**JUNIT**

1. **Division Test**

package com.ust.test.junit;

import static org.junit.Assert.\*;

import org.junit.Test;

public class DivisionTest {

int num1, num2;

public DivisionTest(int num1, int num2){

this.num1 = num1;

this.num2 = num2;

}

public int division() throws ArithmeticException{

return num1/num2;

}

}

1. **Division Testcase**

package com.ust.test.junit;

import static org.junit.Assert.\*;

import org.junit.Test;

public class DivisionTestCase {

DivisionTest divisionTest1 = new DivisionTest(10, 2);

DivisionTest divisionTest2 = new DivisionTest(10, 0);

@Test

public void test() {

assertEquals(5, divisionTest1.division());

}

@Test(expected = ArithmeticException.class)

public void testException() {

assertEquals(5, divisionTest2.division());

}

}

1. **Division Testcase1**

package com.ust.test.junit;

import org.junit.Test;

import static org.junit.Assert.\*;

public class DivisionTestCase1 {

DivisionTest divisionTest1 = new DivisionTest(10, 2);

DivisionTest divisionTest2 = new DivisionTest(10, 0);

@Test

public void test() {

System.out.println("In DivisionTestCase1.test");

assertEquals(5, divisionTest1.division());

}

@Test(expected = ArithmeticException.class)

public void testException() {

System.out.println("In DivisionTestCase1.testException");

assertEquals(5, divisionTest2.division());

}

}

1. **Division Testcase2**

package com.ust.test.junit;

import static org.junit.Assert.\*;

import org.junit.Test;

public class DivisionTestCase2 {

DivisionTest divisionTest = new DivisionTest(10, 5);

@Test

public void test() {

System.out.println("In DivisionTestCase2.test");

assertEquals(2, divisionTest.division());

}

}

1. **JTestexample1**

package com.ust.test.junit;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Ignore;

import org.junit.Test;

import static org.junit.Assert.\*;

public class JTestExample1 {

@BeforeClass

public static void beforeclass() {

System.out.println("In beforeClass method");

}

@AfterClass

public static void afterClass() {

System.out.println("In afterclass method");

}

@Before

public void before() {

System.out.println("In before method");

}

@After

public void after() {

System.out.println("In after method");

}

@Test

public void test() {

System.out.println("In TestCase");

//fail("Not yet implemented");

}

@Test

public void test1() {

System.out.println("In TestCase1");

}

@Ignore

@Test

public void test2() {

System.out.println("In testCase2");

}

}

1. **Test Runner**

package com.ust.test.junit;

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(DivisionTestCase.class);

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

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

}

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

}

}

1. **Test Suite**

package com.ust.test.junit;

import org.junit.runner.RunWith;

import org.junit.runners.Suite;

import static org.junit.Assert.\*;

import org.junit.Test;

@RunWith(Suite.class)

@Suite.SuiteClasses({

DivisionTestCase1.class,

DivisionTestCase2.class

})

class TestSuite {

public class DivisionTestCase1 {

DivisionTest divisionTest1 = new DivisionTest(10, 2);

DivisionTest divisionTest2 = new DivisionTest(10, 0);

@Test

public void test() {

System.out.println("In DivisionTestCase1.test");

assertEquals(5, divisionTest1.division());

}

@Test(expected = ArithmeticException.class)

public void testException() {

System.out.println("In DivisionTestCase1.testException");

assertEquals(5, divisionTest2.division());

}

}

}