# 代码质量分析报告

## 代码质量分析方法

说明对代码质量进行分析的方法和手段。

## 代码质量分析工具介绍

介绍代码质量分析的支撑软件工具，以及如何应用工具来分析代码质量

## 人工分析质量报告

通过阅读代码，分析高质量代码有何特点，体现在哪些代码之中。

### 高质量的编码规范

介绍相关程序设计语言的编码规范，代码在多大程度上遵循了该规范

### 高质量的代码设计

高内聚，低耦合：程序模块的可重用性、移植性很强

### 高质量编程的方法和高水平的编程技能

1、少使用类的继承，多用接口隐藏实现的细节。 Java面向对象编程引入接口除了支持多态外， 隐藏实现细节也是其中一个目的。

2、模块的功能化分尽可能的单一，道理也很简单，功能单一的模块供其它模块调用的机会就少。（其实这是高内聚的一种说法，高内聚低耦合一般同时出现）。

3、遵循一个定义只在一个地方出现。

4、少使用全局变量。

5、类属性和方法的声明少用public，多用private关键字。

6、多用设计模式，比如采用MVC的设计模式就可以降低界面与业务逻辑的耦合度。

7、尽量不用“硬编码”的方式写程序，同时也尽量避免直接用SQL语句操作数据库。

8、避免直接操作或调用其它模块或类（内容耦合）；如果模块间必须存在耦合，原则上尽量使用数据耦合，少用控制耦合，限制公共耦合的范围，避免使用内容耦合。

9、模块只对外暴露最小限度的接口，形成最低的依赖关系。

10、只要对外接口不变，模块内部的修改，就不得影响其他模块。

11、删除一个模块，应当只影响有依赖关系的其他模块，而不应该影响其他无关部分。

### 存在的质量问题

通过阅读分析代码中尚存在哪些质量问题和瑕疵。

## 自动分析质量报告

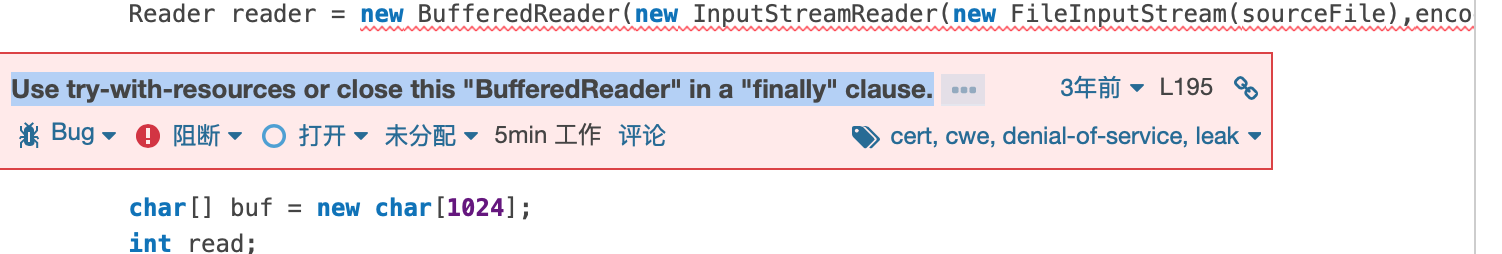
使用Sonarqube进行分析后，utils包中发现的bugs如下：

严重程度：Blocker

1. Use try-with-resources or close this "OutputStreamWriter" in a "finally" clause：应该使用try-with-resources创建资源或将此输出流在finally语句块中关闭。



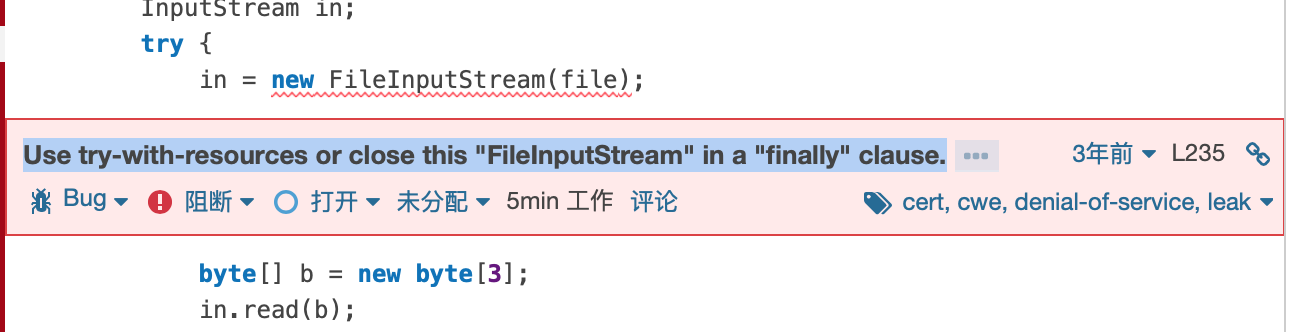
1. Use try-with-resources or close this "BufferedReader" in a "finally" clause：应该使用try-with-resources创建资源或将此输入流在finally语句块中关闭。



