相关注解：

1.@RunWith

@Runwith就是一个运行器,制定测试运行的测试类或者环境。如

[@Runwith](mailto:如@RunWith(PowerMockito.Class))(PowerMockRunner.class)。

2.@PrepareForTest

当需要Mock静态、final、私有方法时，就需要加注解@PrepareForTest。

使用场景：

(1) 当使用PowerMockito.whenNew方法时，必须加注解@PrepareForTest和@RunWith。注解@PrepareForTest里写的类是需要mock的new对象代码所在的类。

(2) 当需要mock final方法的时候，必须加注解@PrepareForTest和@RunWith。注解@PrepareForTest里写的类是final方法所在的类。

(3) 当需要mock静态方法的时候，必须加注解@PrepareForTest和@RunWith。注解@PrepareForTest里写的类是静态方法所在的类。

(4) 当需要mock私有方法的时候, 只是需要加注解@PrepareForTest，注解里写的类是私有方法所在的类。

(5) 当需要mock系统类的静态方法的时候，必须加注解@PrepareForTest和@RunWith。注解里写的类是需要调用系统方法所在的类。

3.@Mock:创建一个Mock

4.@InjectMocks:创建一个实例，简单的说这个Mock可以调用真实代码的方法，其余用@Mock(或者@Spy)注解创建的mock将被注入到该实例中。

5.Mockito.doReturn().when()与Mockito.when().thenReturn()的区别

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mockito.doReturn().when() | Mockito.when().thenReturn() | 默认是否执行对应的方法 |
| @Mock  private ServiceImpl service; | Mock的方法不会执行 | Mock的方法不会执行 | 都不执行 |
| @Spy  private ServiceImpl service = new ServiceImpl(); | Mock的方法不会执行 | Mock的方法会执行 | 都执行 |

如果您使用spied对象（或者@Spy注释）而不是模拟（使用@Mock注释），则两种方法的行为区别：

when().thenReturn()在返回指定值之前进行实际的方法调用。

doReturn().when()根本不调用该方法。

强烈建议使用：Mockito.doReturn().when()；

单元测试是高度局部化的东西，每个测试类都隶属于单一包。

功能测试是用来保证软件能够正常的运行。他们从客户的角度保证质量，并不关心程序员的生产力。

测试的要诀：测试你最担心出错的部分。

测试的重要技巧：寻找边界

监控对象-spy

当为真实对象创建一个监控对象(Spy)，调用监控对象时，真实对象也会被调用。有时，在监控对象上使用when(object)来进行打桩时不可能或者不符合实际的，因为方法很可能依赖真实的环境。因此，当使用监控对象时，尽量使用doReturn | doAnswer | doThrow函数来进行打桩。

Mock使用实例

1、使用doCallRealMethod().when()调用函数真正部分。

2、使用when().thenReturn自定义函数返回值。

Spy使用实例

1、使用when().thenReturn调用函数真正部分执行，依赖真实的环境。

2、使用doReturn.when(object).method自定义函数返回值。

代码统计覆盖率不同

1、@spy使用的真实的对象实例，调用的都是真实的方法，所以通过这种方式进行测试，在进行sonar覆盖率统计时统计出来是有覆盖率；

2、@mock出来的对象可能已经发生了变化，调用的方法都不是真实的，在进行sonar覆盖率统计时统计出来覆盖率为0.00%。

什么是Mock?

1.Mockito是mocking框架，它让你用简洁的API做测试。而且Mockito简单易学，它可读性强和验证语法简洁。

2.在面向对象程序设计中，模拟对象（英语：mock object，也译作模仿对象）是以可控的方式模拟真实对象行为的假的对象。

3.程序员通常创造模拟对象来测试其他对象的行为，很类似汽车设计者使用碰撞测试假人来模拟车辆碰撞中人的动态行为。

@Mock与@InjectMocks的区别

@Mock: 创建一个Mock.

@InjectMocks: 创建一个实例，简单的说是这个Mock可以调用真实代码的方法，其余用@Mock（或@Spy）注解创建的mock将被注入到用该实例中。

相同：Stub和Mock都是模拟外部依赖

不同：Stub是完全模拟一个外部依赖， 而Mock还可以用来判断测试通过还是失败

为什么要使用Mock?

在下面的情形，可能需要使用模拟对象来代替真实对象：

真实对象的行为是不确定的（例如，当前的时间或当前的温度）；

真实对象很难搭建起来；

真实对象的行为很难触发（例如，网络错误）;

真实对象速度很慢（例如，一个完整的数据库，在测试之前可能需要初始化）；

真实的对象是用户界面，或包括用户界面在内；

真实的对象使用了回调机制； 真实对象可能还不存在；

真实对象可能包含不能用作测试（而不是为实际工作）的信息和方法。