



JUNIT Test Suite

Agenda

1

Test Suite

Objectives

At the end of this module, you will be able to:

- Understand Test Suites

Test Suite



Test Suite

- Test Suite is a Convenient way to group together tests that are related
- Used to bundle a few unit test cases and run it together
- Annotations used for this
 - `@RunWith`
 - Used to invoke the class which is annotated to run the tests
in that class
 - `@Suite`
 - Allows you to manually build a suite containing tests from many
classes

User Defined Class 1

```
package junit.first;

public class Stringmanip {
    String datum;

    public Stringmanip(String datum) {
        this.datum = datum;
    }

    public String upperCase() {
        return datum.toUpperCase();
    }
}
```

Test Case for User Defined Class 1

```
package junit.first;

import junit.first.Stringmanip.*;
import java.util.*;
import org.junit.Test;
import org.junit.runners.*;
import org.junit.runner.RunWith;
import static org.junit.Assert.*;

    @RunWith(Parameterized.class)
public class StringmanipTest2
{
    // Fields
    private String datum;
    private String expected;

    public StringmanipTest2(String datum, String expected)    {
        this.datum = datum;
        this.expected = expected;
    }
}
```

Test Case for User Defined Class 1 (Contd.).

@Parameters

```
public static Collection<Object[]> generateData()
{
    Object[][] data = new Object[][]
    {
        { "Smita", "SMITA" },
        { "smita", "SMITA" },
        { "SMitA", "SMITA" }
    };
    return Arrays.asList(data);
}
```

In this example, the parameter generator returns a List of arrays.

Each row has two elements:
{ input_data, expected_output }.
These data are hardcoded into the class, but they could be generated in any way you like.

@Test

```
public void testUpperCase()
{
    Stringmanip s = new Stringmanip(this.datum);
    String actualResult = s.upperCase();
    assertEquals(actualResult, this.expected);
}
}
```


User Defined Class 2

```
package junit.first;

public class Calc {

    public int add( int v1, int v2)    {
        return v1+v2;
    }

    public int sub( int v1, int v2)    {
        return v1-v2;
    }

    // You can add more functions here as needed..

}
```

Test Case for User Defined Class 2

```
package junit.first;

import static org.junit.Assert.*;
import org.junit.Test;

public class CalcTest {

    Calc c = new Calc();

    @Test

    public void testAdd() {
        assertEquals(5, c.add(10,-5));
        assertEquals(5, c.add(10,-5));
        assertEquals(5, c.add(20,-15));
        assertEquals(5, c.add(0,5));
    }
}
```

Test Case for User Defined Class 2

@Test

```
public void testSub() {  
    assertEquals(5, c.sub(10,5));  
    assertEquals(95, c.sub(100,5));  
    assertEquals(5, c.sub(20,15));  
    assertEquals(5, c.sub(10,5));  
}  
}
```

Test Suite

- In JUnit, both **@RunWith** and **@Suite** annotation are used to run the suite test.
- When a class is annotated with **@RunWith**, JUnit will invoke the class it references to run the tests in that class.
- Using Suite as a runner allows you to manually build a suite containing tests from many classes.

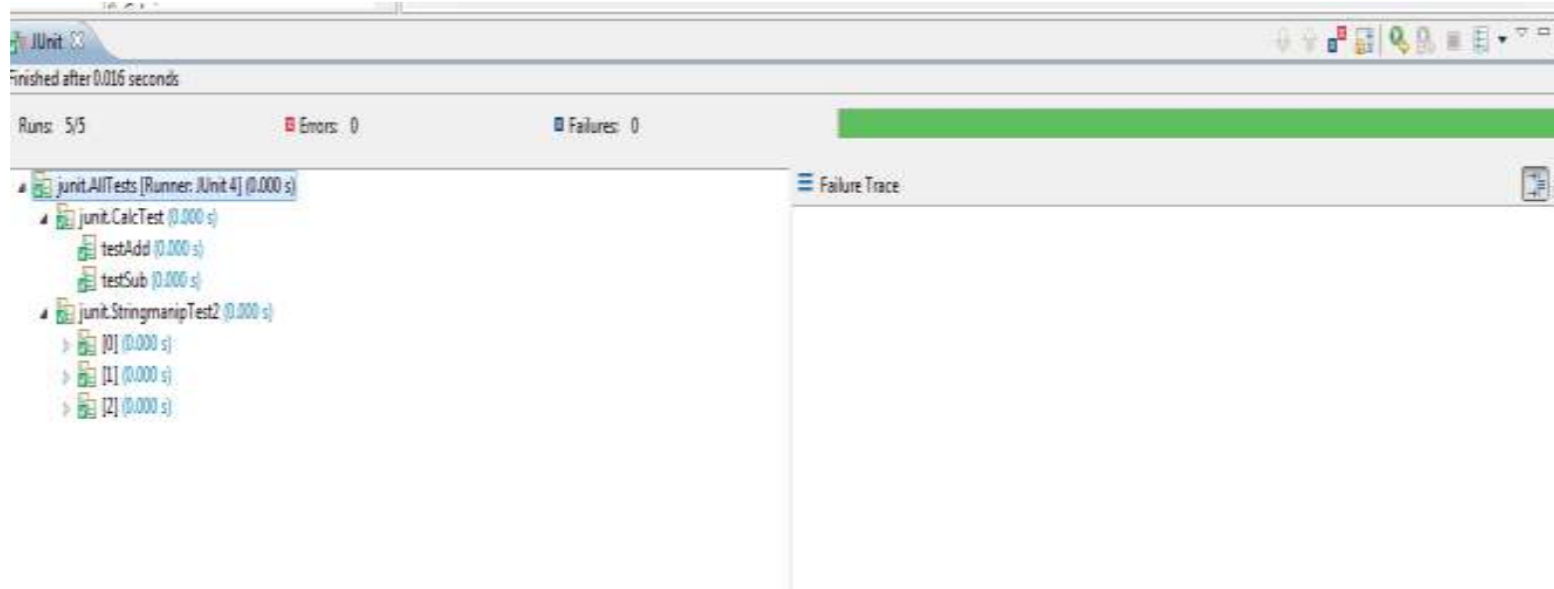
```
@RunWith(Suite.class)
@Suite.SuiteClasses({
    CalcTest.class,
    StringmanipTest2.class
})
```

```
public class AllTests
{
}
```

