1. 抽象类以Abstract或Base开头，异常类使用Exception结尾，测试类以Test结尾。
2. 布尔类型的pojo属性，不要以is开头。
3. 将设计模式体现在接口、类、方法的命名上。
4. 接口中的方法和属性不要加任何修饰符号。（13）
5. 接口与其实现类的命名规则：

5.1 Service或DAO类，实现类一Impl结尾。

5.2 如果是形容能力的接口，通常以able结尾：AbstractTranslator 实现 Translatable 接口。

6、枚举要以Enum后缀，枚举成员名称全大写，单词之间用下划线分隔。

7、领域模型命名：（16）

1） 数据对象：xxxDO，xxx 即为数据表名。

2） 数据传输对象：xxxDTO，xxx 为业务领域相关的名称。

3） 展示对象：xxxVO，xxx 一般为网页名称。

4） POJO 是 DO/DTO/BO/VO 的统称，禁止命名成 xxxPOJO。

8、禁止任何魔法值（即未经预先定义的常量）直接出现在代码中。

9、常量类要按照功能来区分维护，如：缓存相关常量放在类 CacheConsts 下；系统配置相关常量放在类 ConfigConsts 下。（常量定义-3）

10、常量复用的5个层次：跨应用共享常量、应用内共享常量、子工程内共享常量、包

内共享常量、类内共享常量。

1） 跨应用共享常量：放置在二方库中，通常是 client.jar 中的 constant 目录下。

2） 应用内共享常量：放置在一方库中，通常是子模块中的 constant 目录下。

反例：易懂变量也要统一定义成应用内共享常量，两位攻城师在两个类中分别定义了表示“是”的变量：

类 A 中：public static final String YES = "yes";

阿里巴巴 Java 开发手册

类 B 中：public static final String YES = "y";

A.YES.equals(B.YES)，预期是 true，但实际返回为 false，导致线上问题。

3） 子工程内部共享常量：即在当前子工程的 constant 目录下。

4） 包内共享常量：即在当前包下单独的 constant 目录下。

5） 类内共享常量：直接在类内部 private static final 定义。

11、基本数据类型或包装数据类型的使用标准：

所有的pojo类属性必须用包装类型。

RPC方法的返回值和参数必须使用包装数据类型。

所有的局部变量使用基本数据类型。