1. Use try-with-resources or close this "BufferedInputStream" in a "finally" clause：应该使用try-with-resources创建资源或将此输入流在finally语句块中关闭。



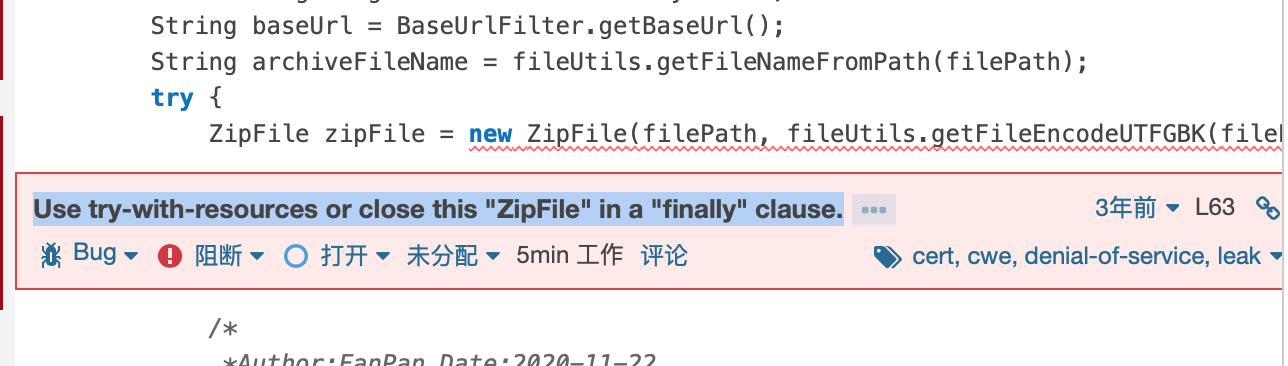
1. Use try-with-resources or close this "FileInputStream" in a "finally" clause：应该使用try-with-resources创建资源或将此输入流在finally语句块中关闭。



1. Use try-with-resources or close this "FileOutputStream" in a "finally" clause.：应该使用try-with-resources创建资源或将此输出流在finally语句块中关闭。



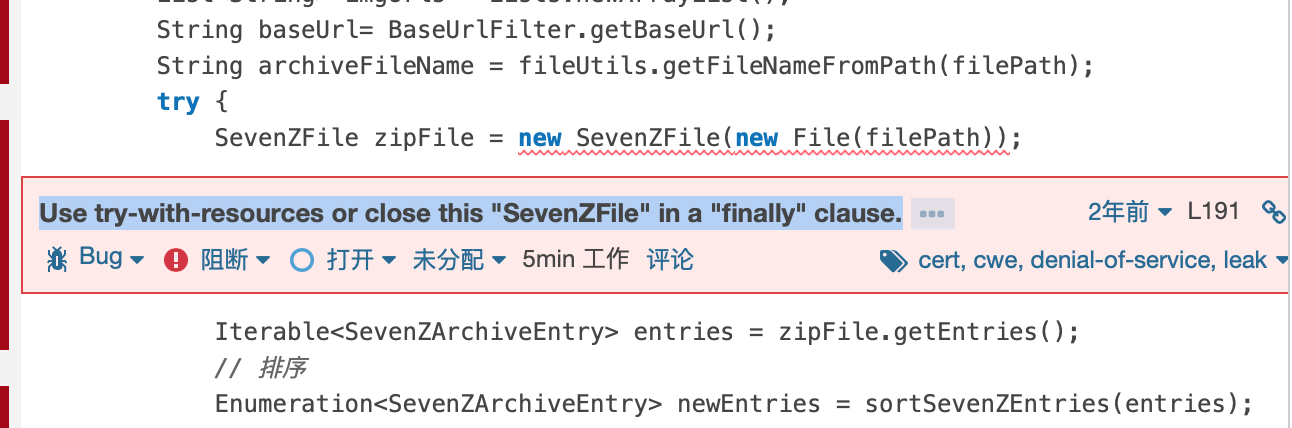
1. Use try-with-resources or close this "ZipFile" in a "finally" clause：应该使用try-with-resources创建资源或将此ZipFile在finally语句块中关闭。



