**java代码评审标准**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 总 则 条 款 | **执行情况** | **说明** |
| **1 命名规则** | | | |
| 1 | 1.1 命名规则是否与所采用的规范保持一致 | 是[  ]  否[  ]  免[  ] |  |
| 2 | 1.2 是否遵循了最小长度最多信息原则 | 是[  ]  否[  ]  免[  ] |  |
| 3 | 1.3 has/can/is前缀的函数是否返回布尔型 | 是[  ]  否[  ]  免[  ] |  |
| **2 注 释** | | | |
| 4 | 2.1注释是否较清晰且必要 | 是[  ]  否[  ]  免[  ] |  |
| 5 | 2.2复杂的分支流程是否已经被注释 | 是[  ]  否[  ]  免[  ] |  |
| 6 | 2.3距离较远的}是否已经被注释 | 是[  ]  否[  ]  免[  ] |  |
| 7 | 2.4非通用变量是否全部被注释 | 是[  ]  否[  ]  免[  ] |  |
| 8 | 2.5函数是否已经有文档注释 | 是[  ]  否[  ]  免[  ] |  |
| 9 | 2.6特殊用法是否被注释 | 是[  ]  否[  ]  免[  ] |  |
| **3  声明 空白 缩进** | | | |
| 10 | 3.1每行是否只声明了一个变量（特别是那些可能出错的类型） | 是[  ]  否[  ]  免[  ] |  |
| 11 | 3.2变量是否已经在定义的同时初始化 | 是[  ]  否[  ]  免[  ] |  |
| 12 | 3.3类属性是否都执行了初始化 | 是[  ]  否[  ]  免[  ] |  |
| 13 | 3.4代码段落是否被合适地以空行分隔 | 是[  ]  否[  ]  免[  ] |  |
| 14 | 3.5是否合理地使用了空格使程序更清晰 | 是[  ]  否[  ]  免[  ] |  |
| 15 | 3.6代码行长度是否在要求之内 | 是[  ]  否[  ]  免[  ] |  |
| 16 | 3.7折行是否恰当 | 是[  ]  否[  ]  免[  ] |  |
| **4 语句/功能划分/规模** | | | |
| 17 | 4.1包含复合语句的{}是否成对出现并符合规范 | 是[  ]  否[  ]  免[  ] |  |
| 18 | 4.2是否给单个的循环、条件语句也加了{} | 是[  ]  否[  ]  免[  ] |  |
| 19 | 4.3 if/if-else/if-else if-else/do-while/switch-case语句的格式是否符合规范 | 是[  ]  否[  ]  免[  ] |  |
| 20 | 4.4单个变量是否只做单个用途 | 是[  ]  否[  ]  免[  ] |  |
| 21 | 4.5单行是否只有单个功能（不要使用；进行多行合并） | 是[  ]  否[  ]  免[  ] |  |
| 22 | 4.6单个函数是否执行了单个功能并与其命名相符 | 是[  ]  否[  ]  免[  ] |  |
| 23 | 4.7操作符++和--操作符的应用是否复合规范 | 是[  ]  否[  ]  免[  ] |  |
| 24 | 4.8单个函数不超过规定行数 | 是[  ]  否[  ]  免[  ] |  |
| 25 | 4.9缩进层数是否不超过规定 | 是[  ]  否[  ]  免[  ] |  |
| **5 可靠性（总则/变量和语句）** | | | |
| 26 | 5.1是否已经消除了所有警告 | 是[  ]  否[  ]  免[  ] |  |
| 27 | 5.2常数变量是否声明为final | 是[  ]  否[  ]  免[  ] |  |
| 28 | 5.3对象使用前是否进行了检查 | 是[  ]  否[  ]  免[  ] |  |
| 29 | 5.4局部对象变量使用后是否被复位为NULL | 是[  ]  否[  ]  免[  ] |  |
| 30 | 5.5对数组的访问是否是安全的（合法的index取值为[0, MAX\_SIZE-1]） | 是[  ]  否[  ]  免[  ] |  |
| 31 | 5.6是否确认没有同名变量局部重复定义问题 | 是[  ]  否[  ]  免[  ] |  |
