TestNG整理 收藏  
1 基本概念  
TestNG：即Testing, Next Generation，下一代测试技术，是根据JUnit和NUnit思想，采用jdk的annotation技术来强化测试功能并借助XML 文件强化测试组织结构而构建的测试框架。最新版本5.12，Eclipse插件最新版本：testng-eclipse-5.12.0.6

TestNG的应用范围：

Ø单元测试

Ø集成测试

Ø系统测试

使用TestNG的主要步骤：

Ø编写测试业务逻辑，在代码中插入TestNG annotations

Ø在配置文件testng.xml或build.xml中配置测试信息

Ø运行TestNG

TestNG运行方式：

1、命令行方式

java org.testng.TestNG testng1.xml [testng2.xml testng3.xml ...]

2、Ant/Maven集成方式

3、Eclipse集成方式

TestNG提供Eclipse插件，在run configure中进行配置

4、IntelliJ‘s IDEA 及 NetBeans

TestNG的主要特点：  
1.Annotation

2.依赖性测试

3.支持并发测试

4.支持错误重运行测试

5.参数化测试

6.支持测试分组

7.通过testng.xml来管理测试

8.详实的报告，可按照自己需要进行二次开发定制

9.可以运行Junit测试

10.不仅仅适用于单元测试

TestNG利用annotation来标注测试类，使用annotation的优点：

方法名称不受任何模式或者格式的束缚；

测试类不需要从任何其他类继承；

如下是TestNG使用的annotation清单：

• @BeforeSuite, @BeforeTest, @BeforeClass,

@BeforeMethod, @BeforeGroups

• @AfterSuite, @AfterTest, @AfterClass, @AfterMethod, @AfterGroups

• @DataProvider

• @ExpectedExceptions

• @Factory

• @Test

• @Parameters

testNG.xml实例：

v Suite：由一个XML文件表示，可以包含一个或者多个Test，suite可以通过标签来运行其他testng的xml文件；

v Test：表示一个测试，可以包含一个或者多个TestNG类；

v TestNG类：就是一个简单的Java类，至少包含一个TestNG annotation；

v Method：一个普通的Java方法，其前由@Test标记

v Groups：TestNG可以将不同的Method分类到不同的Groups中，也可以将Class分类到不同的Groups中

v testNG.xml：该文件是TestNG运行时需要的配置文件，描述了要运行哪些测试，主要是对测试进行控制使用

2 基本使用  
2.1 处理异常  
TestNG可以简单方便的检测异常的发生，通过@ExpectedExceptions标注预期捕获的异常，当确实发生预期的异常，TestNG认为测试通过。也可以指定标注多个异常，例如：

2.2 失败处理  
当testNG运行的case有失败的时候，会在test-output目录下生成一个testng-failed.xml文件，这对于批量执行测试定位问题很有帮助，testng可以单独执行这些失败的测试；例如：

2.3 使用参数  
TestNG传参数的两种方式（xml文件，@DataProvider）

l 使用testng.xml设置参数

Ø 参数在xml文件中可以在suite级别定义，也可以在test级别定义；

Ø TestNG会尝试先在包含当前类的test标签中寻找参数，如果没找到则在上层的suite标签中查找。即在test标签中相同的参数对当前类当前方法的优先级比较高。

Ø TestNG支持这种传参方式的类型如下：

■ String

■ int/Integer

■ boolean/Boolean

■ byte/Byte

■ char/Character

■ double/Double

■ float/Float

■ long/Long

■ short/Short

Ø 对于非上述类型TestNG无法通过这种方式进行传参，可以通过@DataProvider方式传参

l @DataProvider方式传参

Ø 只提供了一个字符串属性：名称，供测试方法作为传递参数的annotation使用

Ø 两种DataProvider，一种是返回一个二维数组对象，如上面的例子；另外一种DataProvider是返回一个Iterator

Ø DataProvider可以向测试方法传递任意类型任意数目的参数

Ø 利用DataProvider提供不同的参数集合对一个测试方法进行多次调用

l @DataProvider和testNG.xml两种方式的比较

  优点  
 不足  
   
testng.xml  
 参数值在xml文件中指定，易于修改，修改后无需编译代码  
 传递参数的类型有限，只能传递基本的数据类型，复杂的Java对象无法传递  
   
@DataProvider  
 可以向测试方法传递任何类型，任意数量的Java对象  
 需要编码实现某些逻辑，以返回正确的对象

2.4 依赖测试  
l Groups依赖

TestNG对此用@Test的两个属性标签（dependsOnGroups，dependsOnMethods）进行支持，例如：

执行测试的顺序依次为：lanchuServer->deploy->test1

l 方法依赖

是指由@BeforeMethod、@BeforeTest、@Beforeclase、@BeforeSuite @BeforeGroups等标识指定的内容。

2.5 并发测试  
TestNG内建了对并发的支持，可以快速执行测试，并验证代码的线程安全性

Ø 其可能的执行结果为

• 线程号为: 11

• 线程号为: 15

• 线程号为: 12

• 线程号为: 14

• 线程号为: 13

Ø 当将threadPoolSize改为3后，其可能的执行结果如下：

• 线程号为: 13

• 线程号为: 12

• 线程号为: 11

• 线程号为: 12

• 线程号为: 11

v 并发执行：针对suite来定义并发线程数等，可以提高执行测试的速度。

parallel="methods" thread-count="2">

...

v Parellel = “methods”,每个测试方法在自己的一个线程中执行

v Parrellel=“tests”,在某个标签内的所有的测试方法在一个线程中执行；在测试线程不安全的代码，通常这样执行