单元测试规范

一、定义

单元测试主要是用于测试程序模块，确保代码运行正确。单元测试是由开发者编写并进行运行测试。一般使用的测试框架是Junit。测试用例一般是针对\_方法\_ 级别的测试。

二、测试的好处

* 保证代码运行与我们预想的一样，代码正确性可以得到保证
* 程序运行出错时，有利于我们对错误进行查找（因为我们忽略我们测试通过的代码）
* 有利于提升代码架构设计（用于测试的用例应力求简单低耦合，因此编写代码的时候，开发者往往会为了对代码进行测试，将其他耦合的部分进行解耦处理）

三、样例

Junit测试样例

1、junit4注解

 

2、Mock



四、编写用例时应遵循的标准

1. 测试类和测试方法应该有一致的命名方案，如在工作类名后加上Test 从而形成测试类名
2. 走通单元测试方式全链路，依赖的第三方库采用mock方式，自己的内部Service及Dao层依赖走真实数据库
3. 将构造测试环境及清理测试环境的代码分别放在 Setup及 Teardown
4. 要保证测试粒度足够小，有助于精确定位问题。单测粒度至多是类级别，一般是方法级别
5. **编写单元测试代码遵守BCDE原则**

* Border，边界值测试
* Correct，正确的输入，并得到预期结果
* Design，与设计文档相结合，来编写单元测试
* Error，强制错误信息输入（如：非法数据、异常流程业务允许等），并得到预期结果

1. 确保测试与时间无关，不要使用过期的数据进行测试，以至于导致在随后的维护过程中很难重现测试,具有独立性、可重复执行；
2. 不要依赖或假定测试运行的顺序
3. 尽可能地利用 JUnit 提供地 assert 和 fail 方法以及异常处理的方法，其可以使代码更为简洁
4. 新增代码影响了原有单元测试，请及时修正
5. QA提出的bug，需要补覆盖此场景的单元测试
6. 每次编写的测试用例都要在Jenkins上执行通过

无效的单元测试案例：

