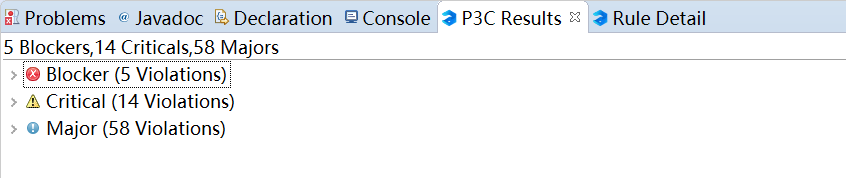
**TimelinePC代码静态分析报告**

**一、运行结果与修正**

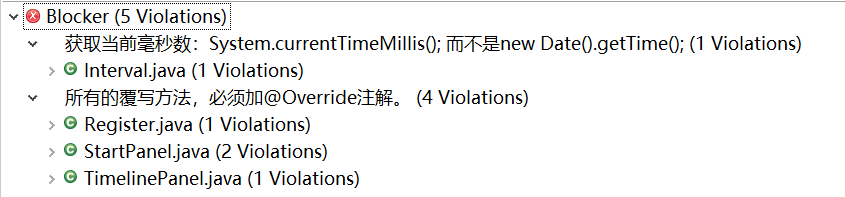
1.代码风格检测器(Code Style Checkers)(应用P3C)

总运行结果如下：



各部分分析：

1.Blocker：

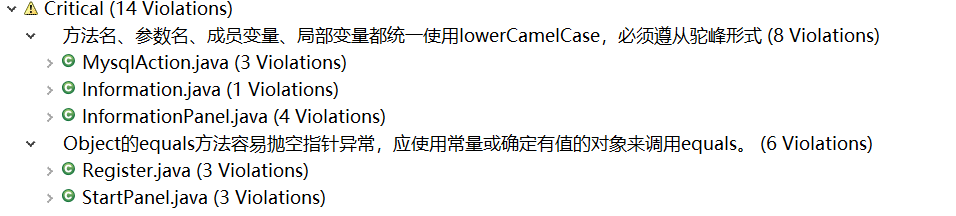


分析：

1).获取当前毫秒数方法不符合规范，应使用System.currentTimeMillis();

2).根据代码规范，所有覆写方法，必须加@override注解。

2.Critical:

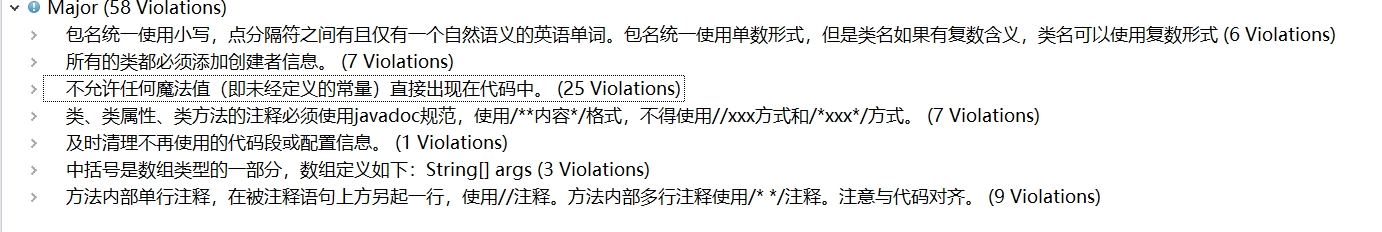


分析：

1).方法名、参数名、成员变量、局部变量都统一使用lowerCamelCase；

2).使用equals方法时容易抛空指针异常，应使用常量或者确定有值的对象来调用equals。

3.Major:



分析：

1).包名统一使用小写;

2).所有类必须添加创建者消息;  
3).不允许魔法值出现;

4).方法的注释必须使用javadoc规范;

