**Junit testcases for the services**

**package** com;

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

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

**import** org.junit.jupiter.api.Test;

**import** org.junit.jupiter.api.extension.ExtendWith;

**import** org.springframework.boot.test.context.SpringBootTest;

**import** org.springframework.test.context.junit.jupiter.SpringExtension;

@ExtendWith(SpringExtension.**class**)

@SpringBootTest

**public** **class** TestCalc {

CalculatorApp cs =**new** CalculatorApp();

/\* @Test

public void testSum()

{

Assertions.assertEquals(5,cs.sum());

}

@Test

public void testMultiply()

{

Assertions.assertEquals(6,cs.multiply());

}

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//testcase for string

@Test

**public** **void** testHello1()

{

Assertions.*assertEquals*("Hello!!!",cs.sayHi());

}

@Test

**public** **void** testHello2()

{

Assertions.*assertNotSame*("Hi", cs.sayHi());

}

//testcase for random variables

@Test

**public** **void** testCounter()

{

Assertions.*assertNotEquals*(6, cs.counterValue());

}

@Test

**public** **void** testCounter2()

{

Assertions.*assertNotNull*(cs.counterValue());

}

//testcase for multiply

@Test

**public** **void** testMultiply\_zero()

{

Parameters params=**new** Parameters();

params.setA(0);

params.setB(5);

Assertions.*assertEquals*(0, cs.multiply(params));

}

@Test

**public** **void** testMultiply\_negative()

{

Parameters params=**new** Parameters();

params.setA(1);

params.setB(-1);

Assertions.*assertEquals*(-1, cs.multiply(params));

}

@Test

**public** **void** testMultiply\_signs()

{

Parameters params=**new** Parameters();

params.setA(+2);

params.setB(-3);

Assertions.*assertEquals*(-6, cs.multiply(params));

}

//testcases for division

@Test

**public** **void** testDivide() {

Parameters params=**new** Parameters();

params.setA(10);

params.setB(-5);

Assertions.*assertEquals*(-2,cs.division(params));

}

@Test

**public** **void** testDivideZero() {

Parameters params=**new** Parameters();

params.setA(10);

params.setB(0);

Assertions.*assertThrows*(ArithmeticException.**class**, () -> cs.division(params));

}

//testcase for getParam

@Test

**public** **void** testgetParam() {

**int** num =8;

Assertions.*assertSame*(8,cs.parametertest(num));

}

}

**Output**

