**Java代码评审标准**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 目的 | 指导你进行有效的代码复查 | # | # | # | 累计 | 累计% |
| 一般性说明 | 在完成每个复查步骤之后，将发现的某个类型的缺陷的个数记录在右面的栏目中。如果该步骤没有发现缺陷，就在右面的栏目中打个表示检查无误的交叉符号（X）。在开始复查下一个程序单元之前，要按照检查表完成对程序、类、对象或方法的检查。 |  |  |  |  |  |
| 完整性 | 验证设计的所有功能是否都已经编码 |  |  |  |  |  |
| 命名 | 1. 命名规则是否与所采用的规范保持一致 2. 命名拼写是否前后一致 |  |  |  |  |  |
| 注释 | 注释是否较清晰且必要 |  |  |  |  |  |
| 初始化 | 检查变量和参数的初始化：  1.在程序的开始；  2.在每个循环的开始；  3.在函数/过程的入口 |  |  |  |  |  |
| 调用 | 检查函数调用的格式：  参数的使用 |  |  |  |  |  |
| 函数 | 1.单个函数是否执行了单个功能并与其命名相符  2.函数对错误的处理是否是恰当的  3.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? |  |  |  |  |  |
| 语句 | 1. 包含复合语句的{}是否成对出现并符合规范 2. 是否给单个的循环、条件语句也加了{} 3. if/if-else/if-else if-else/do-while/switch-case语句的格式是否符合规范？ |  |  |  |  |  |
| 输出格式 | 检查输出格式：  1.换行是否合适？  2.间隔是否合适？ |  |  |  |  |  |
| 警告 | 是否已经消除了所有警告 |  |  |  |  |  |
| 数组 | 对数组的访问是否是安全的？（合法的index取值为[0, MAX\_SIZE-1]）。 |  |  |  |  |  |
| 逐行检查 | 检查每一行代码：  1.语法是否正确？  2.标点是否正确、符号是否正确？  3.关键代码是否做了捕获异常处理? |  |  |  |  |  |
| 标准 | 保证所有代码符合编程标准 |  |  |  |  |  |
| 文件的打开与关闭 | 验证所有的文件：  1.是合适地声明的；  2.是合适地打开的；  3.是合适地关闭的； |  |  |  |  |  |
| 前面检查 | 前面检查以发现系统问题和非期望 |  |  |  |  |  |
| 总计 |  |  |  |  |  |  |