**Setting Up Junit**

CalculatorTest.java

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

*@*Test

public void testAddition() {

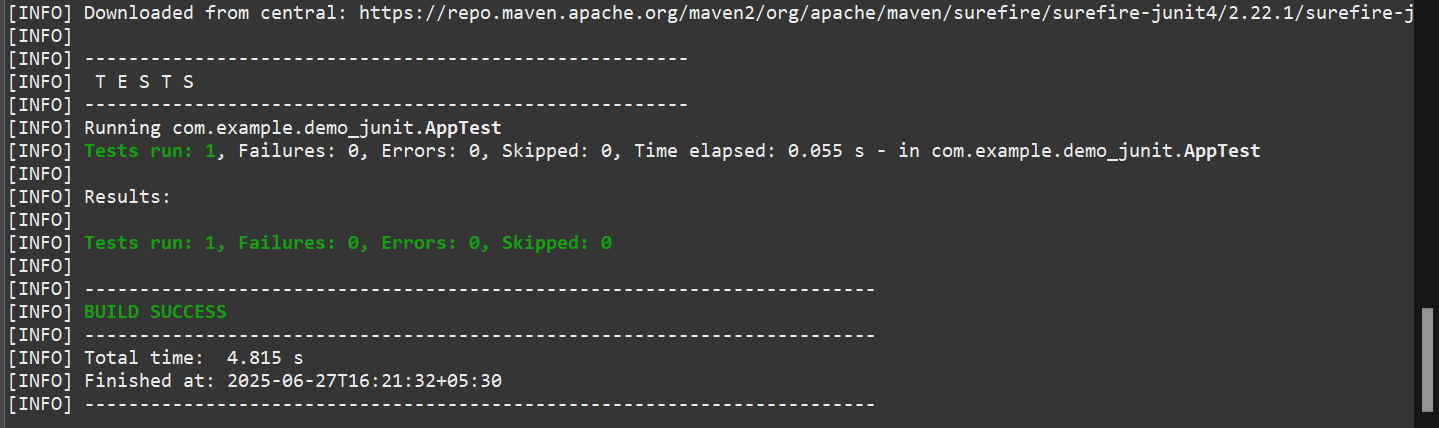
int result = 2 + 3;

assertEquals(5, result);

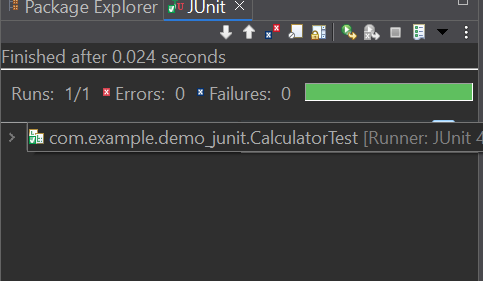
}

}

**Test Output:**



**JUnit Test:**



2. **Assertions in JUnit**

package com.example.demo\_junit;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

*@Test*

public void testAssertions() {

*assertEquals*(5, 2 + 3);

*assertTrue*(5 > 3);

*assertFalse*(5 < 3);

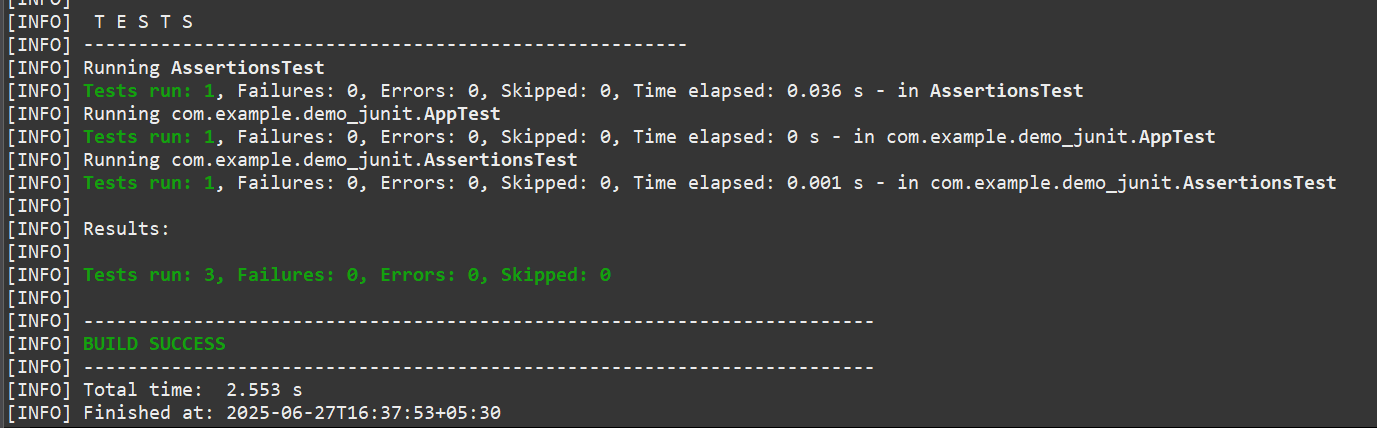
*assertNull*(null);