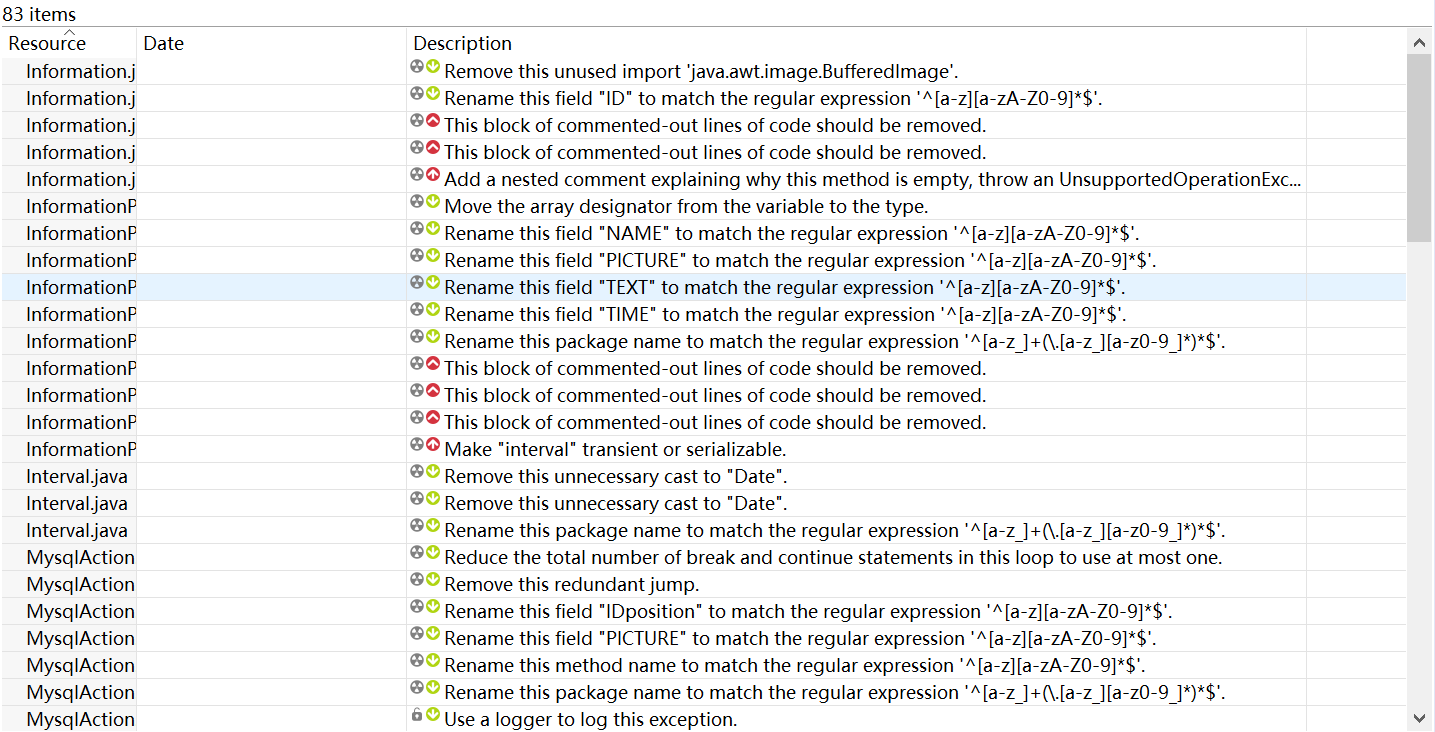
5).即使清理不再使用的代码段及配置信息;

