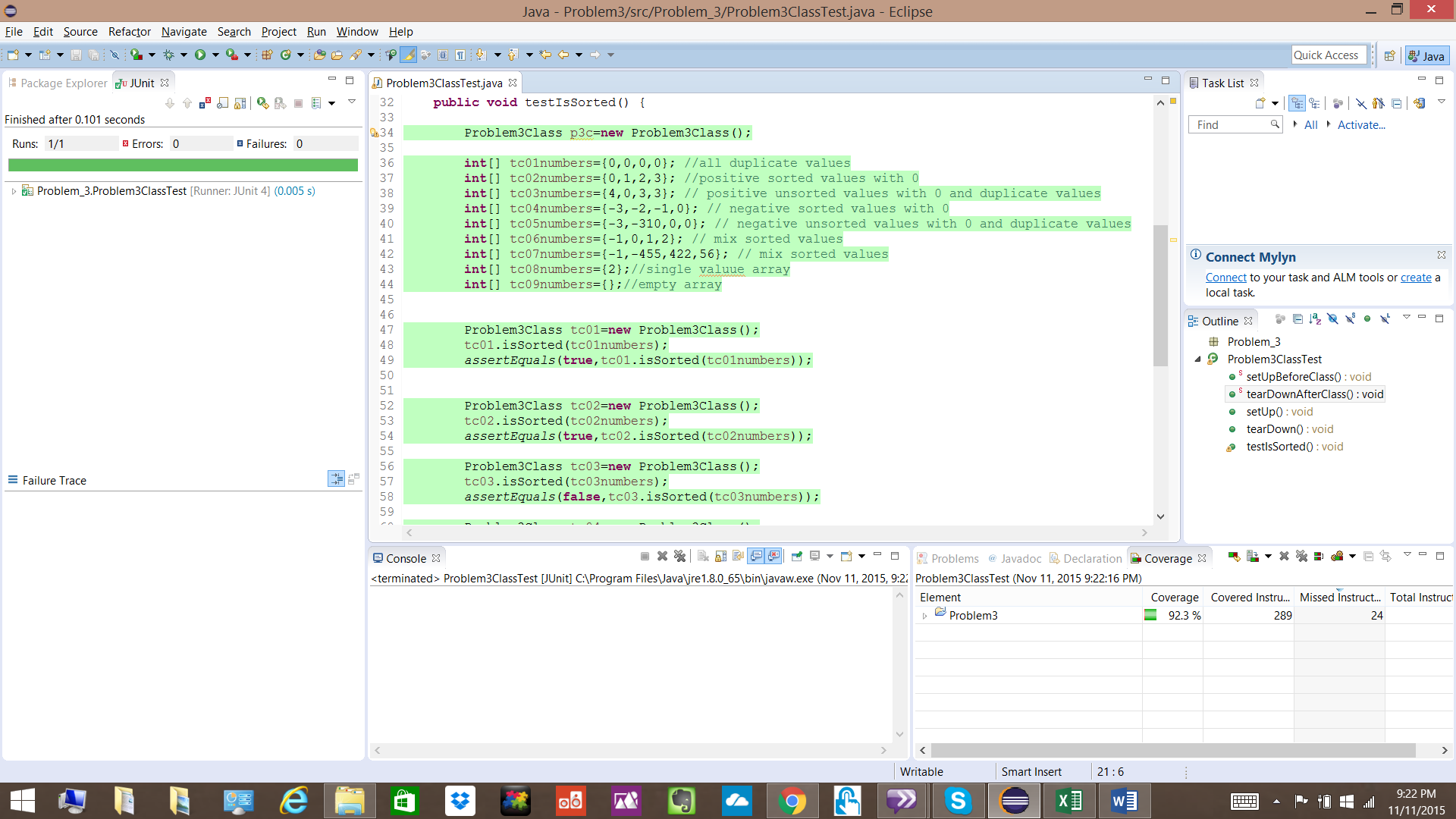
Given code checks if the input array of any length is sorted in ascending order or not.

