<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>coms.usertest</groupId>

<artifactId>Junit</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>usertest</name>

<description>user authentication test</description>

<dependencies>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>5.9.3</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.platform</groupId>

<artifactId>junit-platform-runner</artifactId>

<version>1.9.3</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mockito/mockito-all -->

</dependencies>

</project>

package com.simplilearn;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.DisplayName;

import org.junit.jupiter.api.Nested;

import org.junit.jupiter.api.Test;

import org.junit.platform.runner.JUnitPlatform;

import org.junit.runner.RunWith;

@DisplayName("Nested Test Example")

@RunWith(JUnitPlatform.class)

public class E\_NestedTest {

@BeforeEach

public void init() {

System.out.println("Init on outer invoked");

}

@Nested

class A {

@BeforeEach

public void init() {

System.out.println("Init on inner invoked");

}

@Test

@DisplayName("Test inside class A")

public void test() {

System.out.println(" Example test for class A");

}

@Test

@DisplayName("Test inside class B")

public void testB() {

System.out.println(" Example test for class B");

}

}

}

package com.simplilearn;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.DisplayName;

import org.junit.jupiter.api.RepeatedTest;

import org.junit.jupiter.api.Test;

import org.junit.platform.runner.JUnitPlatform;

import org.junit.runner.RunWith;

import org.mockito.Mockito;

@DisplayName("Junit Repeated Test example")

@RunWith(JUnitPlatform.class)

public class F\_RepeatedTest {

Calculator calculator;

RecordPublisher mockRecordPublisher = Mockito.mock(RecordPublisher.class);

@BeforeEach

void init() {

calculator = new Calculator(mockRecordPublisher);

Mockito.when(mockRecordPublisher.publish()).thenReturn(Boolean.TRUE);

}

@Test

@RepeatedTest(1000)

public void test() {

Assertions.assertEquals(2, calculator.calculate(1, 1));

}

}