代码规范

# 一、编程规约

## （一）命名风格

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，但以下情形例外：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。

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 类中布尔类型的变量，都不要加 is 前缀，否则部分框架解析会引起序列化错误。 反例：定义为基本数据类型 Boolean isDeleted 的属性，它的方法也是 isDeleted()，RPC框架在反向解析的时候，“误以为”对应的属性名称是 deleted，导致属性获取不到，进而抛 出异常。

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语单词。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】严禁不规范的缩写，避免望文不知义。

11.【推荐】为了达到代码自解释的目标，任何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。

12.【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。

13.【推荐】接口类中的方法和属性不要加任何修饰符号（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，肯定是 与接口方法相关，并且是整个应用的基础常量。

14.【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

15.【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 的形式）。

16.【参考】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。

17.【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。

## （二）、常量命名

1. 禁止魔法值

2. long类型的数值，使用L而非l以免造成混淆

3. 如果变量值仅在一个固定范围内变化用 enum 类型来定义。

public enum SeasonEnum {

SPRING(1), SUMMER(2), AUTUMN(3), WINTER(4);

private int seq;

SeasonEnum(int seq){

this.seq = seq;

}

}

## （三）、代码格式

1. 大括号

大括号内为空，简写成{}，无需换行；

非空代码块：

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。

2. 空格

1）括号与其内字符间无空格

2）保留字与括号间有空格

3）运算符左右添加空格

4）参数列表等号无需空格

5）使用4空格代替Tab：

IDEA禁用Use tab character ;

Eclipse 启用 insert spaces for tabs

6）注释符与内容间有一个空格

7）参数列表内，逗号后需加一个空格

3.换行

一行代码过长（80个字符）时需要换行，遵循以下原则：

逗号后换行；

操作符前换行；

括号前不换行；

换行后缩进4空格

4.空行

不同逻辑、不同语义、不同业务的代码之间插入一个空行分隔开来以提升可读性。

5. 编码尽可能简单清晰，禁止使用过于精简的代码而牺牲程序可读性

## （四）、控制语句

1. switch内，每个case必须有终止符break或return，每个switch必须包含一个default并放在最后，即使留空

2. 控制语句必须使用大括号，即使只有一行

3. 高并发场景下，使用区间判断而非等于，避免出现等值判断击穿现象

## （五）、注释

1. 类、类属性、类方法的注释必须使用 Javadoc 规范，使用/\*\*…\*/格式，不得使用 // xxx 方式。

2. 所有的抽象方法（包括接口中的方法）必须要用 Javadoc 注释、除了返回值、参数、 异常说明外，还必须指出该方法做什么事情，实现什么功能。

3. 所有的类都必须添加创建者和创建日期。

4. 注释单起一行，放在需注释代码上一行

5. 在功能和方法内部的块注释应该 和它们所描述的代码具有一样的缩进格式。

6. 块注释之首应该有一个空行，用于把块注释和代码分割开来

7. 块注释可以以/\*-开头，这样 indent(1)就可以将之识别为一个代码块的开始，而不会重排它。

8. 如果一个注释不能在一行内写 完，就该采用块注释

9. 单行注释之前应该有一个空行

10. 若有多个短注释出现于大段代码中，它们应该具有相同的缩进。