# Test the method who throw exception

|  |
| --- |
| **public** **int** devide(**int** a, **int** b) **throws** Exception {  **if** (b == 0) {  **throw** **new** Exception("除数不能为0！");  }  **return** a / b;  } |

|  |
| --- |
| **public** **void** testDevide1() {  Calculator cal = **new** Calculator();  Throwable t = **null**;  **try** {  cal.devide(4, 0);  Assert.*fail*();// must throw exception on last sentence  } **catch** (Exception e) {  t = e;  }  Assert.*assertNotNull*(t);//exception exist  Assert.*assertEquals*(Exception.**class**, t.getClass());//exception type  Assert.*assertEquals*("除数不能为0！", t.getMessage());//error message  } |

|  |
| --- |
| **public** **void** testDevide2() {  Calculator cal = **new** Calculator();  **int** result = 0;  **try** {  result = cal.devide(6, 4);  } **catch** (Exception e) {  Assert.*fail*();// fail the case if got an exception  }  Assert.*assertEquals*(1, result);// assert the result  } |

# Setup() and teardown()

Those methods will be called before/after **every** test method invoked.

There are no methods which be called before/after **all** the test method and only be called once for the JUnit3.8. But already support in JUnit4.(@BeforeClass and @AfterClass)

# How to run a test suite?

|  |
| --- |
| **public** **class** MyTestSuite **extends** **TestCase** {  **public static Test suite**() {**// this signature of this method can't be**  **// change, otherwise it won't OK**  TestSuite suite = **new** TestSuite();  suite.**addTestSuite**(CalculatorTest.**class**);  // suite.addTestSuite(PersonTest.class);  // suite.addTestSuite(MyTestSuite.class); **also can add test suite**  **return** suite;  }  } |

# An example for JUnit4

|  |
| --- |
| **public** **class** CalculatorTest {//No need extends TestCase  @Test(expected = Exception.**class**)  **public** **void** devide1() **throws** Exception {  **new** Calculator().devide(4, 0);  }  @Test  **public** **void** testDevide2() {//no need to name the method name start with “test”  Calculator cal = **new** Calculator();  **int** result = 0;  **try** {  result = cal.devide(6, 4);  } **catch** (Exception e) {  Assert.*fail*();  }  Assert.*assertEquals*(1, result);  }  } |