**代码检查要点表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目名称** | | | 国泰安梦想基地平台软件V1.0 | **检查日期** | | 2018.6.13 | |
| **文件编号、名称** | | | 迭代一源码 | **填写人** | | 周攀 | |
| **编号** | **问题** | | | **是** | **否** | **不适用** | **Bug数量** |
| **VC：变量、属性、常量声明缺陷** | | | | | | | |
| 1 | 变量和常量的命名是否与约定一致？ | | | ☑ | □ | □ | 0 |
| 2 | 是否存在容易混淆的相似变量与属性名？ | | | □ | ☑ | □ | 1 |
| 3 | 变量和属性是否书写正确？ | | | ☑ | □ | □ | 0 |
| 4 | 变量和属性是否被正确初始化？ | | | □ | ☑ | □ | 1 |
| 5 | 非局部变量是否能用局部变量替换？ | | | ☑ | □ | □ | 0 |
| 6 | 所有for循环的控制变量是否都在循环顶部被声明？ | | | ☑ | □ | □ | 0 |
| 7 | 是否有应该命名的常量的文字常量？ | | | ☑ | □ | □ | 0 |
| 8 | 变量和属性是否可以用常量替换？ | | | ☑ | □ | □ | 0 |
| 9 | 属性是否可以用本地变量？ | | | ☑ | □ | □ | 0 |
| 10 | 所有的属性是否都有正确的访问限制符？ | | | ☑ | □ | □ | 0 |
| 11 | 是否有静态属性应该是非静态？ | | | ☑ | □ | □ | 0 |
| **FD：方法定义缺陷** | | | | | | | |
| 1 | 方法名的描述方法是否与命名约定一致？ | | | ☑ | □ | □ | 0 |
| 2 | 每个方法的参数值在使用前是否都做了检查？ | | | ☑ | □ | □ | 0 |
| 3 | 对于每一个方法，它是否都返回了正确的值？ | | | ☑ | □ | □ | 0 |
| 4 | 每种方法是否都有正确的访问限制符？（private,protected,internal,public） | | | ☑ | □ | □ | 0 |
| 5 | 静态方法是否应该为非静态？ | | | □ | ☑ | □ | 0 |
| **CD：类定义缺陷** | | | | | | | |
| 1 | 每一个类的描述是否与命名一致？ | | | ☑ | □ | □ | 0 |
| 2 | 是否书写正确？ | | | ☑ | □ | □ | 0 |
| 3 | 在子类中是否有应该放到父类的通用成员？ | | | ☑ | □ | □ | 0 |
| 4 | 类的继承层次是否能被简化？ | | | ☑ | □ | □ | 0 |
| **DR：数据引用缺陷** | | | | | | | |
| 1 | 对于每个数组引用，下标值是否在定义的范围内？ | | | □ | □ | ☑ | 0 |
| 2 | 对于对象和数组引用，是否确定其值应为非空？ | | | □ | □ | ☑ | 0 |
| **CN：计算/数值缺陷** | | | | | | | |
| 1 | 是否存在不同类型数据之间的混合计算？ | | | □ | ☑ | □ | 0 |
| 2 | 在计算中是否存在上溢或下溢的可能？ | | | □ | ☑ | □ | 0 |
| 3 | 关于数值计算的顺序和优先级的假设是否正确？ | | | ☑ | □ | □ | 0 |
| 4 | 是否用了括号来避免模糊不清？ | | | ☑ | □ | □ | 0 |
| 5 | 文本中是否有拼写和语法上的错误？ | | | □ | ☑ | □ | 0 |
| 6 | 所有的I/O异常处理是否合理？ | | | □ | □ | ☑ | 0 |
| **CR：比较/关系缺陷** | | | | | | | |
| 1 | 对每一个bool测试，正确条件是否被检查？ | | | ☑ | □ | □ | 0 |
| 2 | 比较操作符是否正确？ | | | ☑ | □ | □ | 0 |
| 3 | Bool表达式是否通过内部否定操作进行了简化？ | | | ☑ | □ | □ | 0 |
| 4 | 每个bool表达式是否都正确？ | | | ☑ | □ | □ | 0 |
| 5 | 比较操作是否存在不引人注意的副作用？ | | | □ | ☑ | □ | 0 |
| 6 | “&&”是否被替换成了”&”，”||”是否被替换成了”|”？ | | | □ | ☑ | □ | 0 |
| **CF：流程控制缺陷** | | | | | | | |
| 1 | 对于每一个循环，是否选择了最佳的循环结构？ | | | □ | ☑ | □ | 1 |
| 2 | 所有的循环是否都能结束？ | | | ☑ | □ | □ | 0 |
| 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？ | | | ☑ | □ | □ | 0 |
| 4 | Switch声明是否都有default条件？ | | | ☑ | □ | □ | 0 |
| 5 | 是否所有的switch-case-break对应关系都已更正并加上批注？ | | | □ | □ | ☑ | 0 |
| 6 | 是否named break叙述都跳到正确的地方？ | | | ☑ | □ | □ | 0 |
| 7 | 循环和分支的嵌套是否过深？是否正确？ | | | ☑ | □ | □ | 1 |
