代码评审清单（Code Review Checklist）

版本：1.0

产品/项目组名称： XXX 产品/项目名称： XX 版本号： X.X

被检查人签字： \_\_\_\_\_\_\_\_\_\_\_ 检查内容：\_\_\_ \_\_\_\_\_\_

检查人签字 ：\_\_\_\_\_\_\_\_\_\_\_ 检查日期： \_ \_\_\_\_

说明：

类清单

|  |  |  |
| --- | --- | --- |
| 被检查类名 | 模块名称 | 项目名称 |
|  |  |  |
|  |  |  |
|  |  |  |

检查项

|  |  |  |
| --- | --- | --- |
| 常规 （15分） | 是 | 否 |
| 程序是否明确地注释了其实现功能 |  |  |
| 注释格式是否正确 |  |  |
| 程序单元是否承担单一职责 |  |  |
| 复杂程序是否合理地分解为子程序 |  |  |
| 程序中接口定义是否明晰 |  |  |
| 异常处理是否符合规范 |  |  |
| 包名、类名命名是否符合规范 |  |  |
| 方法名命名是否符合规范 |  |  |
| 是否包含对应的单元测试 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 变量/常量 （10分） | 是 | 否 |
| 相关系列变量是否统一集中声明，自定义数据类型变量是否进行了必要而充分的描述 |  |  |
| 变量是否按相关规范进行了命名（有意义的命名、数据类型的区分等） |  |  |
| 变量是否按其命名含义承担了单一的使用目的，必要时是否清晰定义了额外变量 |  |  |
| 变量引用是否正确关闭 |  |  |
| 常量是否使用getter/setter方法访问 |  |  |
| 是否应用良好定义的枚举类型替代了简单的字符标识或布尔变量使用 |  |  |
| 是否优先使用Java标准库提供的容器（ArrayList、HashSet、HashMap、LinkList）来处理顺序结构、集合、关联数组合堆栈队列 |  |  |
| 对于public尽量谨慎使用 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 算法 （10分） | 是 | 否 |
| 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离 |  |  |
| 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 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 控制 （10分） | 是 | 否 |
| 代码执行路径是否清晰 |  |  |
| 语句是否有缺省分支 |  |  |
| 控制逻辑复杂度是否合理，是否进行了必要而充分的注释 |  |  |
| 每个循环体是否仅执行了单一而明确的功能 |  |  |
| 与常数比较需要将常数放在比较表达式的前面 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 布局和结构 （5分） | 是 | 否 |
| 代码是否良好格式化并能体现其逻辑结构 |  |  |
| 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 |  |  |
| 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 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 设计 （15分） | 是 | 否 |
| 程序是否可读、可扩展并健壮（可扩展性、复用性以及组件化） |  |  |
| 具体实现细节是否已尽可能的“隐藏” |  |  |
| 是否优先使用接口而不是抽象类或具体类 |  |  |
| 方法参数是否在5个以内 |  |  |
| 对于不需要子类来重载的类尽量使用final |  |  |
| 对象作用域是否过大 |  |  |
| 是否合理使用设计模式 |  |  |
| 历史遗留的解决以及技术突破 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 模块性 （10分） | 是 | 否 |
| 模块(方法,类)之间是否具有低偶合性 |  |  |
| 每个模块(方法,类)自身是否具有高聚合性 |  |  |
| 类库的使用是否合理 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| 效率和防御性 （10分） | 是 | 否 |
| 在循环内是否出现远程调用 |  |  |
| 每个业务动作远程调用次数是否过大 |  |  |
| 远程调用数据传输是否有不必要的冗余数据 |  |  |
| 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 |  |  |
| 对于服务调用是否有超时设计或异常处理 |  |  |
| **得分：** |  |  |

|  |  |  |
| --- | --- | --- |
| SQL （15分） | 是 | 否 |
| Sql语句小写 |  |  |
| 引用字符使用单引号 |  |  |
| 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 |  |  |
| 避免隐含的类型转换（不同数据类型字段相加） |  |  |
| 子查询前后必须加上括号 |  |  |
| 禁止使用视图 |  |  |
| 禁止使用not in，建议使用not exist |  |  |
| 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 |  |  |
| Where子句过滤条件，索引列或过滤记录最多的条件应该放在前面 |  |  |
| Case when语句中只能出现=、>=、<=以及is null运算符 |  |  |
| 左连接写法必须带”outer”关键字 |  |  |
| Sql中函数是否在指定范围内 |  |  |
| Join与on必须严格匹配 |  |  |
| **得分：** |  |  |

**问题清单**

|  |  |  |  |
| --- | --- | --- | --- |
| **问题描述** | **预计修改日期** | **修改日期** | **检查人签字** |
|  |  |  |  |