| 32 | 5.7程序中是否只使用了简单的表达式 | 是[  ]  否[  ]  免[  ] |  |
| 33 | 5.8是否已经用（）使操作符优先级明确化 | 是[  ]  否[  ]  免[  ] |  |
| 34 | 5.9所有判断是否都使用了（常量==变量）的形式 | 是[  ]  否[  ]  免[  ] |  |
| 35 | 5.10是否消除了流程悬挂 | 是[  ]  否[  ]  免[  ] |  |
| 36 | 5.11是否每个if-else if-else语句都有最后一个else以确保处理了全集 | 是[  ]  否[  ]  免[  ] |  |
| 37 | 5.12是否每个switch-case语句都有最后一个default以确保处理了全集 | 是[  ]  否[  ]  免[  ] |  |
| 38 | 5.13 for循环是否都使用了包含下限不包含上限的形式（k=0; k<MAX） | 是[  ]  否[  ]  免[  ] |  |
| 39 | 5.14 XML标记书写是否完整，字符串的拼写是否正确 | 是[  ]  否[  ]  免[  ] |  |
| 40 | 5.15对于流操作代码的异常捕获是否有finally操作以关闭流对象 | 是[  ]  否[  ]  免[  ] |  |
| 41 | 5.16退出代码段时是否对临时对象做了释放处理 | 是[  ]  否[  ]  免[  ] |  |
| 42 | 5.17对浮点数值的相等判断是否是恰当的（严禁使用==直接判断） | 是[  ]  否[  ]  免[  ] |  |
| **6 可靠性（函数）** | | | |
| 43 | 6.1入口对象是否都被进行了判断不为空 | 是[  ]  否[  ]  免[  ] |  |
| 44 | 6.2入口数据的合法范围是否都被进行了判断(尤其是数组) | 是[  ]  否[  ]  免[  ] |  |
| 45 | 6.3是否对有异常抛出的方法都执行了try...catch保护 | 是[  ]  否[  ]  免[  ] |  |
| 46 | 6.4是否函数的所有分支都有返回值 | 是[  ]  否[  ]  免[  ] |  |
| 47 | 6.5 int的返回值是否合理（负值为失败，非负值成功） | 是[  ]  否[  ]  免[  ] |  |
| 48 | 6.6对于反复进行了int返回值判断是否定义了函数来处理 | 是[  ]  否[  ]  免[  ] |  |
| 49 | 6.7关键代码是否做了捕获异常处理 | 是[  ]  否[  ]  免[  ] |  |
| 50 | 6.8是否确保函数返回CORBA对象的任何一个属性都不能为null | 是[  ]  否[  ]  免[  ] |  |
| 51 | 6.9是否对方法返回值对象做了null检查，该返回值定义时是否被初始化 | 是[  ]  否[  ]  免[  ] |  |
| 52 | 6.10是否对同步对象的遍历访问做了代码同步 | 是[  ]  否[  ]  免[  ] |  |
| 53 | 6.11是否确认在对Map对象使用迭代遍历过程中没有做增减元素操作 | 是[  ]  否[  ]  免[  ] |  |
| 54 | 6.12线程处理函数循环内部是否有异常捕获处理，防止线程抛出异常而退出 | 是[  ]  否[  ]  免[  ] |  |
| 55 | 6.13原子操作代码异常中断，使用的相关外部变量是否恢复先前状态 | 是[  ]  否[  ]  免[  ] |  |
| 56 | 6.14函数对错误的处理是恰当的 | 是[  ]  否[  ]  免[  ] |  |
| **7 可维护性** | | | |
| 57 | 7.1实现代码中是否消除了直接常量（用于计数起点的简单常数例外） | 是[  ]  否[  ]  免[  ] |  |
| 58 | 7.2是否消除了导致结构模糊的连续赋值（如a= (b=d+c )） | 是[  ]  否[  ]  免[  ] |  |
| 59 | 7.3是否每个return前都要有日志记录 | 是[  ]  否[  ]  免[  ] |  |
| 60 | 7.4是否有冗余判断语句（如：if (b) return true; else return false;） | 是[  ]  否[  ]  免[  ] |  |
| 61 | 7.5是否把方法中的重复代码抽象成私有函数 | 是[  ]  否[  ]  免[  ] |  |
|  |  |  |  |