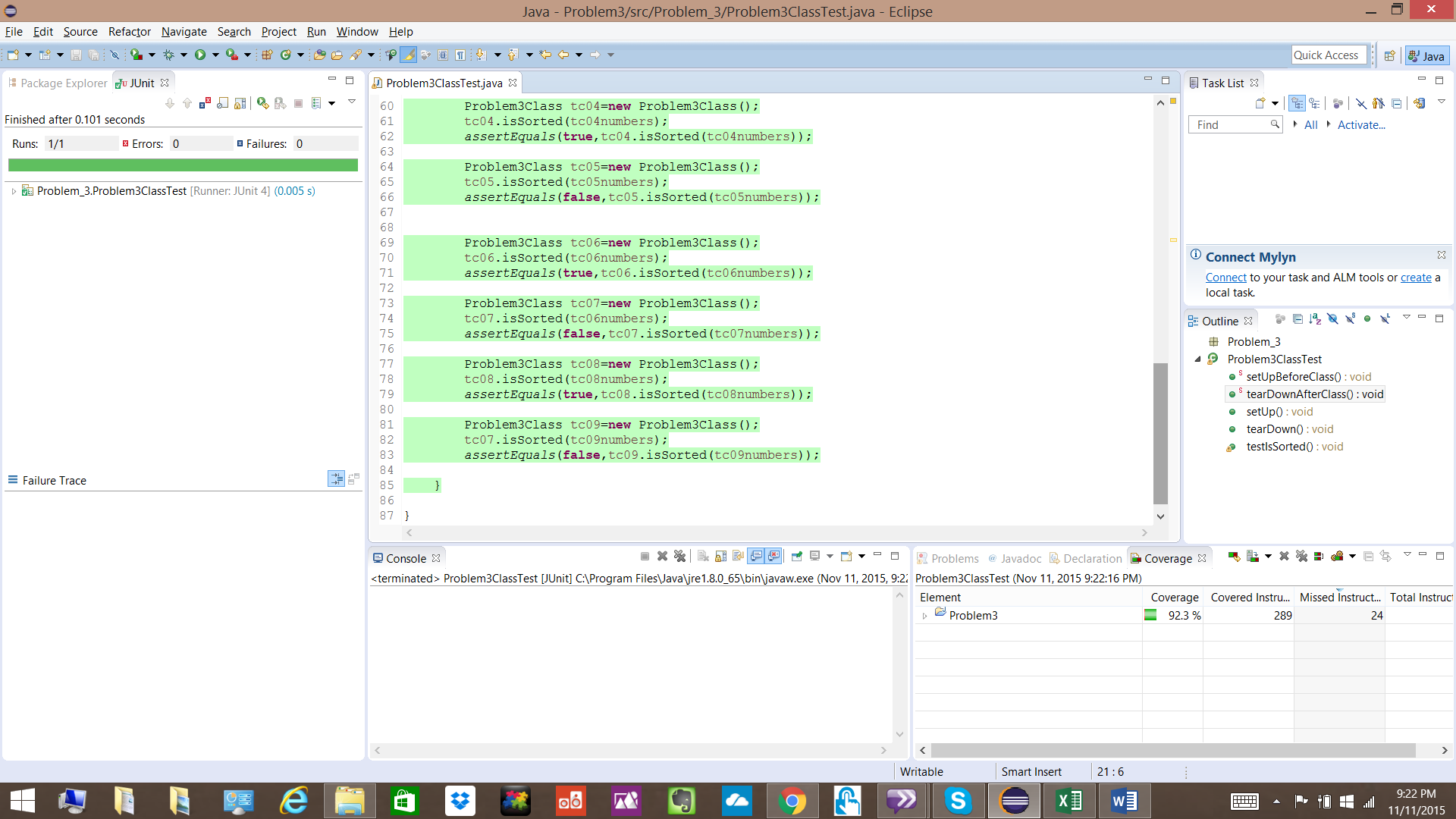
There is no problem with the code. Code runs correct for various test case inputs as seen in the Junit test code below.