*assertNotNull*(new Object());

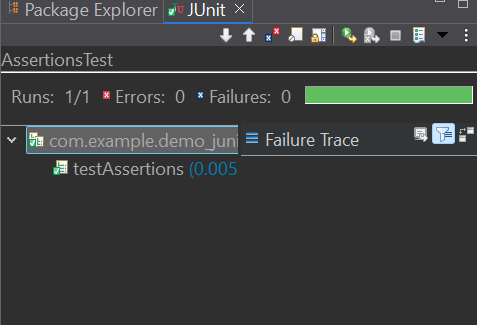
}

}

**Test Output:**



**Junit Test**



**3. AAA Pattern:**

Calculator.java:

package com.example.demo\_junit;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int multiply(int a, int b) {

return a \* b;

}

}

**CalculatorTest.java:**

package com.example.demo\_junit;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

*@Before*

public void setUp() {

calculator = new Calculator(); // Arrange

System.***out***.println("Setup complete.");

}

// Teardown method – runs after each test

*@After*

public void tearDown() {

calculator = null;

System.***out***.println("Teardown complete.");

}

*@Test*

public void testAddition() {

// Arrange done in setUp()

// Act

int result = calculator.add(2, 3);

// Assert

*assertEquals*(5, result);

}

*@Test*

public void testMultiplication() {

// Act

int result = calculator.multiply(4, 5);

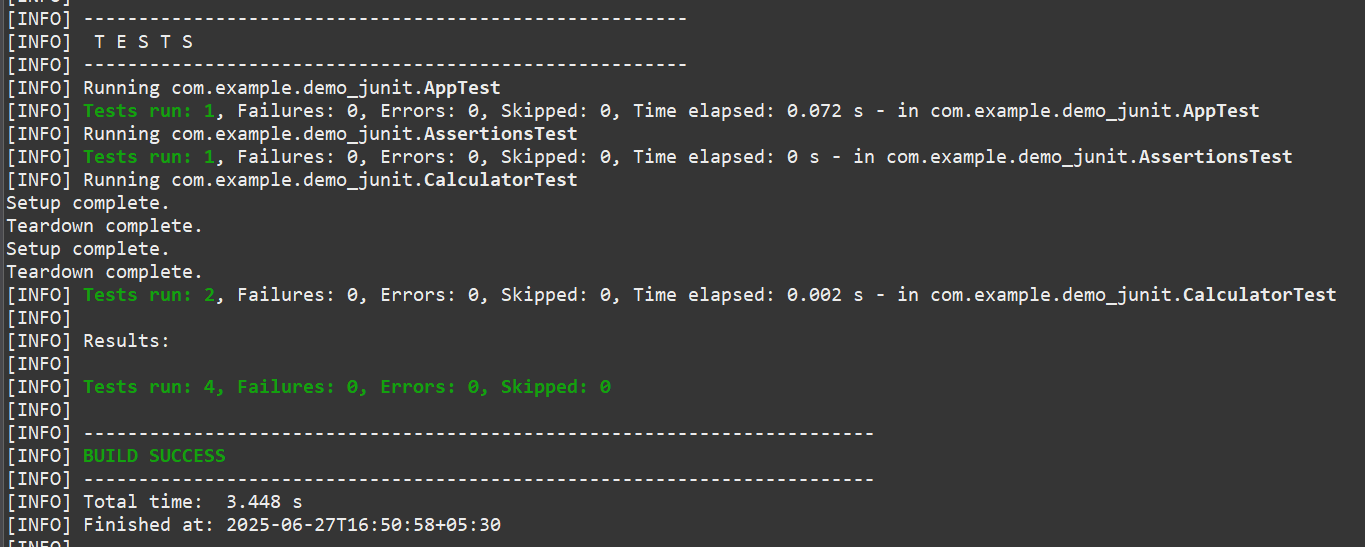
// Assert

*assertEquals*(20, result);

