**代码评审标准**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目名称 |  | | | | | | | | |
| 填写人 |  | | | | 填写日期 |  | | | |
| 编号 | 问 题 | | | 类别 | | | 项目总分 | 得分 | 说明 |
| **一、变量,属性和常量声明缺陷(VC)** | | | | **基础** | | | **6** |  |  |
| 1 | 变量和常量的命名是否与约定保持一致? | | | | | | | |
| 2 | 是否存在容易混淆的相似的变量和属性名? | | | | | | | |
| 3 | 变量和属性是否书写正确? | | | | | | | |
| 4 | 变量和属性是否被正确的初始化? | | | | | | | |
| 5 | 非局部变量是否能用局部变量替换? | | | | | | | |
| 6 | 所有的属性是否都有正确的访问限制符? | | | | | | | |
| **二、方法定义缺陷(FD)** | | | | **基础** | | | **4** |  |  |
| 1 | 方法名的描述方法是否与命名约定一致? | | | | | | | |
| 2 | 每种方法是否都有正确的访问限制符? | | | | | | | |
| 3 | 每个方法的参数值在使用之前是否都做了检查? | | | | | | | |
| 4 | 对于每一个方法，是否都返回了正确的值? | | | | | | | |  |
| **三、类定义缺陷** | | | | **基础** | | | **6** |  |  |
| 1 | 每一个类是否都有正确的构造函数? | | | | | | | |
| 2 | 在子类中是否有应该放到父类中的通用成员? | | | | | | | |
| 3 | 类的继承层次是否能被简化? | | | | | | | |
| **四、数据引用缺陷** | | | | **基础** | | | **4** |  |  |
| 1 | 对于每一个数组引用,下标值是否在定义的范围内? | | | | | | | |
| 2 | 对于对象和数组引用,是否组确定其值应为非空? | | | | | | | |
| **五、计算/数值缺陷** | | | | **基础** | | | **5** |  |  |
| 1 | 是否存在不同类型数据之间的混合计算? | | | | | | | |
| 2 | 关于数值计算的顺序和优先级的假设是否正确? | | | | | | | |
| 3 | 代码是否使用括号来避免模糊不清? | | | | | | | |
| 4 | 数值类型声明以及精度控制是否合理? | | | | | | | |
| 5 | 数值计算是否存在溢出的可能性? | | | | | | | |
| **六、比较/关系缺陷** | | | | **基础** | | | **5** |  |  |
| 1 | | 对每一个布尔测试，正确条件是否被检查? | | | | | | |  |
| 2 | | 比较操作符是否正确? | | | | | | |  |
| 3 | | 布尔表达式是否通过内部否定操作进行了简化? | | | | | | |  |
| 4 | | 每个布尔表达式是否正确? | | | | | | |  |
| 5 | | “&&”是否被不小心替换为“&”? “||”是否被不小心替换为“|”? | | | | | | |  |
| **七、流程控制缺陷** | | | | **基础** | | | **10** |  |  |
| 1 | 对于每一个循环:是否选用了最佳的循环结构? | | | | | | | |
| 2 | 所有的循环是否都能结束? | | | | | | | |
| 3 | 如果一个循环有多个出口,是否每个出口都有必要并且得到正确处理? | | | | | | | |
| 4 | switch声明是否都有default条件? | | | | | | | |
| 5 | 循环和分支的嵌套是否过深?是否正确? | | | | | | | |
| 6 | 空控制条件是否都正确，并加上括号及批注? | | | | | | | |
| 7 | 所有异常是否得到了正确的处理? | | | | | | | |  |
| **八、输入输出缺陷** | | | | **基础** | | | **8** |  |  |
| 1 | 文件在被使用之前是否都被打开? | | | | | | | |
| 2 | 输入对象的属性是否与使用的文件一致? | | | | | | | |
| 3 | 在操作文件时是否判断了文件存在与否? | | | | | | | |
| 4 | 文件在被使用之后是否都被关闭? | | | | | | | |
| **九、注释缺陷** | | | | **基础** | | | **10** |  |  |
| 1 | 每一个方法，类和文件是否都有适当的头注释? | | | | | | | |
| 2 | 每一个属性，变量和常量的声明是否都有注释? | | | | | | | |
| 3 | 每个类和方法的潜在行为是否都有用简易的语言进行解释? | | | | | | | |
| 4 | 方法和类的头注释是否和它们的功能保持一致? | | | | | | | |
| 5 | 注释和代码是否保持一致? | | | | | | | |
| 6 | 注释对于理解代码是否有帮助？ | | | | | | | |  |
| **十、程序结构** | | | | **重点** | | | **12** |  |  |
| 1 | 代码是否正确完整的实现了设计? | | | | | | | |
| 2 | 代码中有没有被调用的或无用的程序,或没有被执行的代码，或者测试代码？ | | | | | | | |
| 3 | 是否存在能被调用外部复用组件或库函数替代的代码? | | | | | | | |
| 4 | 有没有能被压缩成简单程序的程序块或重复的代码? | | | | | | | |
| 5 | 内存及其它资源是否被有效利用? | | | | | | | |
| 6 | 是否有过于复杂的模块需要重新构造或拆分成多个程序? | | | | | | | |
| **十一、布局和封包缺陷** | | | | **基础** | | | **10** |  |  |
| 1 | 代码布局格式和缩排标准是否前后一致? | | | | | | | |
| 2 | 对于每一个方法,它的代码量是否都不超过规定行数? | | | | | | | |
| 3 | 对于每一个编译模块,它的代码量是否都不超过规定行数? | | | | | | | |
| **十二、模块性缺陷** | | | | **重点** | | | **10** |  |  |
| 1 | 模块(方法,类)之间是否具有低偶合性? | | | | | | | |
| 2 | 每个模块(方法,类)自身是否具有高聚合性? | | | | | | | |
| 3 | Java类库的使用是否合理? | | | | | | | |
| **十三、性能缺陷 (PE)** | | | | **重点** | | | **10** |  |  |
| 1 | 是否有更好的数据结构和算法可以采用? | | | | | | | |
| 2 | 是否可以通过对数值进行一次计算并将结果保存来减少对它重新计算带来的消耗? | | | | | | | |
| 3 | 每一个计算出并保存了的结果是否都被应用? | | | | | | | |
| 4 | 计算是否能被移到循环之外?是否在循环内频繁创建对象？ | | | | | | | |
| 5 | 在循环内是否有不需要的判断? | | | | | | | |
| **填写人** | | |  | | | | | **总分** |  |
| **评论** | | |  | | | | | | |
|