

代码复审检查表

1. 概要部分

(1) 代码能符合需求和规格说明么？

答：能符合需求与规格

(2) 代码设计是否有周全的考虑？

答：不太周全，修改了很多次

(3) 代码可读性如何？

答：简单易懂

(4) 代码容易维护么？

答：不太容易，主要是检查重复的逻辑太多，需要后期完善

(5) 代码的每一行都执行并检查过了吗？

答：是的，检查过

2. 设计规范部分

(1) 设计是否遵从已知的设计模式或项目中常用的模式？

答：没有，目前接触的设计模式不多

(2) 有没有硬编码或字符串/数字等存在？

答：很大部分，基本上都是以字符串和数字进行操作

(3) 代码有没有依赖于某一平台，是否会影响将来的移植（如Win32到Win64）

答：未知。没有尝试过

(5) 有没有无用的代码可以清除？（很多人想保留尽可能多的代码，因为以后可能会用上，这样导致程序文件中有很多注释掉的代码，这些代码都可以删除，因为源代码控制已经保存了原来的老代码。）

答：有，已清除

3. 代码规范部分

(1) 修改的部分符合代码标准和风格么（详细条文略）？

答：符合代码标准

4. 具体代码部分

(1) 有没有对错误进行处理？对于调用的外部函数，是否检查了返回值或处理了异常？

答：查阅资料并讨论以后处理完成，并没有检查异常，这是本次代码的一个缺陷

(2) 参数传递有无错误，字符串的长度是字节的长度还是字符（可能是单/双字节）的长度，是以0开始计数还是以1开始计数？

答：有错误已改正，字符串的长度是字节的长度，以0开始计数

(3) 边界条件是如何处理的？Switch语句的Default是如何处理的？循环有没有可能出现死循环？

答：通过讨论并且分类得到的，或者问同学解决的，循环不会出现死循环

(4) 有没有使用断言 (Assert) 来保证我们认为不变的条件真的满足?

答: 没有使用

(5) 对资源的利用, 是在哪里申请, 在哪里释放的? 有没有可能导致资源泄露 (内存、文件、各种GUI资源、数据库访问的连接, 等等)? 有没有可能优化?

答: 在网上找到的, 不会导致资源泄漏

(6) 数据结构中是否有无用的元素?

答: 检查过后没有

5. 效能

(1) 代码的效能 (Performance) 如何? 最坏的情况是怎样的?

答: 代码正确, 程序运行正常。

(2) 代码中, 特别是循环中是否有明显可优化的部分 (C++中反复创建类, C#中 string 的操作是否能用StringBuilder 来优化)?

答: 有可优化部分, 但是代码正确就没有修改

6. 可读性

代码可读性如何? 有没有足够的注释?

答: 代码不是很复杂, 没有任何注释

7. 可测试性

代码是否需要更新或创建新的单元测试?

可能会需要, 本次的测试并不全面