一、简单介绍

1.测试方法上必须使用@Test进行修饰

2.测试方法必须使用public void 进行修饰，不能带任何的参数

3.新建一个源代码目录来存放我们的测试代码

4.测试类的包应该和被测试类保持一致

5.测试单元中的每个方法必须可以独立测试，测试方法间不能有任何的依赖

6.测试类使用Test作为类名的后缀（不是必须）

7.测试方法使用test作为方法名的前缀（不是必须）

二、测试失败的两种情况

1、Failure一般由单元测试使用的断言方法判断失败所引起的，这经表示测试点发现了 问题， 就是说程序输出的结果和我们预期的不一样

2、error是由代码异常引起的，它可以产生于测试代码本身的错误，也可以是被测试 代码中的一个 隐藏的bug

3、测试用例不是用来证明你是对的，而是用来证明你没有错

1. Jnit运行流程

1.@BeforeClass修饰的方法会在所有方法被调用前执行，

而且该方法是静态的，所以当测试类加载后接着就会运行它，

而且在内存中它只会存在一份实例，它比较适合加载配置文件。

2.@AfterClass所修饰的方法通常用来对资源的清理，如关闭数据库的连接

3.@Before和@After会在每个测试方法的前后各执行一次。

1. 常用注解

·@Test:将一个普通的方法修饰成为一个测试方法

·@Test(expected=XX.class)：若抛出指定异常则测试成功,若未抛出指定异

常或抛出其他异常则测试失败

·@Test(timeout=1000)：超过1000毫秒则自动停止并报errors

·@BeforeClass:它会在所有的方法运行前被执行，static修饰

·@AfterClass:它会在所有的方法运行结束后被执行，static修饰

·@Before:它会在每一个测试方法运行前执行一次

·@After:它会在每一个测试方法运行后执行一次

·@Ignore:所修饰的测试方法会被测试运行器忽略

·@RunWith:可以更改测试运行器 org.junit.runner.Runner。一般使用默认

的测试器即可

1. 测试套件

1. 测试套件就是组织测试类一起运行的

(1)写一个作为测试套件的入口类，这个类里不包含其他的方法

(2)更改测试运行器Suite.class即在类前添加@Runwith(Suite.class)

(3)将要测试的类作为数组传入到Suite.SuiteClasses({a.class,b.class,...})

1. 参数化设置

\*1.更改默认的测试运行器为RunWith(Parameterized.class)

\*2.声明 变量来存放预期值和结果值

\*3.声明一个返回值为Collection的公共静态方法，并使用@Parameters进行修饰

\*4.为测试类声明一个带有参数的公共构造函数，并在其中为之声明变量赋值