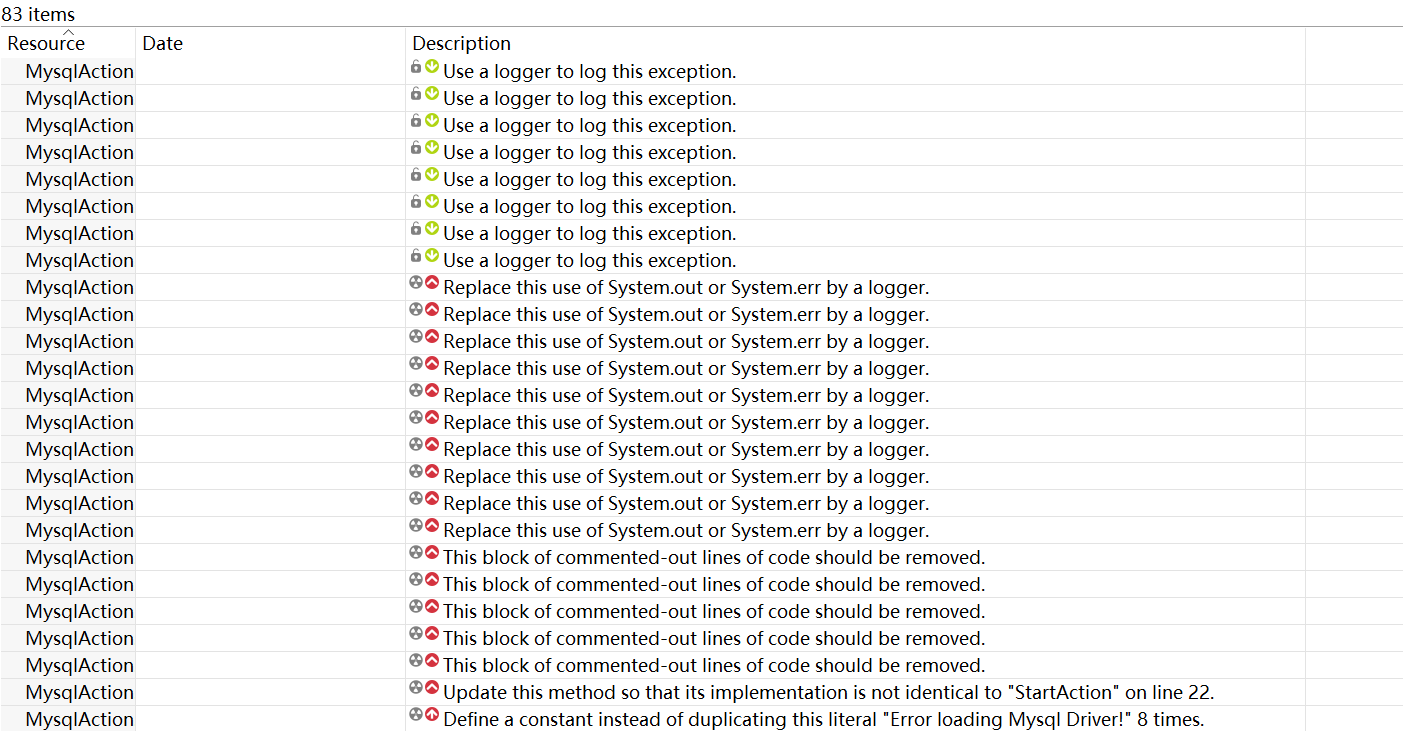
6).数组定义不规范;

