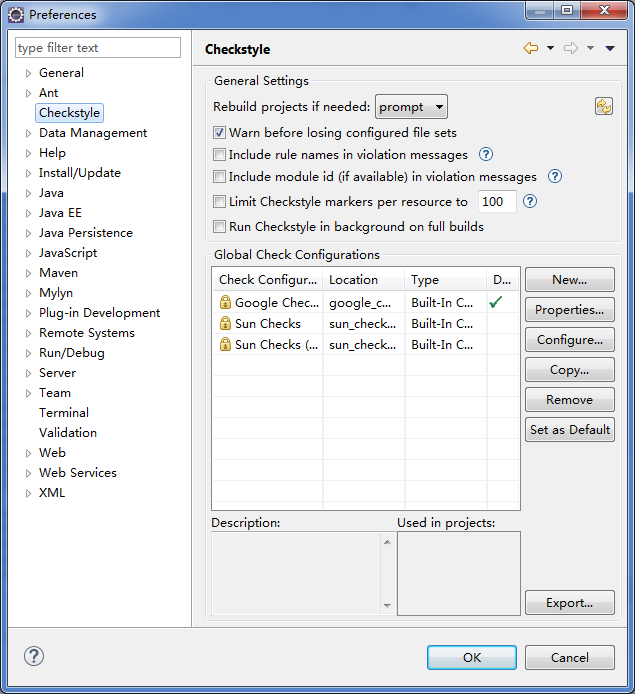
单元测试作业二

HFUT2012214461

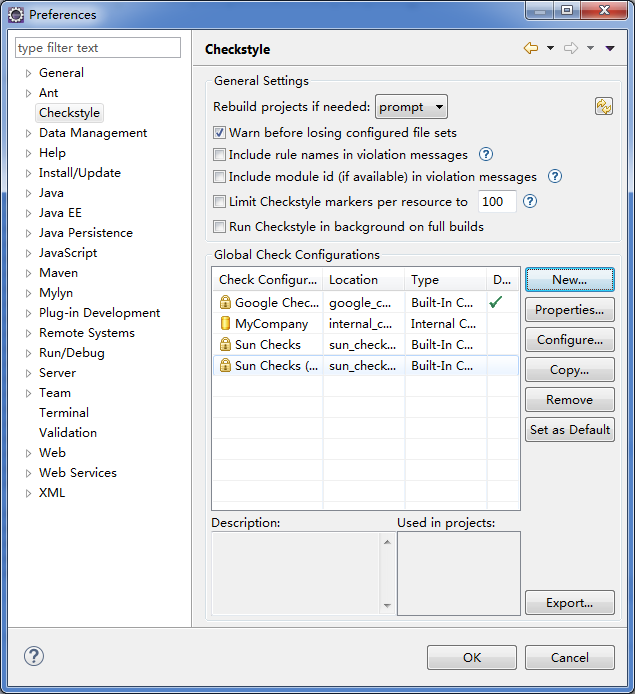
1. Checkstyle

因为网络还是什么原因，我在安装checkstyle时失败，后来就上网查到了离线安装的方法，最后安装成功了。

安装好之后，打开eclipse，在preference选项中就出现了checkstyle选项，如下图所示。

