JUnit 4

—— Terry

概念：

Junit是一个可用于执行软件单元测试的程序。JUnit的一般用法是创建一组单元测试，对软件进行修改时，这组单元测试可以自动运行，这样开发人员可以确保对他们正在创建的软件所做的改动不会破坏软件以前实现的功能。

TDD：测试驱动开发，是极限编程中倡导的程序开发方法，以倡导先写测试程序，然后编码实现其功能得名。

测试方法：Java类中的一个方法，他包含一个单元测试

测试类：包含一个或多个测试方法的Java类

断言：用于检查测试结果是否和预期一样的测试方法中的一条语句

测试固件（Test Fixture）：用于设置多个测试状态的类，通常在设置例程“代价高昂”或执行时间过长时使用

测试套件：一起运行的一组测试类

Junit 4：

JUnit 4新功能：参数测试，异常测试，超时测试，灵活固件，忽略测试的简单方法，对测试进行逻辑分组的新方法

摒弃了两条旧的规则：1.要求任何作为逻辑测试的编写方法要以test开头

2.类要从JUnit 的TestCase中扩展

新的规则：实现测试功能的方法只需要用一个@Test注释来修饰

异常测试：@Test(ecpected = IndexOutOfBoundsException.class)

忽略测试：@Ignore

固件测试：JUnit 4使用注释来减少固件花费的成本，允许为每个测试运行一次固件，或为整个类运行一次固件，或一次也不运行。有四种固件注释：针对类层次的固件有两种，针对方法层次的固件有两种

<类层次：@BeforeClass @AfterClass

方法层次：@Before @After

>

setup()和teardown() ：简单的理解就是，setup()是实现测试前的初始化工作，teardown()是主要实现测试完后的垃圾回收等工作

两个新的注释：@RunWith 设计它是为了方便让不同的运行器执行一个特别的测试类

@SuiteClasses 它将一个意欲表示测试套件的类列表作为参数

参数测试：能够创建由参数之供给的通用测试，结果是，你可以创建一个单个的测试用例并多次运行

参数测试需要5个步骤：

1.创建一个不含参数的通用测试

2.创建一个返回Collection类型的static feeder方法，并用@Parameter注释加以修饰

3.为在步骤1中定义的通用方法所要求的参数类型创建类成员

4.创建一个持有这些参数类型的构造函数，并把这些参数类型和步骤3中定义的类成员相应的联系起来

5.通过@RunWith注释，指定测试用例和Parameterized类一起运行