<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 java.util.ArrayList;

import java.util.Arrays;

import java.util.Collection;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.DisplayName;

import org.junit.jupiter.api.DynamicTest;

import org.junit.jupiter.api.TestFactory;

import org.junit.platform.runner.JUnitPlatform;

import org.junit.runner.RunWith;

import org.mockito.Mockito;

@DisplayName("Dynamic Test")

@RunWith(JUnitPlatform.class)

public class H\_DynamicTest {

Calculator c;

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

@BeforeEach

void init() {

c = new Calculator(mockRecordPublisher);

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

}

@TestFactory

Collection<DynamicTest> dynamicTests() {

Collection<DynamicTest> tests = new ArrayList<>();

tests.add(DynamicTest.dynamicTest("both arguments positive", ()-> { Assertions.assertEquals(2, c.calculate(1, 1)); } ));

tests.add(DynamicTest.dynamicTest("one positive and one negative", ()-> { Assertions.assertEquals(0, c.calculate(1, -1)); } ));

tests.add(DynamicTest.dynamicTest("both arguments negative", ()-> { Assertions.assertEquals(-2, c.calculate(-1, -1)); } ));

tests.add(DynamicTest.dynamicTest("both arguments zero", ()-> { Assertions.assertEquals(0, c.calculate(0, 0)); } ));

return tests;

}

}