**Junit and JACOCO coverage snapshots as below.**

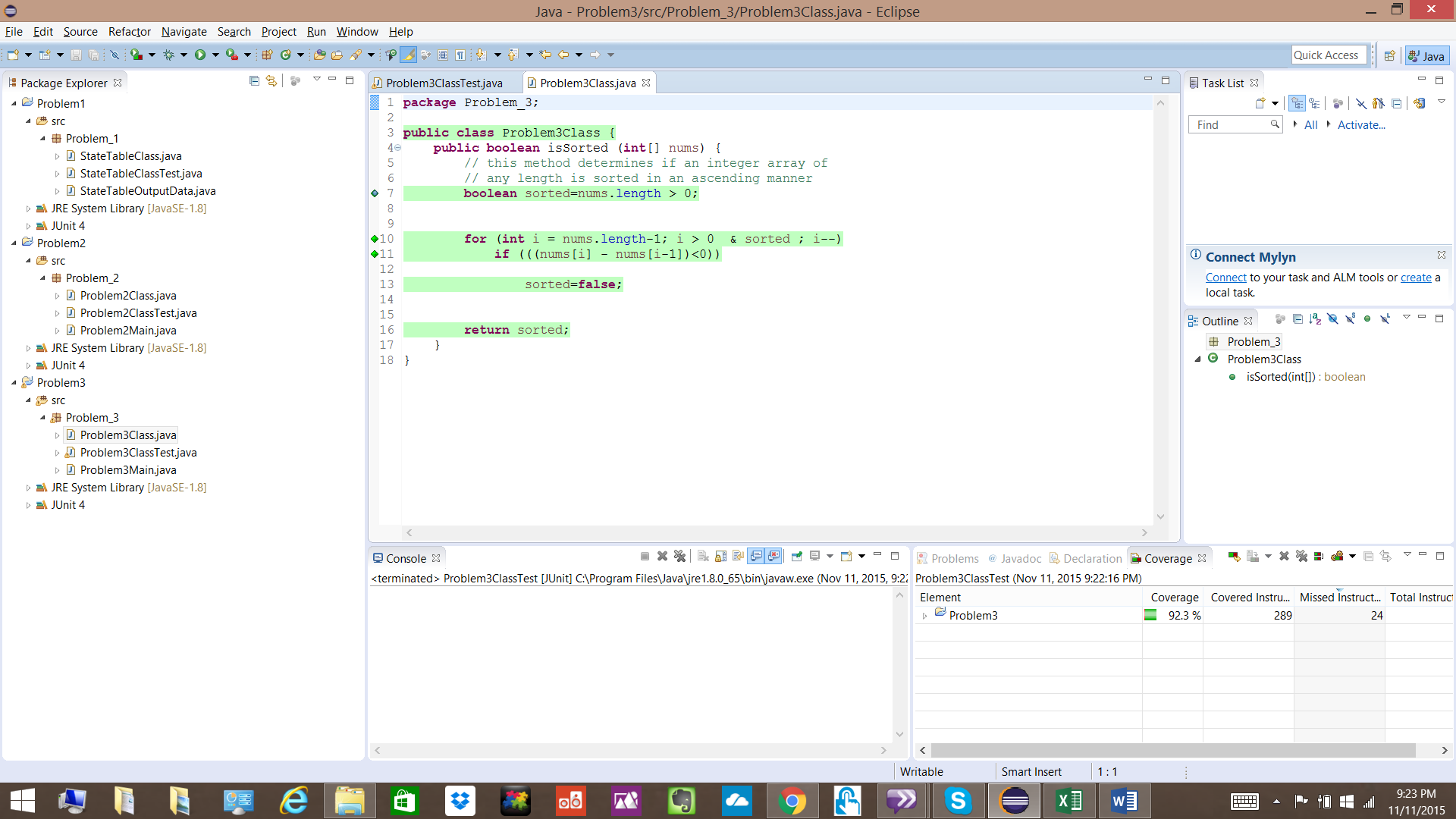
**Problem3ClassTest.java**







**Problem3Class.java**



**Problem3Main.java**