1. Use try-with-resources or close this "Archive" in a "finally" clause：应该使用try-with-resources创建资源或将此Archive在finally语句块中关闭。



1. Use try-with-resources or close this "SevenZFile" in a "finally" clause. 应该使用try-with-resources创建资源或将此"SevenZFile"在finally语句块中关闭。



1. Use try-with-resources or close this "SevenZFile" in a "finally" clause. 应该使用try-with-resources创建资源或将此"SevenZFile"在finally语句块中关闭。



1. Use try-with-resources or close this "FileOutputStream" in a "finally" clause. 应该使用try-with-resources创建资源或将此输入流在finally语句块中关闭。



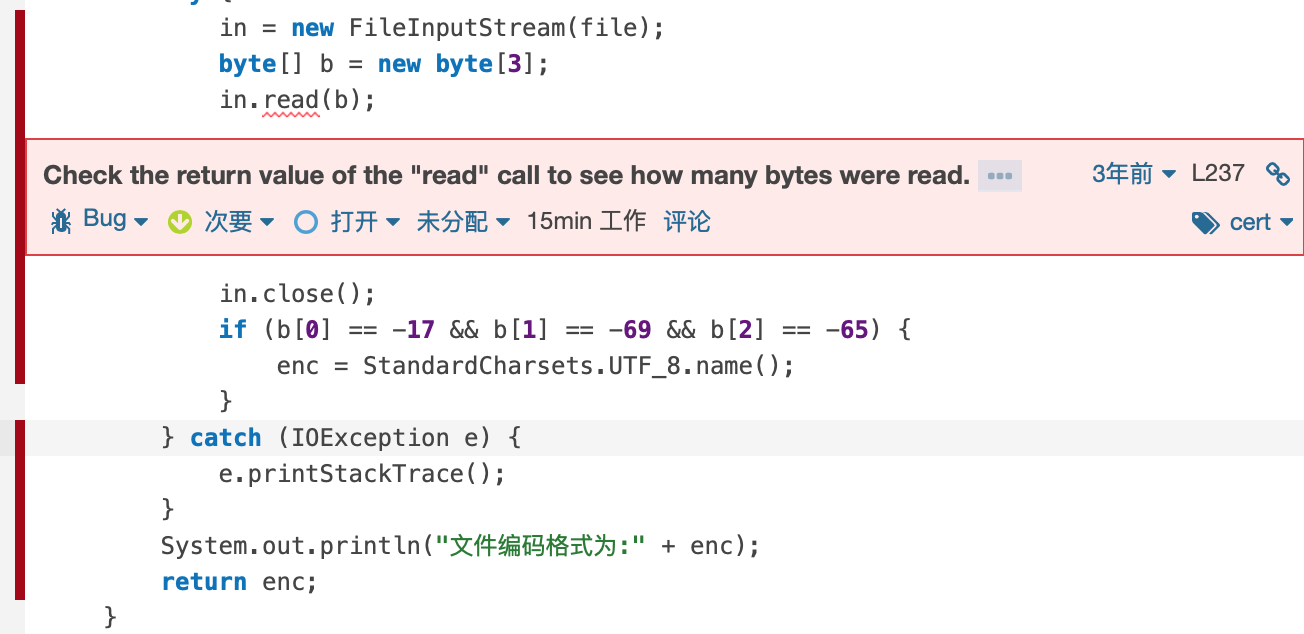
以上bugs皆属于同一种问题。

实现可关闭接口或其超接口、AutoCloseable的连接、流、文件和其他类在使用后需要关闭。而且，必须在finally块中完成这关闭，否则可能会出现异常使调用无法进行。最好，当类实现AutoCloseable时，应该使用“try-with-resources”模式创建资源，并自动关闭。

如果不能正确地关闭资源，将会导致资源泄漏，首先会使应用程序崩溃，然后可能会使机器瘫痪。

严重程度：Minor

1. Check the return value of the "read" call to see how many bytes were read：检查返回值read来判断多少个字节被读取了



不能假设任何给定的流读取调用都会填充传入到该方法的byte[]。相反，必须检查read方法返回的值，以查看读取了多少字节。如果做不到这一点，就会引入既有害又难以繁殖的bug。

类似地，您不能假设InputStream.skip实际上会跳过请求的字节数，但是必须检查方法返回的值。

当接收一个byte数组的InputStream.read被调用时，该规则会引发一个问题。该规则也适用于InputStream子类。