**4. Write 10 Integration test cases for a Calculator application, covering various operations and scenarios.**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.junit.Assert;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class CalculatorTest {

private WebDriver driver;

@Before

public void setUp() {

System.setProperty("webdriver.chrome.driver", "c/desktop/chromedriver");

driver = new ChromeDriver();

driver.get("https://www.online-calculator.com/full-screen-calculator/");

}

@After

public void tearDown() {

driver.quit();

}

private void inputValue(String value) {

driver.findElement(By.id("inputFieldId")).sendKeys(value); }

private void clickButton(String buttonLabel) {

driver.findElement(By.xpath("//button[text()='" + buttonLabel + "']")).click();

}

private String getDisplayValue() {

return driver.findElement(By.id("displayFieldId")).getText();

}

@Test

public void testAdditionOfTwoPositiveNumbers() {

inputValue("5");

clickButton("+");

inputValue("10");

clickButton("=");

Assert.assertEquals("15", getDisplayValue());

}

@Test

public void testAdditionOfPositiveAndNegativeNumber() {

inputValue("7");

clickButton("+");

inputValue("-3");

clickButton("=");

Assert.assertEquals("4", getDisplayValue());

}

@Test

public void testSubtractionResultingInNegativeNumber() {

inputValue("5");

clickButton("-");

inputValue("10");

clickButton("=");

Assert.assertEquals("-5", getDisplayValue());

}

@Test

public void testMultiplicationOfTwoNegativeNumbers() {

inputValue("-4");

clickButton("\*");

inputValue("-6");

clickButton("=");

Assert.assertEquals("24", getDisplayValue());

}

@Test

public void testDivisionByZero() {

inputValue("10");

clickButton("/");

inputValue("0");

clickButton("=");

Assert.assertEquals("Error: Division by zero is undefined.", getDisplayValue());

}

@Test

public void testMixedOperations() {

inputValue("2");

clickButton("+");

inputValue("3");

clickButton("\*");

inputValue("4");

clickButton("-");

inputValue("5");

clickButton("=");

Assert.assertEquals("9", getDisplayValue());

}

@Test

public void testSquareRoot() {

clickButton("√");

inputValue("16");

clickButton("=");

Assert.assertEquals("4", getDisplayValue());

}

@Test

public void testPercentageCalculation() {

inputValue("20");

clickButton("%");

inputValue("50");

clickButton("=");

Assert.assertEquals("10", getDisplayValue());

}

@Test

public void testAdditionOfDecimalNumbers() {

inputValue("5.5");

clickButton("+");

inputValue("4.2");

clickButton("=");

Assert.assertEquals("9.7", getDisplayValue());

}

@Test

public void testChainingOperations() {

inputValue("3");

clickButton("+");

inputValue("2");

clickButton("\*");

inputValue("5");

clickButton("-");

inputValue("1");

clickButton("/");

inputValue("2");

clickButton("=");

Assert.assertEquals("10", getDisplayValue());

}

public static void main(String[] args) {

org.junit.runner.JUnitCore.main("CalculatorTest");

}

}