问题一：添加私有构造方法替换隐藏公共构造方法

阐述：代码规范要求工具类添加私有构造方法。

吐槽：代码规范强制要求工具类添加私有构造函数，是对JavaBean的构造方法理解具有偏颇。

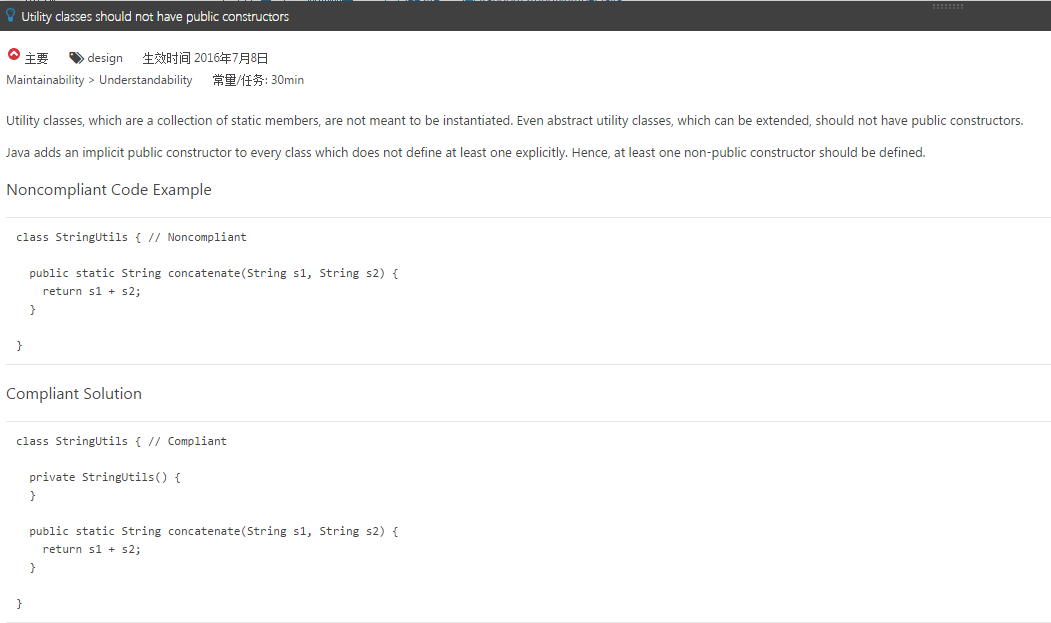
第一、私有构造方法，主要应用于经典的单例设计模式。

第二、如若实体类中不编写显式的构造方法，则Java虚拟机编译时会自动生成默认的构造函数。

第三、工具类访问静态方法，执行方法体，不一定要进行对象实例化，因此，代码中不需要刻意编写构造函数。

最后，SonarQube代码规范属于规范管理系统，由人为配置控制，但配置此代码规范之人对于构造方法和访问权限的理解太过古板，不得不吐槽。