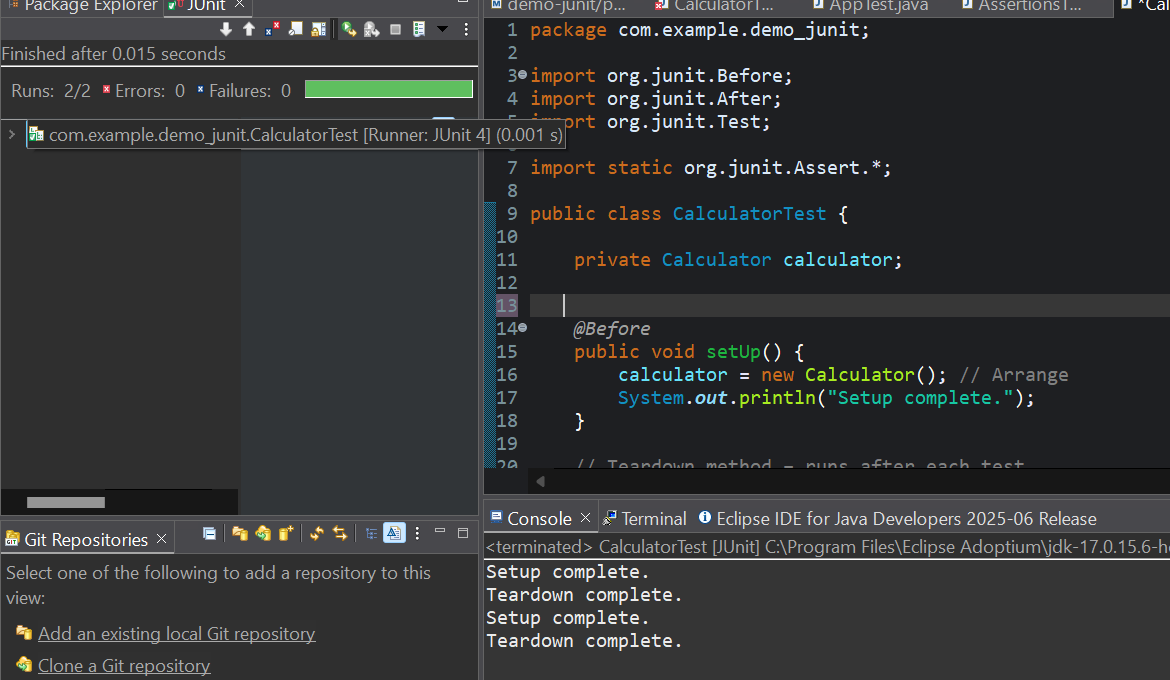
}

}

**Output:**

****

**Junit Output:**



**4. Mockito :**

**Mocking and Stubbing**

ExternalApi.java

package com.example.mockito;

public interface ExternalApi {

String getData();

}

MyService.java

package com.example.mockito;

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

MyServiceTest.java

package com.example.mockito;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@*Test

public void testExternalApi() {

// Create mock

ExternalApi mockApi = mock(ExternalApi.class);

// Stub method

when(mockApi.getData()).thenReturn("Mock Data");

// Inject mock into service

MyService service = new MyService(mockApi);

String result = service.fetchData();

// Assert the result

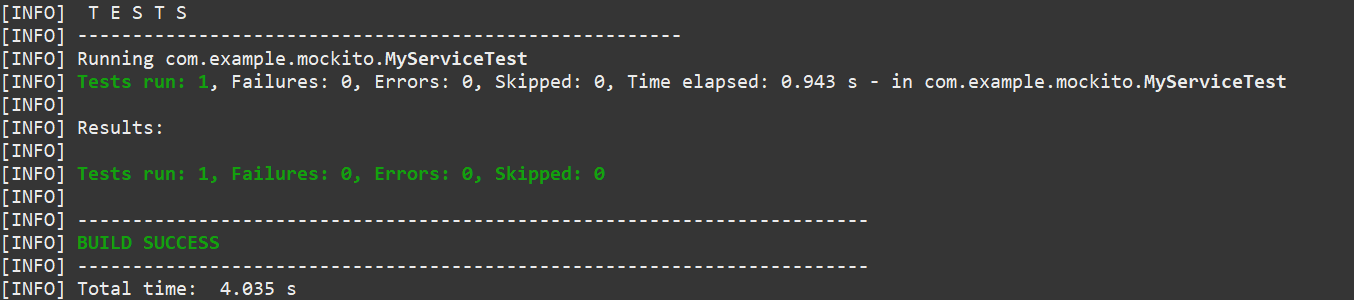
assertEquals("Mock Data", result);

}

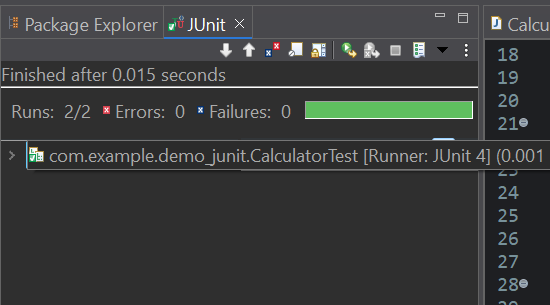
}

Maven Tests:

MyService.java



Junit Test:



**Verifying Interations**

TestDataUploader.java

package com.mockito;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class TestDataUploader {

*@Test*

public void testUploadCalledWithCorrectArgument() {

// Step 1: Create mock

NetworkClient mockClient = Mockito.*mock*(NetworkClient.class);

// Step 2: Call method with specific argument

DataUploader uploader = new DataUploader(mockClient);

uploader.performUpload();

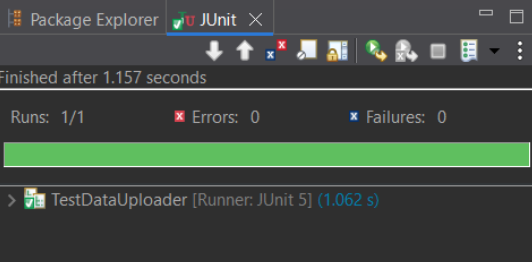
// Step 3: Verify interaction

*verify*(mockClient).upload("TestPayload");

}

}

**JUnit Test:**



**Maven Test:**

TestDataUploader.java

