**华东师范大学软件学院实验报告**

|  |  |  |
| --- | --- | --- |
| **实验课程**：软件测试与验证 | **年级**：2016级 | **实验成绩**： |
| **实验名称：** 码农酒店时钟系统静态分析报告 | **姓名/学号**：费燕虹 10165101109 | |
| **实验日期**：2018.10 |
|  |
|  | | |

**一、实验目的**

1/ 练习静态分析工具的使用

2/ 掌握静态分析的概念和思想

**二、实验内容与实验步骤**

针对码农酒店时钟系统选择静态分析工具进行代码静态分析

**三、实验环境**

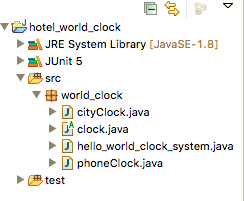
1/ 计算机一台

2/ Eclipse环境

3/代码静态分析工具：**Sonarlint**

**四、实验过程与分析**

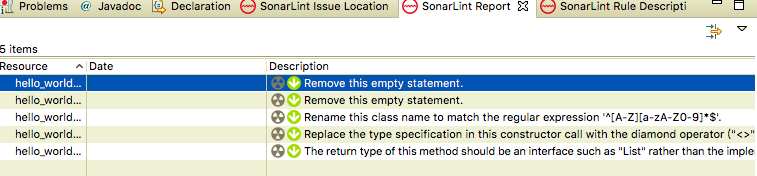
**要实现码农酒店时钟系统，一共需要4个类。分别是酒店时钟系统类，clock类，以及clock的2个子类——phoneClock类和cityClock类。类与类之间的关系是时钟系统类有个属性是cityClock对象组成的list，phoneClock类有个属性是时钟系统对象（因为phoneClock对象的时间一旦设定，时钟系统对象中的所有cityClock对象的时间也随之调整）。**

