

- 1.【强制】代码中的命名均不能以下划线或美元符号开始,也不能以下划线或美元符号结束。
- 2.【强制】代码中的命名严禁使用拼音与英文混合的方式,更不允许直接使用中文的方式。
- 3.【强制】类名使用 UpperCamelCase 风格,但一下情形例外: DO/BO/DTO/VO/AO/PO/UID 等。
- 4.【强制】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
- 5.【强制】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。
- 6.【强制】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。
- 7.【强制】类型与中括号紧挨相连来表示数组。
- 8.【强制】POJO 类中布尔类型变量都不要加 is 前缀,否则部分框架解析会引起序列化错误。
- 9.【强制】包名统一使用小写,点分隔符之间有且仅有一个自然语义的英文单词。包名统一使用单数形式,但是类名如果有复数含义,类名可以使用复数形式
- 10.【强制】避免在子父类的成员变量之间、或者不同代码块的局部变量之间采用完全相同的命名,使可读性降低。

说明:子类、父类成员变量名相同,即使是 public 类型的变量也是能够通过编译,而局部变量在同一个方法内的不同代码块中同名也是合法的,但是避免使用。对于非 setter/getter 的参数名称也要避免与成员变量名称相同。

- 11.【强制】杜绝完全不规范的缩写,避免望文不如其义。
- 12.【推荐】为了达到代码自解释的目标,任务自定义编程元素在命名时,使用尽量完整的单词组合来表达其意。
- 13.【推荐】在常量与变量的命名时,表示类型的名字放在词尾,以提升辨识度。
- 14.【推荐】如果模块、接口、类、方法使用了设计模式,在命名时需体现出具体模式。

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

- 15.【推荐】接口类中的方法和属性不要加任何修饰符(public 也不要加),保持代码的简洁性,并加上有效的 JavaDoc 注释。尽量不要在接口里定义变量,如果一定要定义变量,肯定与接口方法相关,并且是整个应用的基础常量。

正例:接口方法签名 void commit();

接口基础常量 String COMPANY = "alibaba";

反例:接口方法定义 public abstract void f();

- 16.接口和实现类的命名有两套规则:

- 1)【强制】对于 Service 和 DAO 类,基于 SOA 的理念,暴露出来的服务一定是接口,内部的实现类用 Impl 的后缀与接口区别
- 2)【推荐】如果是形容能力的接口名称,取对应的形容词为接口名(通常是-able 的形容词)。

- 17.【参考】枚举类名带上 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类,域成员均为常量,且构造方法被默认强制是私有。

正列:枚举名字为 ProcessStatusEnum 的成员名称: SUCCESS/ UNKNOWN_REASON

- 18.【参考】各层命名规约:

A) Service/DAO 层方法命名规约

- 1) 获取单个对象的方法用 get 做前缀。
- 2) 获取多个对象的方法用 list 做前缀,复数形式结尾如: listObjects。
- 3) 获取统计值的方法用 count 做前缀。

- 4) 插入的方法用 save/insert 做前缀。
 - 5) 删除的方法用 remove/delete 做前缀。
 - 6) 修改的方法用 update 做前缀。
- B) 领域模型命名规约
- 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。