| 8 | 是否有if嵌套可以转换成switch嵌套？ | | | ☑ | □ | □ | 1 |
| 9 | 空控制叙述是否都正确，并加上括号及批注？ | | | ☑ | □ | □ | 0 |
| 10 | 所有的异常是否都得到了正确的处理？ | | | ☑ | □ | □ | 0 |
| 11 | 每一个方法是否都结束？ | | | ☑ | □ | □ | 0 |
| **IO：输入输出缺陷** | | | | | | | |
| 1 | 文档在被使用之前是否都被打开？ | | | □ | □ | ☑ | 0 |
| 2 | 输入对象的属性是否与使用的文件一致？ | | | □ | □ | ☑ | 0 |
| 3 | 文件在被使用之后是否都被关闭？ | | | □ | □ | ☑ | 0 |
| 4 | 独占性的文档检查是否打开前已经被使用？ | | | □ | □ | ☑ | 0 |
| **MI：模块间接口缺陷** | | | | | | | |
| 1 | 方法调用的参数的数量、顺序、类型和值是否与该声明一致？ | | | □ | □ | ☑ | 0 |
| 2 | 度量单位是否一致？ | | | □ | □ | ☑ | 0 |
| 3 | 如果对象或数组被传递，他们是否改变？是否被调用方法正确改变？ | | | □ | □ | ☑ | 0 |
| **CM：注释缺陷** | | | | | | | |
| 1 | 每一个方法、类和文件是否都有适当的头注释？ | | | ☑ | □ | □ | 0 |
| 2 | 每一个属性、变量和常量的声明是否都有注释？ | | | ☑ | □ | □ | 0 |
| 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？ | | | ☑ | □ | □ | 0 |
| 4 | 方法和类的头注释是否和它们的功能保持一致？ | | | □ | ☑ | □ | 1 |
| 5 | 注释和代码是否保持一致？ | | | □ | ☑ | □ | 1 |
| 6 | 注释对于理解代码是否有帮助？ | | | ☑ | □ | □ | 0 |
| 7 | 代码中的注释是否充分？ | | | ☑ | □ | □ | 0 |
| 8 | 代码中的注释是否过多？ | | | □ | ☑ | □ | 0 |
| **LP：布局和封包缺陷** | | | | | | | |
| 1 | 代码布局格式和缩排标准是否一致？ | | | □ | □ | ☑ | 0 |
| 2 | 对于每一个方法，它的代码量是否都不超过100行？ | | | □ | □ | ☑ | 0 |
| 3 | 对于每一个编译模块，它的代码来那个是否都不超过600行？ | | | □ | □ | ☑ | 0 |
| **MO：模块性缺陷** | | | | | | | |
| 1 | 模块（方法、类）之间是否具有低耦合性？ | | | ☑ | □ | □ | 0 |
| 2 | 每个模块（方法、类）自身是否具有高聚合性？ | | | □ | ☑ | □ | 1 |
| 3 | 是否存在重复的代码，它的功能是否可以通过调用其他方法实现或进行抽象？ | | | □ | ☑ | □ | 0 |
| 4 | 类库的使用是否适时适量？ | | | ☑ | □ | □ | 0 |
| **SU：存储器使用缺陷** | | | | | | | |
| 1 | 数据是否足够大？ | | | □ | □ | ☑ | 0 |
| 2 | 存储的方式是否合理？是否节省空间或方便快速寻找？ | | | □ | □ | ☑ | 0 |
| 3 | 数组和对象不再使用之后，他们的引用是否被赋为空值？ | | | □ | □ | ☑ | 0 |
| **PE：性能缺陷** | | | | | | | |
| 1 | 是否有更好的数据结构和算法可以采用？ | | | □ | □ | ☑ | 0 |
| 2 | 测试安排是否合理，使易于通过的且代价低廉的测试优先与代价较高且通过频率较低的测试？ | | | ☑ | □ | □ | 0 |
| 3 | 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？ | | | □ | ☑ | □ | 0 |
| 4 | 每一个计算并保存了结果是否都被应用？ | | | ☑ | □ | □ | 0 |
| 5 | 计算是否能被移到循环之外？ | | | □ | ☑ | □ | 0 |
| 6 | 在循环内是否有不需要的测试？ | | | □ | ☑ | □ | 0 |
| 7 | 短循环是否可以取消？ | | | □ | ☑ | □ | 0 |
| 8 | 对同一个数据进行操作的两个循环是否可以合并成一个？ | | | ☑ | □ | □ | 1 |
| **其他** | | | | | | | |
| 1 | 拆卸步骤控制类太长，可以抽象成多个类 | | | □ | □ | □ |  |
| 2 | 通过性能跟踪，发现拆卸场景由内存泄漏情况 | | | □ | □ | □ |  |
| 3 |  | | | □ | □ | □ |  |
| 4 |  | | | □ | □ | □ |  |
| 5 |  | | | □ | □ | □ |  |
| 6 |  | | | □ | □ | □ |  |
| 7 |  | | | □ | □ | □ |  |
| 8 |  | | | □ | □ | □ |  |
| 9 |  | | | □ | □ | □ |  |
| 10 |  | | | □ | □ | □ |  |
| **结论：**  ☑通过 □有条件通过 □不通过 | | | | | | | |
| **说**  **明** | | 无 | | | | | |