****

**下面对这四个类的代码进行静态分析**

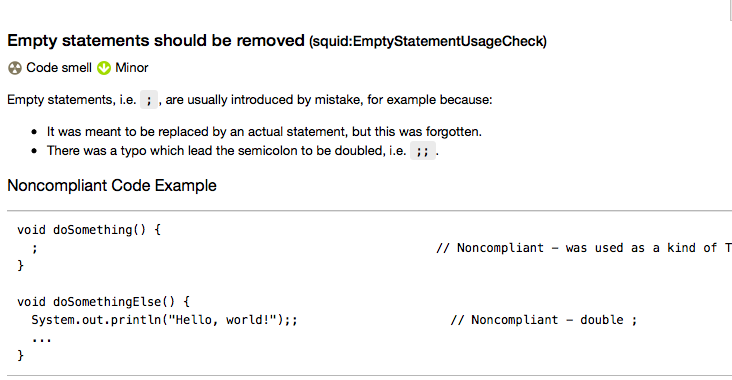
**一 酒店时钟系统class**

静态分析结果如下

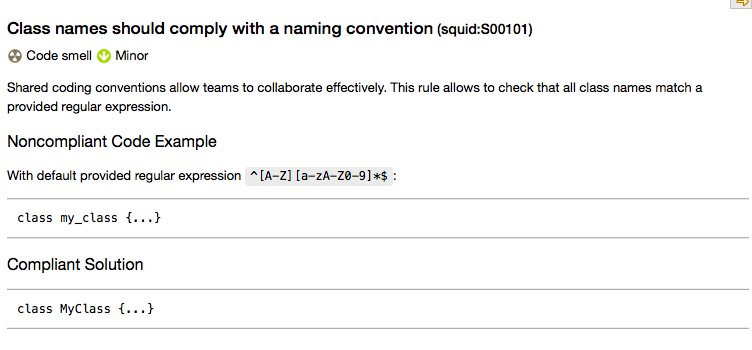


下面是静态分析工具对warning的具体描述，相同warning就不赘述了。

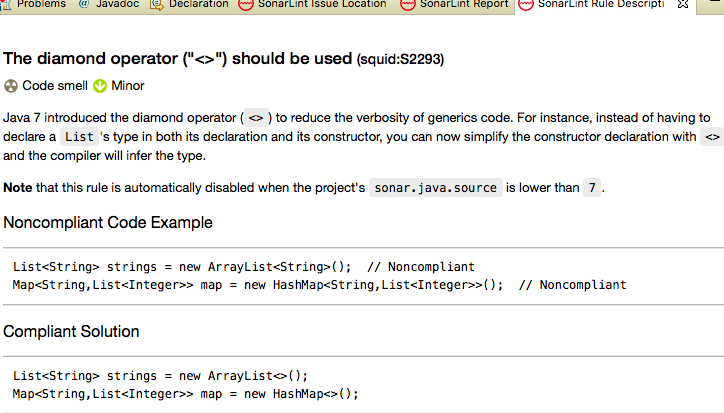
第一种warning是分号的多余。我有c语言基础，而c语言函数的定义和声明要加分号，所以在java 类中定义和声明method时我习惯性地在函数名{函数体}外面加了分号。经过sonarlint的分析，我知道了函数／方法在java中的编码习惯。



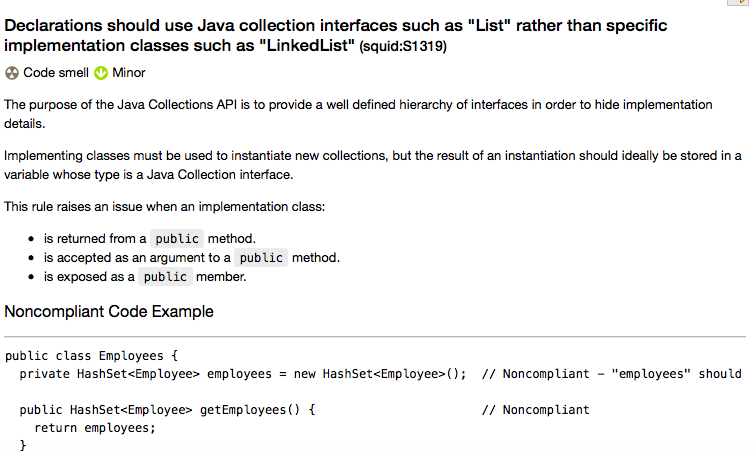
第二种warning是类的命名不符合命名规范。良好的命名方式应使用小写字母／大写字母／数字，尽量避免下划线



第三种warning是模版类（比如ArrayList，若每个元素是int，则ArrayList<int>；若每个元素是String，则ArrayList<String>）在new对象的时候，调用构造函数时，<>里的元素类型可以省略

