鸟域道场

产品研发 JAVA 单元测试规范

（V1.0）

Nander 研发团队

2023 年 3 月

文档信息

文档编号				发布版本	
起草时间	2023-03-03	定稿时间		当前版本	V1.0
起草人	姓名	部门	电话	电子邮件	
	商鞅	鸟域行政司		26089183@qq.com	
审阅人			签发人		

文档修改记录

序号	修改时间	修改人	主要修改内容	存档版本
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

目录

1 目的 4
2 适用范围 4
3 单元测试规范 4
4 单元测试编写步骤 6
4.1 单元测试方法编写步骤 6
4.2 Mock 编写步骤 6
5 框架与工具 6
5.1 JUNIT5 6
5.2 Mockito 框架 7
5.3 代码覆盖率工具 Jacoco 7
5.4 SonarQube 7
6 附则 8
7 参考文献 8

1 目的

为了帮助、指导 Java 开发人员进行单元测试编写，从而提高代码质量和可维护性，特制定本规范。

2 适用范围

本规范适用于组织内所有自研产品与交付类项目的 Java 单元测试编写。

3 单元测试规范

1、【强制】工程目录采用如下结构：
- src/main/java：应用程序 Java Source 目录。
- src/main/resource：应用程序资源文件目录。
- src/test/java：应用程序单元测试 Java Source 目录。
- src/test/resource：应用程序单元测试资源文件目录。

2、【强制】包名一致：
- src/test/java 下的包名和对应的 src/main/java 下的包名保持一致。

3、【强制】单元测试文件名字命名规则：
- 被测试文件名 + Test。

4、【强制】集成测试文件名字命名规则：
- 被测试文件名 + Tests。

5、【强制】单元测试方法的名字规则：
- test + 被测试的方法名字（首字母大写）。

6、【强制】好的单元测试必须遵守AIR：自动化（Automatic）、独立性（Independent）、可重复执行（Repeatable）原则。

7、【强制】单元测试应该是全自动执行的并且非交互式的。
- 测试用例通常是被定期执行的，执行过程必须完全自动化，不依赖人工检查。
- 单元测试中不准使用 System.out 来进行验证，必须使用断言来验证。

8、【强制】保持单元测试的独立性。
- 为了保证单元测试稳定可靠且便于维护，单元测试用例之间决不能互相调用，也不能依赖执行的先后次序。
- 反例：method2 需要依赖 method1 的执行，将执行结果作为 method2 的输入。

9、【强制】单元测试是可以重复执行的，不能受到外界环境的影响。
- 单元测试通常会被放到持续集成中，每次有代码签入时单元测试都会被执行。如果单测对外部环境（网络、服务、中间件等）有依赖，容易导致持续集成机制的不可用。对于某些不容易构造或者某些尚未实现的对象可以通过 Mock 的方式创建一个虚拟的对象实

现。通过这种方式可以专注于本模块的测试，避免其他模块出错引起本模块测试错误。推荐使用 Mockito 和 PowerMock 实现。

10、【强制】对于单元测试，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。
- 只有测试粒度小才能在出错时尽快定位到出错位置。

11、【强制】核心业务、核心应用、核心模块的增量代码确保单元测试通过。

- 新增代码及时补充单元测试，如果新增代码影响了原有单元测试，请及时修正。

12、【强制】单元测试代码必须写在如下工程目录：src/test/java。

- 不允许单元测试代码写在业务代码目录 src/main/java 下。
- 源码编译时会跳过此目录，而单元测试框架默认是扫描此目录。

13、【强制】单元测试作为一种质量保障手段，在项目提测前完成单元测试。

- 设计阶段，研发负责人与质量保证负责人确定单元测试业务范围和覆盖率要求。
- 编码阶段，开发人员产出单元测试报告，质量保证负责人对单元测试报告进行评审，通过后方可进入测试阶段。
- 传统分层架构项目主要测试 Service 层，如果 Controller 层存在复杂的逻辑，考虑将其重构后重新进行单元测试的编写。DDD 架构项目主要测试 Domain 层。

覆盖率指标：

■ 行覆盖率：度量被测程序的每行代码是否被执行，判断标准行中是否至少有一个指令被执行。

■ 分支覆盖率：度量if和switch语句的分支覆盖情况，计算一个方法里的分支总数，确定执行和不执行的分支数量。注意：异常处理不在此计算范围内。

14、【推荐】编写单元测试代码遵守BCDE原则，以保证被测试模块的交付质量。

- B：Border，边界值测试，包括循环边界、特殊取值、特殊时间点、数据顺序等。
- C：Correct，正确的输入，并得到预期的结果。
- D：Design，与设计文档相结合，来编写单元测试。
- E：Error，强制错误信息输入（如：非法数据、异常流程、业务允许之外状态等），并得到预期的结果。

