**(一) 命名风格**

1. 类名：类名使用UpperCameCase风格，遵从驼峰形式。

正例：MarcoPolo 反例：macroPolo

1. 方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从驼峰形式。

正例： localValue / getHttpMessage() / inputUserId

1. 常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

正例：MAX\_STOCK\_COUNT 反例：MAX\_COUNT

1. 抽象类命名使用 Abstract 或 Base 开头；异常类命名使用 Exception 结尾；测试类 命名以它要测试的类的名称开始，以 Test 结尾。
2. 如果模块、接口、类、方法使用了设计模式，在命名时体现出具体模式。

说明：将设计模式体现在名字中，有利于阅读者快速理解架构设计理念。

正例: public class OrderFactory;

public class LoginProxy;

public class ResourceObserver;

1. 枚举类名建议带上 Enum 后缀，枚举成员名称需要全大写，单词间用下划线隔开。

说明：枚举其实就是特殊的常量类，且构造方法被默认强制是私有。

正例：枚举名字为 ProcessStatusEnum 的成员名称：SUCCESS / UNKOWN\_REASON

**(二) 常量定义**

1. 不允许任何魔法值（即未经定义的常量）直接出现在代码中。

反例：String key = "Id#taobao\_";

1. long 或者 Long 初始赋值时，使用大写的 L，不能是小写的 l，小写容易跟数字 1 混 淆，造成误解

说明：Long a = 2l; 写的是数字的 21，还是 Long 型的 2?

**(三) 代码格式**

1. 大括号的使用约定。如果是大括号内为空，则简洁地写成{}即可，不需要换行；如果 是非空代码块则：

1） 左大括号前不换行。

2） 左大括号后换行。

3） 右大括号前换行。

4） 右大括号后还有 else 等代码则不换行；表示终止的右大括号后必须换行

1. 左小括号和字符之间不出现空格；同样，右小括号和字符之间也不出现空格。详见 第 5 条下方正例提示。

反例：if (空格 a == b 空格)

1. 注释的双斜线与注释内容之间有且仅有一个空格。
2. 方法参数在定义和传入时，多个参数逗号后边必须加空格。

正例：method("a", "b", "c");

最后：建议在每个类中写上main函数，其中包含用于单元测试的代码段