

Last TP software testing

zakariya Sabiri

November 2025

The screenshot shows a Java development environment with the following details:

- Code Editor:** The main window displays the `CalculatorTest.java` file. It contains a single test method:

```
class CalculatorTest {  
    @Test  
    void add_twoPositiveNumbers_shouldReturnSum() {  
        // Arrange  
        Calculator calc = new Calculator();  
        // Act  
        int result = calc.add(2, 3);  
        // Assert  
        assertEquals(5, result, "message" + 2 + 3 should equal 5 + );  
    }  
}
```
- Run Window:** Below the editor, a "Run" window is open, showing the results of the test run:

Test	Time	Status
CalculatorTest	83 ms	1 test passed
addTwoPositiveNumbers	83 ms	✓

The "addTwoPositiveNumbers" test is highlighted with a blue selection bar.

Figure 1: Step 5

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows files Main.java, Calculator.java, and CalculatorTest.java.
- CalculatorTest.java:** The active file contains a single test method:

```
20 class CalculatorTest {  
21     @Test  
22     void add_twoPositiveNumbers_shouldReturnSum() {  
23         // Arrange  
24         Calculator calc = new Calculator();  
25         // Act  
26         int result = calc.add(a: 2, b: 3);  
27         // Assert  
28         assertEquals(expected: 5, result, message: "2 + 3 should equal 5");  
29     }  
30 }
```
- Run Tab:** Shows a failed test run:
 - CalculatorTest**: 78 ms
 - add_twoPositiveNum**: 78 ms
 - 1 test failed**: 1 test total, 78 ms
- Output Tab:** Displays the failure details:

```
org.opentest4j.AssertionFailedError: 2 + 3 should equal 5 ==>  
Expected :5  
Actual   :-1  
<Click to see difference>
```
- Bottom Status Bar:** Shows the path: TP_soft_Units_test > src > test > java > CalculatorTest > add_twoPositiveNumbers_shouldReturnSum

Figure 2: step 6

The screenshot shows a Java code editor interface. The title bar displays "TP TP_soft_Unite_test" and "Version control". The tabs at the top are "Main.java", "Calculator.java" (which is selected), and "CalculatorTest.java". The code editor pane contains the following Java code:

```
1 public class Calculator { 2     /* 3         Returns the sum of two integers . 4     */ 5     public int add ( int a , int b ) { 6         return a + b; 7     } 8     public int sub(int a,int b){ no usages 9         return a-b; 10    } 11    public int multiply(int a,int b){ no usages 12        return a*b; 13    } 14    public double divide(int a,int b){ no usages 15        return a / b; 16    } 17 }
```

The code editor features dark-themed syntax highlighting. A vertical toolbar on the left includes icons for file operations (New, Open, Save, Find, Replace, Delete, Undo, Redo). The status bar at the bottom shows the path "TP_soft_Unite_test > src > main > java > Calculator > add".

Figure 3: Step 7

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** TP soft_United_test
- File List:** Main.java, Calculator.java, CalculatorTest.java (selected)
- Code Editor:** Content of CalculatorTest.java:

```
5  class CalculatorTest {  
6      @Test  
7      void testdivide() {  
8          Calculator calc = new Calculator () ;  
9          int a = 5;  
10         int b = 0;  
11         assertNull(b);  
12         double result = calc.divide(a,b) ;  
13         assertEquals ( expected: 0 , result , message: " 5/0 " ) ;  
14     }  
15 }
```
- Run Tab:** Run, CalculatorTest x, CalculatorTest x
- Toolbars:** Standard icons for run, stop, refresh, etc.
- Bottom Status Bar:** TP_soft_United_test > src > test > java > CalculatorTest > testdivide
- Test Results:** A failed test summary:
 - CalculatorTest (70ms)
 - testdivide() (70ms)
 - 1 test failed 1 test total, 70ms
 - Exception details:

```
org.opentest4j.AssertionFailedError:  
Expected :null  
Actual   :0  
<Click to see difference>
```

Figure 4: Step 8