

1. 关于代码格式

【强制】在 long 或者 Long 赋值时，数值后使用大写的 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 2l，还是 Long 型的 2？

【强制】if/for/while/switch/do 等保留字与括号之间都必须加空格

【强制】任何二目、三目运算符的左右两边都需要加一个空格。说明：运算符包括赋值运算符=、逻辑运算符&&、加减乘除符号等

正例：（涉及 1-5 点）

```
public static void main(String[] args) {  
  
    // 缩进 4 个空格  
    String say = "hello";  
    // 运算符的左右必须有一个空格  
    int flag = 0;  
    // 关键词 if 与括号之间必须有一个空格，括号内的 f 与左括号，0 与右括号不需要空格  
    if (flag == 0) {  
        System.out.println(say);  
    }  
  
    // 左大括号前加空格且不换行；左大括号后换行  
    if (flag == 1) {  
        System.out.println("world");  
    // 右大括号前换行，右大括号后有 else，不用换行  
    } else {  
        System.out.println("ok");  
    // 在右大括号后直接结束，则必须换行  
    }  
}
```

【强制】注释的双斜线与注释内容之间有且仅有一个空格。

【强制】单行字符数限制不超过 120 个，超出需要换行，换行时遵循如下原则： 1) 第二行相对第一行缩进 4 个空格，从第三行开始，不再继续缩进，参考示例。 2) 运算符与下文一起换行。 3) 方法调用的点符号与下文一起换行。 4) 方法调用中的多个参数需要换行时，在逗号后进行。 5) 在括号前不要换行，见反例

【推荐】单个方法的总行数不超过 80 行。说明：包括方法签名、结束右大括号、方法内代码、注释、空行、回车及任何不可见字符的总行数不超过 80 行。正例：代码逻辑分清

红花和绿叶，个性和共性，绿叶逻辑单独出来成为额外方法，使主干代码 更加清晰；共性逻辑抽取成为共性方法，便于复用和维护

静态变量直接通过类名访问，拒绝通过实例对象访问。

所有的覆写方法，必须加@Override 注解。

【强制】外部正在调用或者二方库依赖的接口，不允许修改方法签名，避免对接口调用方产生影响。接口过时必须加@Deprecated 注解，并清晰地说明采用的新接口或者新服务是什么。

【强制】Object 的 equals 方法容易抛空指针异常，应使用常量或确定有值的对象来调用 equals。 正例：`"test".equals(object)`；反例：`object.equals("test")`；

关于基本数据类型与包装数据类型的使用标准如下：

- 1) **【强制】**所有的 POJO 类属性必须使用包装数据类型。
- 2) **【强制】**RPC 方法的返回值和参数必须使用包装数据类型。
- 3) **【推荐】**所有的局部变量使用基本数据类型。

说明：POJO 类属性没有初值是提醒使用者在需要使用时，必须自己显式地进行赋值，任何 NPE 问题，或者入库检查，都由使用者来保证。

正例：数据库的查询结果可能是 null，因为自动拆箱，用基本数据类型接收有 NPE 风险。

反例：比如显示成交总额涨跌情况，即正负 x%，x 为基本数据类型，调用的 RPC 服务，调用不成功时，返回的是默认值，页面显示为 0%，这是不合理的，应该显示成中划线。所以包装 数据类型的 null 值，能够表示额外的信息，如：远程调用失败，异常退出。

【强制】定义 DO/DTO/VO 等 POJO 类时，不要设定任何属性**默认值**。反例：POJO 类的 getCreate 默认值为 `new Date()`，但是这个属性在数据提取时并没有置入具体值，在更新其它字段时又附带更新了此字段，导致创建时间被修改成当前时间

【强制】构造方法里面禁止加入任何业务逻辑，如果有初始化逻辑，请放在 init 方法中

如果代码中使用了某种设计模式，类的命名时最好体现出设计模式。

接口中定义方法最好不要加上public，保持代码简洁性。

不允许任何未经定义的常量出现在代码中。

常量的定义，一定要分类，分作用域，模块内的就定义在该模块的constant包中。整个应用的就定义在应用下面的包里。类的就定义在类内（public static final），包就定义在该包下。

如果变量值在一定范围内变化，且带有名称之外的延展值，必须定义成枚举类型，enum。

entity与数据库相对应的实体类中最好不要有默认值，比如：某个字段为创建时间，假设默认为当前时间，可能在做更新操作时，更新了这个字段，就会导致错误。

POJO类中必须写toString方法，便于排查问题。

类中方法的顺序是：构造方法在前>公有方法>保护方法>私有方法。因为构造方法，公有方法才是大家最关心的。私有方法只是黑盒的实现。

循环内的字符串连接，必须使用StringBuilder，否则每+都会生成1个新的对象。

类中的方法、变量访问必须控制从严，

1. 构造函数如果不允许直接new的，必须是private的。
2. 类中非static变量且与子类共享的则必须是protected的。
3. 本类使用的静态变量、或其他变量必须是private的。
4. static的变量必须考虑是否是final的。

总之做到一点，该类中的变量或方法在什么作用域内，则控制权限不能多放，否则不容易模块之间解耦。private的方法、变量想删除就可以删除，而public删除时必须小心。

看到了第7页第10条。序列化