15、【推荐】和数据库相关的单元测试可以设定自动回滚机制，不给数据库造成脏数据，或者对单元测试产生的数据有明确的前后缀标识。

16、【推荐】对于不可测的代码在适当的时机做必要的重构，使代码变得可测，避免为了达到测试要求而书写不规范测试代码。

17、【推荐】为了更方便地进行单元测试，业务代码应避免以下情况：

- 构造方法中做的事情过多。
- 存在过多的全局变量和静态方法。
- 存在过多的外部依赖。
- 存在过多的条件语句。
- 多层条件语句建议使用卫语句、策略模式、状态模式等方式重构。

4 单元测试编写步骤

4.1 单元测试方法编写步骤

单元测试方法的编写一般包含以下三个动作：

动作	说明
构建参数	构建并初始化输入参数
执行测试	调用业务方法并传递参数
验证结果	通过断言验证结果

4.2 Mock 编写步骤

为了专注于本模块的测试，可通过mock方法模拟依赖项，mock代码的编写一般包含以下四个动作：

动作	说明
模拟	建立用例容器，并且创建Mock对象
观察	观察依赖调用，设计打桩方法
打桩	对调用进行打桩，模拟返回值
验证	利用verify方法对打桩进行验证

5 框架与工具

5.1 JUNIT5

JUNIT是java领域最流行的单元测试框架，JUNIT5为最新版本。

- 官网：JUNIT5
- 官方文档：官方文档

JUNIT5常用注解如下：

- BeforeEach 注解

初始化方法，在任何一个测试方法执行之前，必须执行的代码。在该注解的方法中，可以进行一些准备工作，比如初始化对象，打开网络连接等。

- AfterEach 注解

释放资源，在任何一个测试方法执行之后，需要进行的收尾工作。

- Test 注解

表明这是一个测试方法。

- DisplayName 注解

为测试类或者测试方法设置展示名称。

- Disabled 注解

用于禁用一个测试类或测试方法。

- BeforeAll 注解

针对所有测试，也就是整个测试类中，在所有测试方法执行之前，都会先执行由它注解的方法，而且只执行一次。

- AfterAll 注解

针对所有测试，也就是整个测试类中，在所有测试方法都执行完之后，才会执行由它注解的方法，而且只执行一次。

5.2 Mockito 框架

对依赖一个或者多个外部服务的服务执行测试时，一般需要引入mock工具才能顺利测试。Mockito是常用的mock工具之一，提供了许多打桩和注解供我们使用，较常用的有@Mock和@InjectMock两个注解，以及一些判断方法。

- 官网：Mockito
- 官方文档：官方文档

Mockito常用注解如下：

- Mock 注解

创建的是全部mock的对象，既在对具体的方法打桩之前，mock对象的所有属性和方法全被置空（0或者null）。与之对应的是@Spy这个注解，@Spy可以创建部分mock的对象，部分mock对象的所有成员方法都会按照原方法的逻辑执行，直到被打桩返回某个具体的值。

- InjectMock 注解

声明了一个待测试的对象时，若该对象有类型为@Mock的成员变量，则@Mock定义的mock对象将会被注入到这个待测试的对象。

- thenReturn 方法

Mockito.when().thenReturn()，用来mock返回的值。

- verify 方法

用来验证预期的方法是否被调用了。

5.3 代码覆盖率工具 Jacoco

Jacoco是一个开源的覆盖率工具，它针对的开发语言是java，其使用方法很灵活，可以嵌入到Ant、Maven中。很多第三方的工具提供了对Jacoco的集成，如sonar、Jenkins等。

- 官网：jacoco
- 官方文档：官方文档

5.4 SonarQube

SonarQube是一款用于代码质量管理的开源工具，它主要用于管理源代码的质量。通过插件形式可以支持众多计算机语言，比如：java、C#、go、C/C++、PL/SQL、Cobol、JavaScript、

Groovy等。sonar可以通过PMD、CheckStyle、Findbugs等等代码规则检测工具来检测代码，帮助发现代码的漏洞，Bug，异味等信息。

- 官网：SonarQube
- 官方文档：官方文档

6 附则

- 本规范自发布之日起实施。
- 本规范由鸟域道场进行解释。
- 本文件著作权归北京鸟域花香科研室所有，用户除认真学习外不得以任何理由在未经同意的情况下擅自将文件用于其他用途。

7 参考文献

无