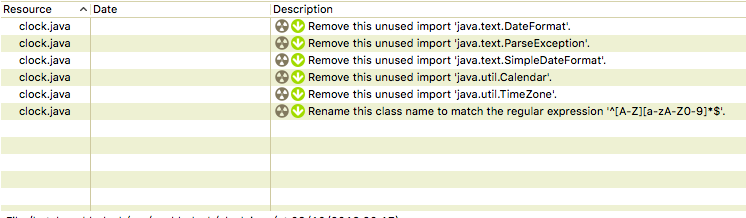


第四种warning是声明变量类型，应该使用java提供的集成接口而不是专门的实现类。比如应该用List而不是ArrayList



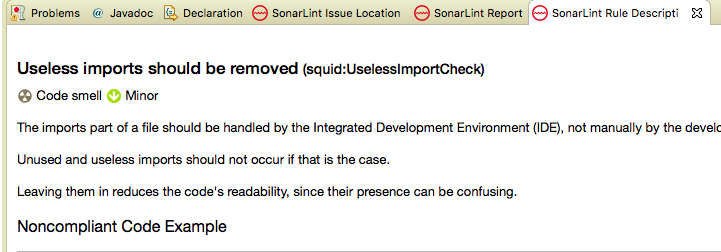
**二 时钟class**

静态分析结果如下

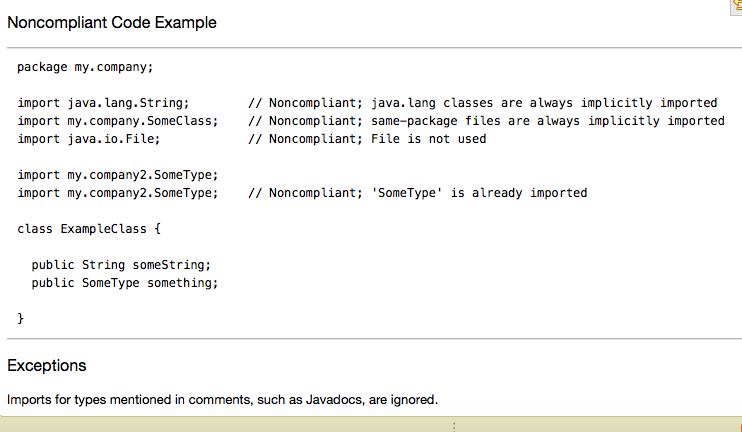


我一开始没有完全弄清题目意思。题目要求utc标准时间是自己设置的，而我以为是让java自己调用库函数获取当前utc标准时间，所以import了许多包，比如java.util.Calendar, java.text.SimpleDateFormat, java.util.TimeZone 等等，后来因为了解到utc标准时间是自己设置的，所以这些包自然没有用到。针对未使用的包，sonarlint分析后提出的建议是这些未使用的包应该被remove

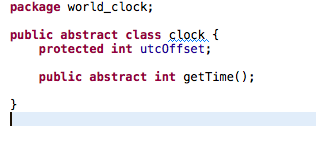
静态分析工具建议：useless imports should be removed



