# 代码评审标准

|  |  |
| --- | --- |
| **项目名称** |  |
| **评审时间** |  |
| **评审结果等级** | 很好：1 好：0.75 一般：0.5 不通过：0 |

|  |  |
| --- | --- |
| **常规项** | **结果** |
| 程序是否能够正常工作？ |  |
| 程序是否实现预期的功能，逻辑是否正确？ |  |
| 代码是否符合自己所遵循的编程规范？（包括大括号的位置，变量名和函数名、注释等） |  |
| 程序是否存在多余的或是重复的代码？ |  |
| 程序是否尽可能的进行了模块化？ |  |
| 复杂程序是否合理地分解为子程序？ |  |
| 是否有可以删除的日志或调试代码？ |  |
| 是否有可以被库函数替代的代码？ |  |
| 异常处理是否符合规范？ |  |
| 若存在未完成代码，是否进行了移除，或者用合适的标记进行了标注？（比如TODO） |  |

|  |  |
| --- | --- |
| **命名** | **结果** |
| 命名规则是否与所采用的规范保持一致？ |  |
| 是否遵循了最小长度最多信息原则？ |  |
| has/can/is前缀的函数是否返回布尔型？ |  |
| 类名是否存在重名问题？ |  |

|  |  |
| --- | --- |
| **语句/功能分布/规模** | **结果** |
| 是否给单个的循环、条件语句也加了{}？ |  |
| 操作符＋＋和— —操作符的应用是否符合规范？ |  |
| 单个变量是否只做单个用途？ |  |
| 单行是否只有单个功能？（不要使用；进行多行合并） |  |
| 单个函数是否执行了单个功能并与其命名相符？ |  |
| 包含复合语句的{}是否成对出现并符合规范？ |  |
| 缩进层数是否不超过规定？ |  |
| 单个函数不超过规定行数？ |  |

|  |  |
| --- | --- |
| **SQL** | **结果** |
| SQL语句是否小写？ |  |
| 引用符号是否使用单了引号？ |  |
| 严禁使用select \* 形式的语句，必须指出具体字段 |  |
| 严禁使用insert into table values(?,?,?)，必须指出具体要赋值的字段 |  |
| 禁止在一条SQL语句中使用3层以上的嵌套 |  |
| 如果有多表连接时，应该有主从之分，尽量从一个表取数 |  |
| 字符串连接时必须使用“||” |  |
| SQL中函数是否在指定范围内？ |  |
| 对重要的计算应说明其功能。、 |  |

|  |  |
| --- | --- |
| **算法** | **结果** |
| 数据结构是否合理、精简？ |  |
| 算法是否可以独立测试，是否与数据库和其他算法隔？ |  |
| 是否尽量避免了嵌套的运用？ |  |
| 复杂逻辑是否进行了必要而充分的注释？ |  |

|  |  |
| --- | --- |
| **可靠性（总则/变量和语句）** | **结果** |
| 是否已经消除了所有警告？ |  |
| 常数变量是否声明为final？ |  |
| 对象使用前是否进行了检查？ |  |
| 成员变量，局部变量是否在使用前被赋值？ |  |
| 局部对象变量使用后是否被复位为NULL？ |  |
| 对数组的访问是否是安全的？（合法的index取值为[0, MAX\_SIZE-1]）。 |  |
| 是否确认没有同名变量局部重复定义问题? |  |
| 程序中是否只使用了简单的表达式？ |  |
| 是否已经用（）使操作符优先级明确化？ |  |
| 所有判断是否都使用了（常量==变量或者常量.equals(变量)）的形式？ |  |
| 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？ |  |
| 是否每个switch-case语句都有最后一个default以确保处理了全集？ |  |
| XML标记书写是否完整，字符串的拼写是否正确? |  |
| 对于流操作代码的异常捕获是否有finally操作以关闭流对象? |  |
| 方法调用前是否有非空判断？ |  |

|  |  |
| --- | --- |
| **安全** | **结果** |
| 所有的数据输入是否都进行了检查（检测其类型、长度、格式和范围）并且编码？ |  |
| 输出的数值是否进行了检查并且编码？ |  |
| 无效的参数值是否能够进行了处理？ |  |
| 在哪里使用了第三方工具，返回的错误是否能够被捕获？ |  |
| 是否对非常规行为和边界情况进行了处理？ |  |

|  |  |
| --- | --- |
| **测试** | **结果** |
| 代码是否可以测试。比如，不要添加太多的或是隐藏的依赖关系，不能够初始化对象，测试框架可以使用方法等。 |  |
| 是否存在测试，它们是否可以被理解。比如，至少达到你满意的代码覆盖(code coverage) |  |
| 单元测试是否真正的测试了代码是否可以完成预期的功能？ |  |
| 是否有可以被已经存在的API所替代的测试代码？ |  |
| 是否检查了数组的“越界“错误？ |  |

|  |  |
| --- | --- |
| **文档** | **结果** |
| 是否有注释，并且描述了代码的意图？ |  |
| 对所有的函数是否都有注释，标注其功能及用处？ |  |
| 对非常规行为和边界情况的处理是否有描述？ |  |
| 第三方库的使用和函数是否有文档？ |  |
| 数据结构和计量单位是否进行了解释？ |  |