# ****项目编码规范****

**1.命名：**包名：包名全部用小写字母，通过 . 将各级连在一起。不应该使用下划线。类名：类型的命名，采用以大写字母开头的大小写字符间隔的方式（UpperCamelCase）。class命名一般使用名词或名词短语。interface的命名有时也可以使用形容词或形容词短语。annotation没有明确固定的规范。测试类的命名，应该以它所测试的类的名字为开头，并在最后加上Test结尾。例如：HashTest 、 HashIntegrationTest。(PS:经常见到写成TestXXXX的类名，以后注意了)。方法名：方法命名，采用以小写字母开头的大小写字符间隔的方式（lowerCamelCase）。方法命名一般使用动词或者动词短语。在JUnit的测试方法中，可以使用下划线，用来区分测试逻辑的名字，经常使用如下的结构：test<MethodUnderTest>\_<state> 。例如：testPop\_emptyStack 。测试方法也可以用其他方式进行命名。常量名：常量命名，全部使用大写字符，词与词之间用下划线隔开。（CONSTANCE\_CASE），常量一般使用名词或者名词短语命名。PS：常量是一个静态成员变量，但不是所有的静态成员变量都是常量。在选择使用常量命名规则给变量命名时，需要明确这个变量是否是常量。

**2.花括号：**一般用在if, else, for, do, 和 while等语句。甚至当它的实现为空或者只有一句话时，也需要使用。

**3.**if语句的嵌套层数保证在3层以内。太多层嵌套，最后自己都看不懂。直观感受一下，加上代码块，就会很复杂，而且还会有else。尤其是在代码不断增加需求的过程中，要重构一些代码，造成代码if嵌套混乱。

4.类的文件代码长度不要超过1000行。方法的长度不要超过150行，超出的考虑拆分一下。

5.枚举经常需要被外部引用，如果名字不能直接表达，需要加javadoc。枚举名需要全大写，单词用“\_”分割。

6.局部变量：局部变量不应该习惯性地放在语句块的开始处声明，而应该尽量离它第一次使用的地方最近的地方声明，以减小它们的使用范围。局部变量应该在声明的时候就进行初始化。如果不能在声明时初始化，也应该尽快完成初始化。

7.switch必须有default：每个switch语句中，都需要显式声明default标签。即使没有任何代码也需要显示声明。

8.修饰符的顺序：多个类和成员变量的修饰符，按Java Lauguage Specification中介绍的先后顺序排序。

9.除注释外的代码，不允许出现中文。经常看到log里面用中文，这样不允许，还有用中文直接equals比较，也同样不允许。

10.项目中禁止使用System.out.println()。如果有log需求，直接用log4j好了，可以随意指定输出位置。

11.@Override：@override 都应该使用。@override annotations只要是符合语法的，都应该使用。PS:如果没写Override intellij&eclipse默认都会有警告出现。

12.异常捕获，不应该被忽略：一般情况下，catch住的异常不应该被忽略，而是都需要做适当的处理。例如将错误日志打印出来，或者如果认为这种异常不会发生，则应该作为断言异常重新抛出。

13. 在测试类里，有时会针对方法是否会抛出指定的异常，这样的异常是可以被忽略的。但是这个异常通常需要命名为： expected。

14.重载toString方法：优先使用：ToStringBuilder(org.apache.commons.lang3.builder.ToStringBuilder)。

15.codereview规范

codereview：发布代码前在aone指定codereview的人，进行代码审查。但是代码审查并不是说只找一个人，这可以是团队多人一起做。进行reivew的人也不一定必须是老人，有时我们可能觉得“经验比较浅，不能对别人codereivew”，其实并不一定，俗话说：三个臭皮匠，顶个诸葛亮。即使是经验欠缺，多个人review，也能完成高质量的代码。