**java语言技术管理文档**

强制：

1. 代码中的命名均不能以”\_”或”$”开始，也不能以”\_”或”$”结束
2. 代码中的命名严禁使用拼音
3. 类名采用大驼峰命名法
4. 方法名、参数名、成员变量、局部变量都采用小驼峰命名法
5. 常量命名全部大写，单词间以下划线隔开
6. 抽象类命名使用Abstract或Base开头；异常类命名使用Exception结尾；测试类命名以被测试的类的名称开始，以Test结尾
7. 包名统一使用反向域名
8. POJO类中的Boolean变量严禁以is作为前缀
9. if、for、while、switch、do等保留字与括号之间都必须加空格
10. 如果是大括号内为空，则简洁地写成{}即可；如果是非空代码块，左大括号前不换行，左大括号后换行，右大括号前换行，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
11. 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
12. if、else、for、while、do语句中必须使用大括号，即使代码只有一行也要添加
13. 在一个switch块内，每个case要么通过break/return等来终止，要么注释说明程序将继续执行到哪一个case为止；在一个switch块内，都必须包含一个default语句并且放在最后，即使它什么代码也没有
14. 任何二目、三目运算符的左右两边都需要加一个空格
15. 单行字符数限制不超过120个，超出需要换行
16. 方法参数在定义和传入时，多个参数逗号后边必须加空格
17. 只要重写equals，就必须重写hashCode
18. ArrayList的subList结果不可强转成ArrayList，否则会抛出ClassCastException异常
19. 线程池不允许使用 Executors 去创建，而是通过 ThreadPoolExecutor 的方式
20. 类、类属性、类方法的注释必须使用 Javadoc 规范，使用/\*\*内容\*/格式，不得使用//xxx方式
21. 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
22. 所有枚举类型字段必须要有注释，说明每个数据项的用途
23. RuntimeException可以通过预先检查进行规避，而不应该通过catch来处理
24. 有try块放到了事务代码中，catch异常后，如果需要回滚事务，一定要手动回滚事务
25. 数据库的表名字段名只能使用小写字母命名，单词间以下划线隔开
26. 如果存储的字符串长度几乎相等，使用char定长字符串类型

推荐：

1. 接口类中的方法和属性不要加任何修饰符号，保持代码的简洁性，并加上有效的Javadoc注释
2. 如果使用到了设计模式，建议在类名中体现出具体模式
3. 方法体内的执行语句组、变量的定义语句组、不同的业务逻辑之间或者不同的语义之间插入一个空行，以增强代码的可读性
4. 使用索引访问用String的split方法得到的数组时，需做最后一个分隔符后有无内容的检查
5. POJO类必须写toString方法
6. 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
7. 集合初始化时，根据预估的元素数量指定集合初始值大小
8. 使用entrySet遍历Map类集合的键值对，而不是使用keySet方式进行遍历
9. 并发修改同一记录时，避免更新丢失，需要加锁
10. 除常用方法（如getXxx、isXxx）外，不要在条件判断中执行复杂语句。应将复杂逻辑判断结果赋值给有意义的布尔变量名，提高可读性
11. 表的命名建议采用“业务名称\_表的作用”方式

允许：

1. 在特定的框架或库中，如果约定俗成的命名方式与该文档命名规则有冲突，允许遵循该框架或库的命名规范
2. 在一些过于简单的情况下，如单行的 if 语句，允许使用简化格式，如if (condition) statement; 但需保证代码的可读性
3. 在一些简单的代码块中，如单行的变量定义，允许使用简短的注释，但需保证注释的清晰性
4. 在一些特殊的情况下，如历史遗留的数据库表结构，允许对命名规则进行适当的调整，但需在代码注释中说明原因

参考：google和阿里的规范