Note >>
Test Classes
which are defined in
previous pages are
included here...

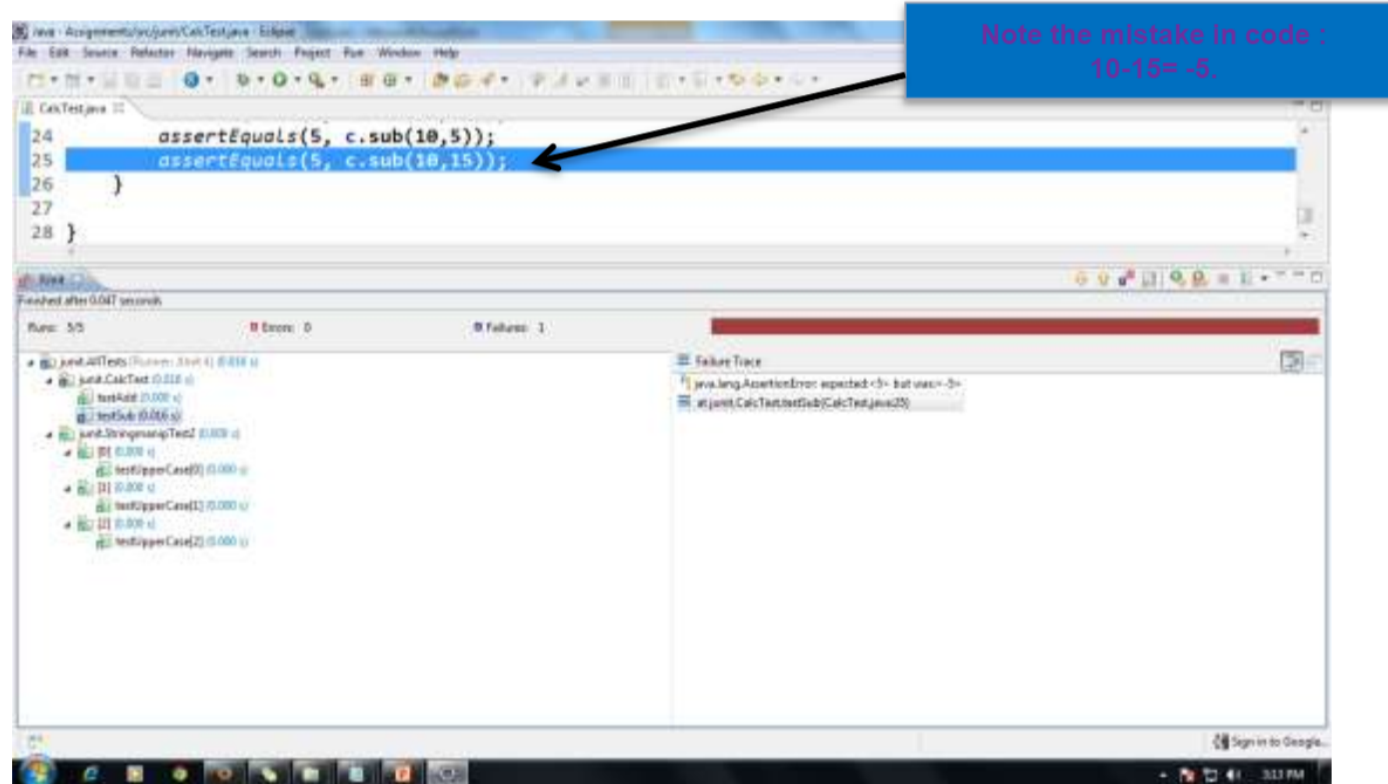
Test Suite(Contd.).

- When all the test cases are executed successfully, it shows **green color** signal as shown below.



Test Suite(Contd.).

- When any one test cases fails, it shows **Brown color** signal as shown below.



Quiz

1. Which of the following annotations has to be used before each of the test method?

- a. @Before
- b. @BeforeClass
- c. @After
- d. None of the above

None of the
above

2. Which of the following are true?

- a. All assert methods are static methods
- b. The JUnit test methods can be private
- c. The JUnit test methods should start with the test keyword
- d. All of the above true

All assert methods are static
methods



Summary

In this module, you were able to:

- Understand Test Suites



Thank You