Day 23 july 31

Name Aravind Kasanagottu

Id mvsnarav

// Task 1 - Basic JUnit Test

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class TestJunit {

@Test

public void testSubtract() {

int a = 10, b = 5;

assertEquals(5, a - b);

}

}

// TestRunner for Task 1

import org.junit.runner.JUnitCore;

import org.junit.runner.Result;

import org.junit.runner.notification.Failure;

public class TestRunner {

public static void main(String[] args) {

Result result = JUnitCore.runClasses(TestJunit.class);

for (Failure failure : result.getFailures()) {

System.out.println(failure.toString());

}

System.out.println(result.wasSuccessful());

}

}

// Task 2 - Test Suite Example

// MessageUtil.java

public class MessageUtil {

private String message;

public MessageUtil(String message) {

this.message = message;

}

public String printMessage() {

return message;

}

public String printUpperCase() {

return message.toUpperCase();

}

public String appendExclamation() {

return message + "!";

}

}

// TestCase001.java

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class TestCase001 {

@Test

public void testPrintMessage() {

MessageUtil msgUtil = new MessageUtil("Hello");

System.out.println("Running TestCase001 - testPrintMessage");

assertEquals("Hello", msgUtil.printMessage());

}

}

// TestCase002.java and TestCase003.java follow similar pattern...

// TestSuiteDemo.java

package July30;

import org.junit.runner.RunWith;

import org.junit.runners.Suite;

@RunWith(Suite.class)

@Suite.SuiteClasses({

TestCase001.class,

TestCase002.class,

TestCase003.class

})

public class TestSuiteDemo {

}

// Task 4 - Assert Methods Implementation

public class MessageUtil {

public String getMessage() {

return "Hello World";

}

public int[] getNumbers() {

return new int[]{1, 2, 3};

}

}

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

import org.junit.jupiter.api.Test;

public class TestCase001 {

@Test

public void testAssertSame() {

MessageUtil util1 = new MessageUtil();

MessageUtil util2 = util1;

assertSame(util1, util2, "Both should point to same object");

}

@Test

public void testAssertNotSame() {

MessageUtil util1 = new MessageUtil();

MessageUtil util2 = new MessageUtil();

assertNotSame(util1, util2, "Objects should not be same");

}

@Test

public void testAssertArrayNotEquals() {

int[] expected = {1, 2, 3};

int[] actual = {1, 2, 4};

try {

assertArrayEquals(expected, actual);

fail("Arrays are equal but expected not to be equal");

} catch (AssertionError e) {

// Pass

}

}

}

// Task 5 - Ignore Test

import static org.junit.Assert.assertEquals;

import org.junit.Test;

import org.junit.Ignore;

public class TestCase001 {

@Ignore

@Test

public void testPrintMessage() {

MessageUtil msgUtil = new MessageUtil("Hello");

System.out.println("Happy");

assertEquals("Hello", msgUtil.printMessage());

}

}

// Task 6 - Timeout Test

import org.junit.Test;

public class TimeoutFailTest {

@Test(timeout = 1000)

public void testFailsDueToTimeout() throws InterruptedException {

Thread.sleep(20000);

System.out.println("test passed within timeout.");

}

}

// Task 7 - Exception Test

import org.junit.Test;

public class ArithmeticExceptionTest {

@Test(expected = ArithmeticException.class)

public void testArithmeticException() {

int result = 10 / 0;

}

}  
  
  
  
task 03:  
  
Common Assertion Methods:

1. assertEquals(expected, actual) - Checks equality

2. assertNotEquals(expected, actual) - Checks inequality

3. assertTrue(condition) - Verifies true condition

4. assertFalse(condition) - Verifies false condition

5. assertNull(object) - Checks for null

6. assertNotNull(object) - Checks for not null

7. assertSame(expected, actual) - Checks object reference

8. assertNotSame(expected, actual) - Checks different references

9. assertArrayEquals(expectedArray, actualArray) - Compares arrays

10. assertThrows(expectedException.class, executable) - Checks exceptions

11. assertDoesNotThrow(executable) - Verifies no exceptions

12. assertAll(executables...) - Groups assertions

13. fail(message) - Forces test failure