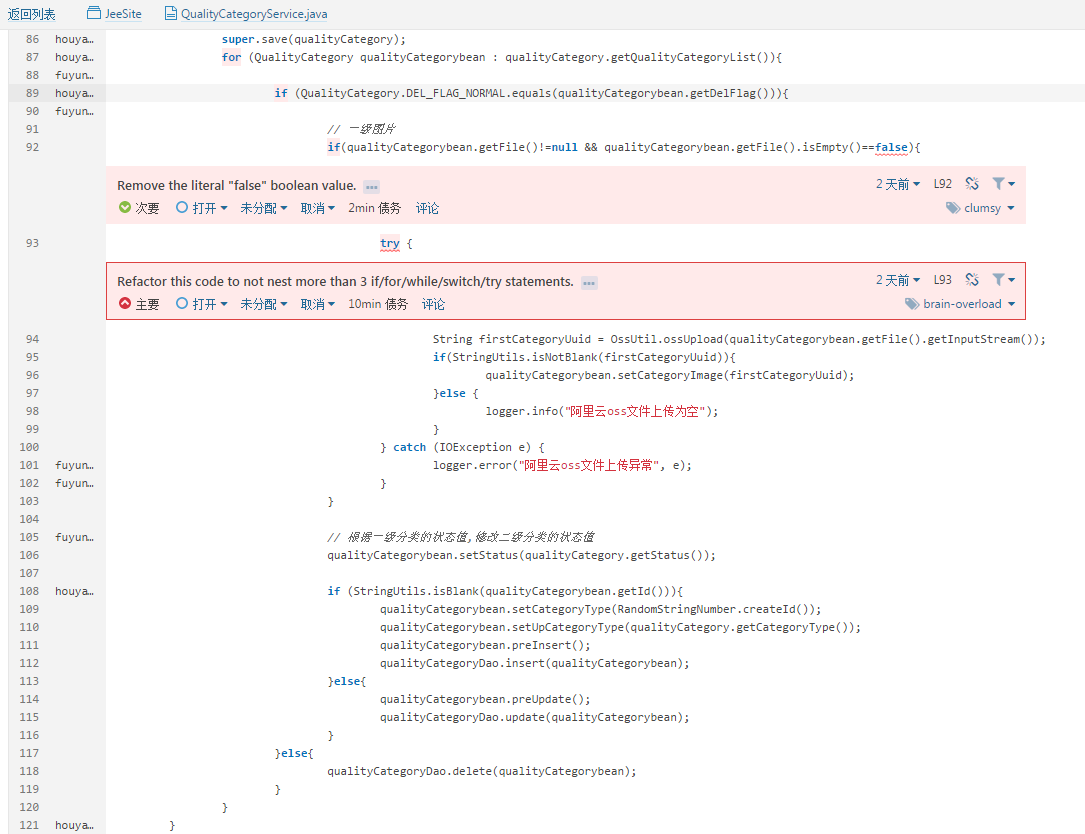


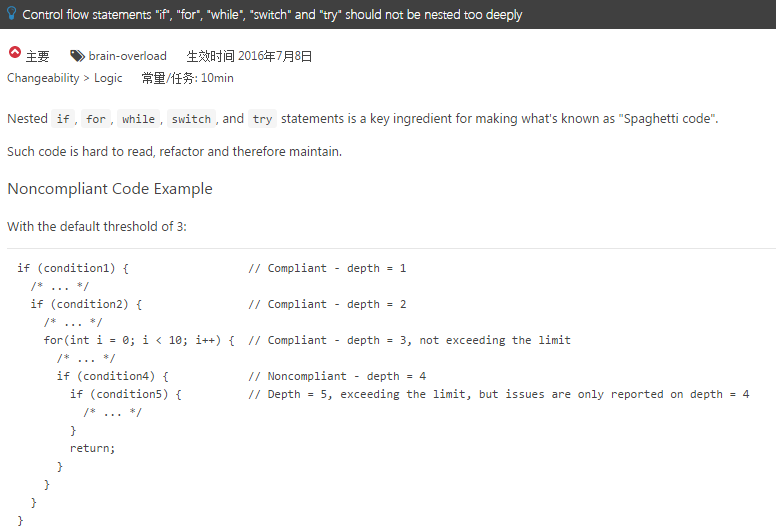


问题二：过于严格控制代码复杂度

阐述：代码规范过于严格控制代码复杂度

吐槽：在实际的代码开发中，不可能完全做到控制在一定层次内的复杂度。同时，规范将try、if与循环判断等同，这就在一定程度上增加了自我认定循环深度，因此，是不正确的。实际开发中，循环深度不应超过三层是合理要求，但是把非循环认定为增加循环复杂度，是不合适的。





问题三：代码复杂度错误限定

阐述：代码规范错误的判断代码复杂度

吐槽：代码规范针对方法名进行复杂度限定，但是实际上产生代码复杂度的位置在于方法体中，从提示规范而言，也应该优化代码，而不是简单改变方法名。

