通常情况下，围绕着一个被测接口，根据边界值、等价类等测试方法，我们会涉及多条用例，使用不同的数据来进行测试，此时我们是否需要重复的写很多测试代码来实现这种场景下的测试呢？

答案是否定的。TestNG 提供了参数化的特性，允许使用不同的值重复运行同一个测试方法。

TestNG支持两种不同的方式将参数直接传递给测试方法：

使用 @Parameters注解，需结合 testng.xml使用。

使用 @DataProvider注解。

本文中主要介绍常用的 @DataProvider注解方式，将参数传递给@Test方法。

**简单的@DataProvider示例**

我们创建一个简单的@DataProvider示例，传递一个int 参数。

import org.testng.Assert;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class TestDemo {

@Test(dataProvider = "testOracleProvideData")

public void testOracle(int actual,int expect) {

System.out.println("Run testOracle" + " Parameters：" + actual +","+ expect);

Assert.assertEquals(actual , expect);

}

@DataProvider(name = "testOracleProvideData")

public Object[][] provideData() {

return new Object[][] { {9527,9527}, {9528,9528}, {9526,9526} };

}

}

执行示例 测试类代码，运行结果如下：

Run testOracle Parameters：9527,9527

Run testOracle Parameters：9528,9528

Run testOracle Parameters：9526,9526

===============================================

Default Suite

Total tests run: 3, Failures: 0, Skips: 0

===============================================