|  |  |  |  |
| --- | --- | --- | --- |
| **代码评审标准表** | | | |
| 编号 | 问题 | 类别 | 项目总分 |
| 一、变量，属性和常量声明缺陷 | | 基础 | 5 |
| 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代？ | | |
| 6 | 所有的for循环的控制变量是否都在循环顶部被声明？ | | |
| 7 | 是否有应该命名为常量的文字常量？ | | |
| 8 | 变量和属性是否可以用常量替换？ | | |
| 9 | 属性是否可以用本地变量？ | | |
| 10 | 所有的属性是否都有正确的访问限制符（private、protected、public）？ | | |
| 11 | 是否有静态属性应该是非静态（是否有非静态属性应该是静态）？ | | |
| 二、方法定义缺陷 | | 基础 | 5 |
| 1 | 方法名的描述方法是否与命名约定一致？ | | |
| 2 | 每个方法的参数值在使用之前是否都作了检查？ | | |
| 3 | 对于每一个方法，它是否都返回了正确的值？ | | |
| 4 | 每种方法是否都有正确的访问限制符（private、protected、public）？ | | |
| 5 | 静态方法是否应该为非静态（反之亦然）？ | | |
| 6 | 接口方法的声明是否正确的表达语义？是否正确的给出返回值或抛出异常？ | | |
| 三、类定义缺陷 | | 基础 | 5 |
| 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 | 5 |
| 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，是否确定其值应为非空？ | | |
| 3 | 是否存在大内存对象的频繁创建或引用？ | | |
| 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运算？ | | |
| 2 | 在计算机中是否存在上溢或下溢的可能？ | | |
| 3 | 关于数值计算的顺序和优先级的假设是否正确？ | | |
| 4 | 代码是否避免了对浮点型数值的相等比较操作？ | | |
| 5 | 被除数是否做了零值测试？ | | |
| 6 | 是否用了括号来避免模糊不清？ | | |
| 7 | 数值类型声明以及精度控制是否合理？ | | |
| 8 | 数值计算是否存在溢出的可能性？ | | |
| 六、流程控制缺陷 | | 基础 | 5 |
| 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 | 是否所有的case-switch-break对应关系都已更正并加上批注？ | | |
| 6 | 是否named break叙述都跳到正确的地方？ | | |
| 7 | 循环和分支的嵌套是否过深?是否正确? | | |
| 8 | 是否有if嵌套可以转换程switch嵌套? | | |
| 9 | 空控制条件是否都正确，并加上括号及批注？ | | |
| 10 | 所有的异常是否都得到了正确的处理？ | | |
| 11 | 每一个方法是否都能结束? | | |
| 七、输入输出缺陷 | | 基础 | 5 |
| 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? | | |
| 5 | 网络连接是否合理池化(反之亦然)？ | | |
| 6 | 网络程序是否有合理的异常处理以及方法的异常声明？ | | |
| 7 | 网络连接是否正常关闭？ | | |
| 八、注释缺陷 | | 基础 | 5 |
|  | 每一个方法,类和文件是否都有适当的头注释? | | |
|  | 变量和常量的声明是否都有注释? | | |
|  | 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? | | |
|  | 方法和类的头注释是否和它们的功能保持一致? | | |
|  | 注释和代码是否保持一致？ | | |
|  | 注释对于理解代码是否有帮助？ | | |
|  | 代码中的注释是否充分？ | | |
|  | 代码中的注释是否过多？ | | |
| 九、程序结构 | | 重点 | 10 |
|  | 代码是否正确完整地实现了设计？ | | |
|  | 代码是否符合相关的编码标准？ | | |
|  | 代码结构是否适当，风格和格式是否保持一致？ | | |
|  | 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？ | | |
|  | 代码中是否还有多余的脏程序或测试代码？ | | |
|  | 是否存在能被调用外部复用组件或库函数替代的代码？ | | |
|  | 有没有能被压缩成简单程序的程序块或重复的代码？ | | |
|  | 内存及其他资源是否被有效利用？ | | |
|  | 数字和字符串常量是否用符号代替？ | | |
|  | 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、布局和封包缺陷 | | 基础 | 5 |
|  | 代码布局格式和缩排标准是否前后一致？ | | |
|  | 对于每一个方法，它的代码量是否都不超过规定行数？ | | |
|  | 对于每一个编译模块，它的代码量是否都不超过规定行数？ | | |
| 十一、模块性缺陷 | | 重点 | 10 |
|  | 模块（方法，类）之间是否具有低耦合性？ | | |
|  | 每个模块（方法，类）自身是否具有高聚合性？ | | |
|  | 是否存在重复的代码，它的功能可以通过调用其他的方法来实现？ | | |
|  | Java类库的使用是否合理？ | | |
| 十二、存储器使用缺陷 | | 基础 | 5 |
|  | 数组是否过大？ | | |
|  | 数组和对象不再使用之后，它们的引用是否被赋为空值？ | | |
| 十三、防御性编程 | | 重点 | 10 |
|  | 索引和下标是否经过了数组，记录或文件的边界测试？ | | |
|  | 是否验证了导入的数据或输入的参数的正确性和完整性？ | | |
|  | 所有的输出变量是否都被赋值？ | | |
|  | 在每个声明中的数据是否被正确操作？ | | |
|  | 分配的内存空间是否都被释放？ | | |
|  | 对于服务调用是否有超时设计或异常处理？ | | |
|  | 在程序结束的时候所有的文件或服务是否都保持了正确的状态？ | | |
| 十四、性能缺陷 | | 重点 | 10 |
|  | 是否有更好的数据结构和算法可以采用？ | | |
|  | 测试安排是否合理，使易于通过的且代价低廉的测试优先于代价较高且通过频率较第的测试？ | | |
|  | 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？ | | |
|  | 每一个计算出并保存了的结果是否都被应用？ | | |
|  | 计算是否能被移到循环之外？是否在循环内频繁创建对象？ | | |
|  | 在循环内是否有不需要的判断？ | | |
|  | 短循环是否可以取消？ | | |
|  | 对同一个数据进行操作的两个循环是否可以合并成一个？ | | |
|  | 长连接和短连接是否使用得当？ | | |
| 十五、程序安全 | | 重点 | 10 |
|  | 是否存在sql注入？ | | |
|  | 是否存在跨站脚本攻击？ | | |
|  | 异常页是否泄露数据库字段等存在安全问题的信息？ | | |
|  | Cookie中是否存在导致账号被盗等安全问题的信息？ | | |
| 十六、架构设计/程序设计 | | 附加 | 10 |
|  | 是否遵循开闭原则？ | | |
|  | 是否合理使用设计模式？ | | |
|  | 业务开发的可扩展性、复用性以及组件化 | | |
|  | 重大历史遗留的解决以及技术突破 | | |