sonarlint非常实用，不仅指出代码不妥原因，还附了出现相同原因的例子便于理解。

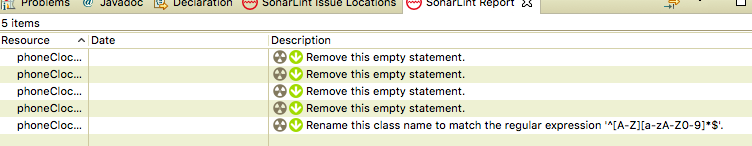


根据sonarlint建议修改后的代码

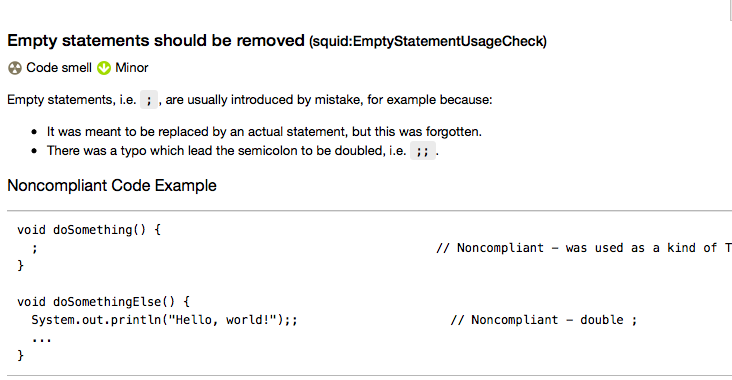


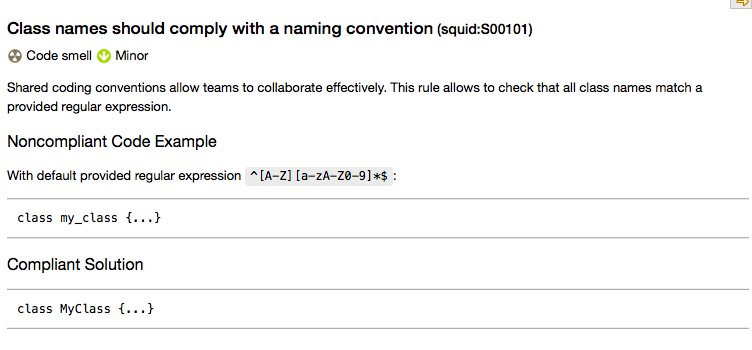
**三 clock类的子类——phoneClock类**

静态分析结果如下



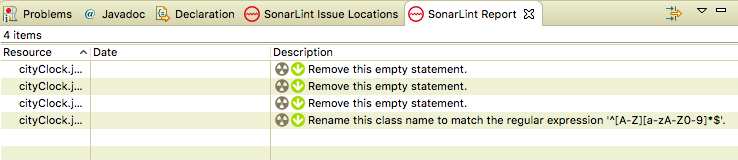
和之前的warning类似，第一个是函数声明｛函数体｝外的分号多余；第二个是没有遵守类名的命名规则：使用小写字母／大写字母／数字，尽量避免下划线



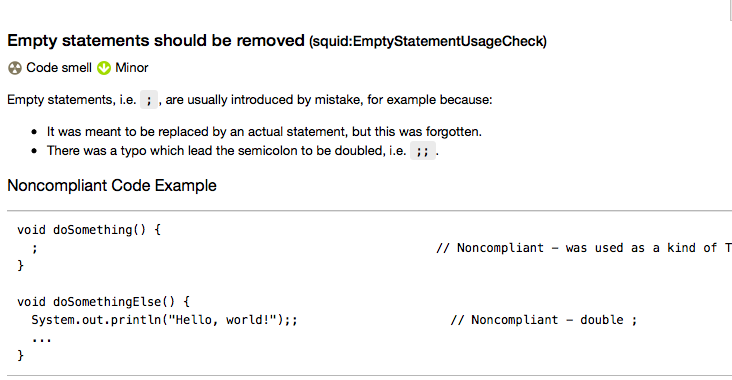


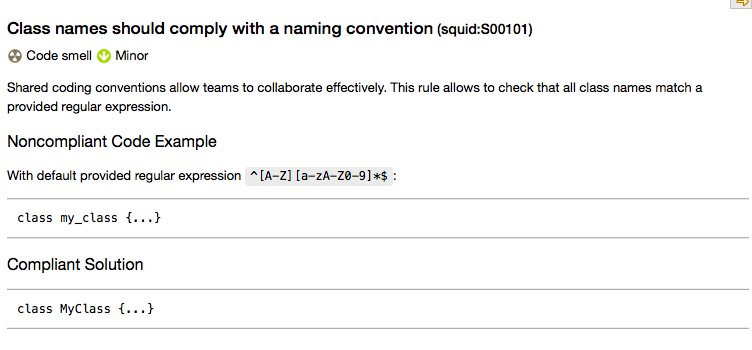
**四 clock子类cityClock类**

静态分析结果如下



同样是函数声明｛函数体｝外的分号多余；以及类的命名不规范：使用小写字母／大写字母／数字，尽量避免下划线





**五、实验结果总结**

1.通过这次代码静态分析实践，我对sonarlint的几个优点感到心动：操作简便／同时sonarlint给出的分析结果清晰易懂，比如不仅指出错误，通过“go to”可知错误代码位置，通过“rule description”可知具体的错误原因，并且还列出来相同错误的例子等等

2个人感觉sonarlint指出的有些错误无伤大雅，略为琐碎。有些建议不一定可取，就像Eclipse的黄色警告不一定每条都要处理。

3 sonarlint的本质还是代码静态分析工具。因为后面的作业是代码自动化测试，通过这二者的比较，我对代码静态分析有了更深入的认识。代码静态分析不需要另外写测试代码，是在代码**运行前**对代码文本的规范检查（还是更偏向语法检查或编码习惯检查）。