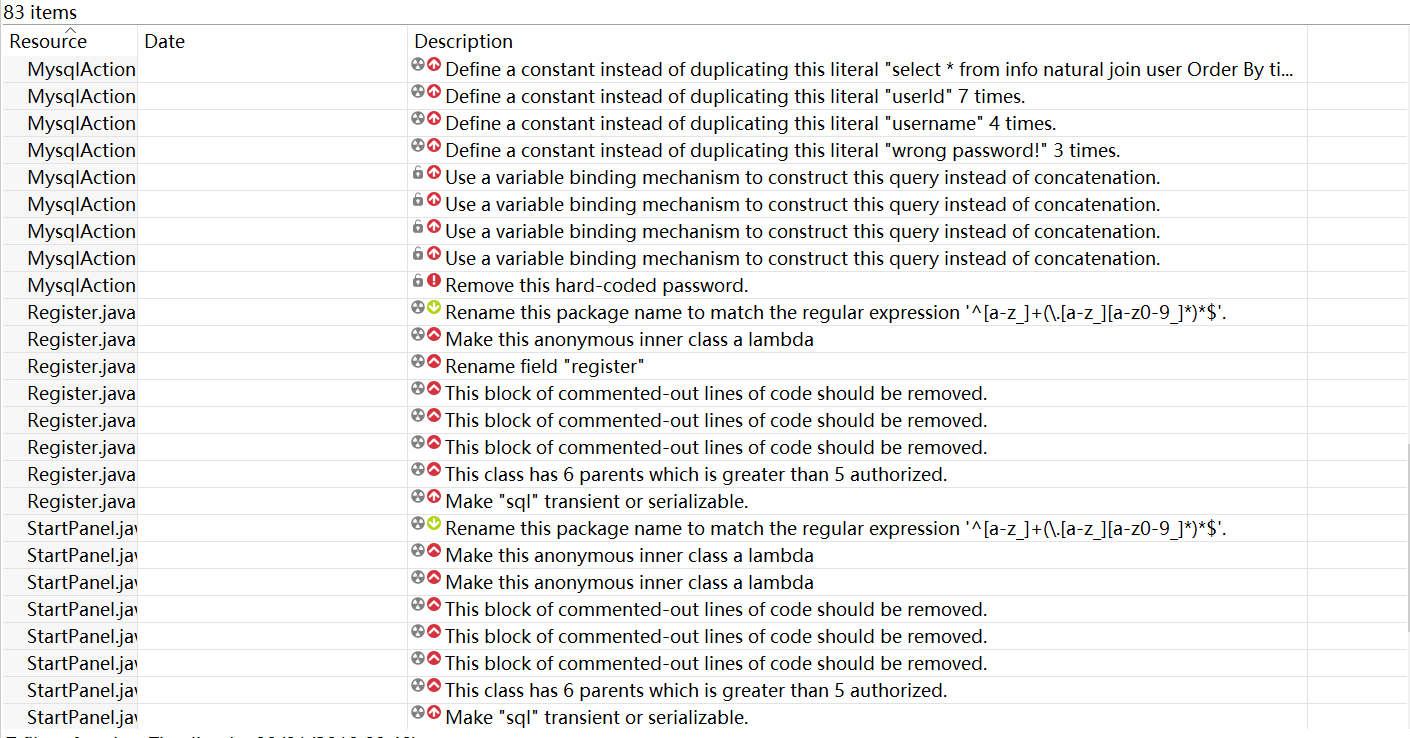
7).注释位置不规范问题

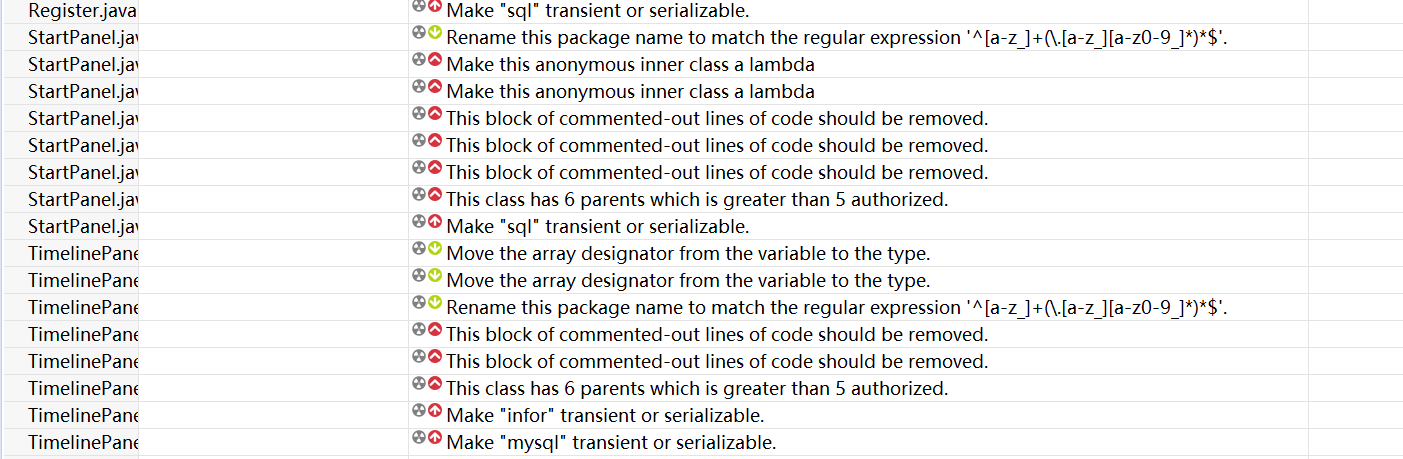
2.缺陷静态检测器(应用Sonarlint)

总运行结果如下：









分析：

经仔细研究运行结果，发现主要有以下问题：

1.命名不规范问题

2.常用String应设为constant

3.注释代码不规范

在后续过程中均已改正。

二、代码静态分析反思

P3C规范中对代码风格提出了较高要求，在修改完毕后，我对P3C规范又有了一些新的掌握和认识。

Sonarlint测试执行后，仍发现较多风格上的缺陷，在下次编写代码时应多加注意。