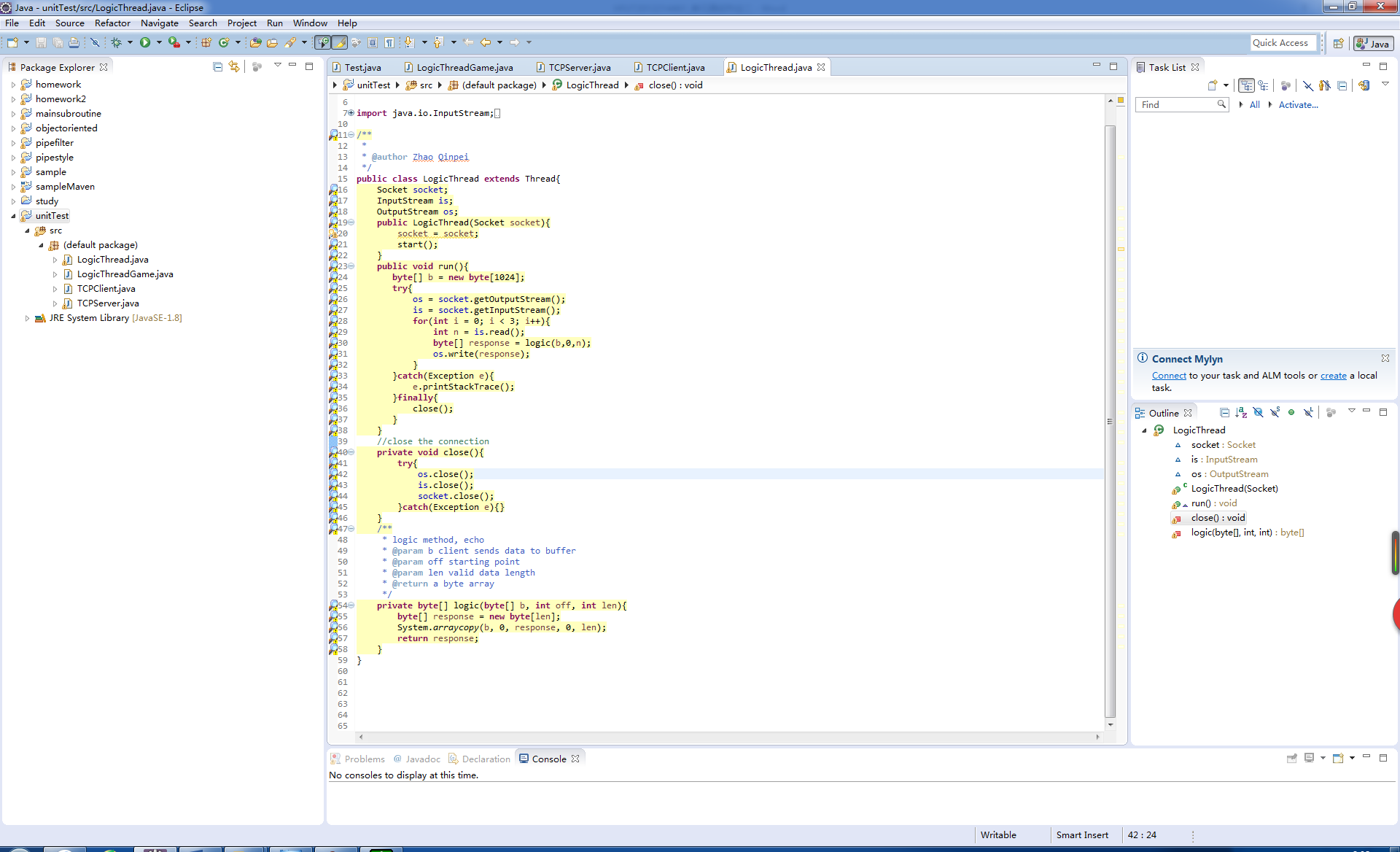


在这里可以对相关的属性进行修改，并可以新建一个自己的style，比如我按老师视频里的一样新建了一个MyCompany



之后我也还是按照Google公司的默认style进行了检查。

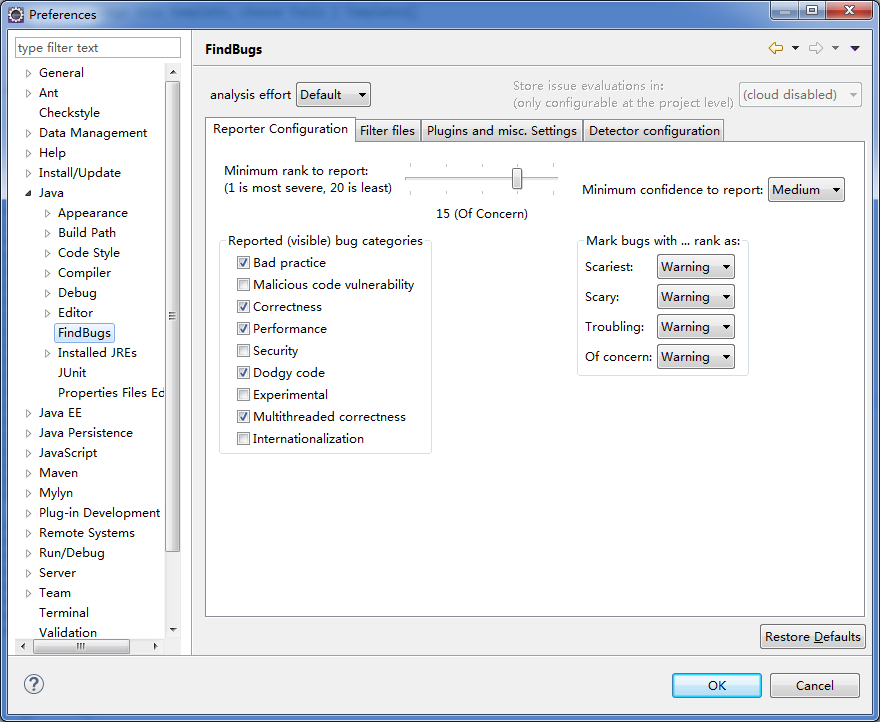
结果如下



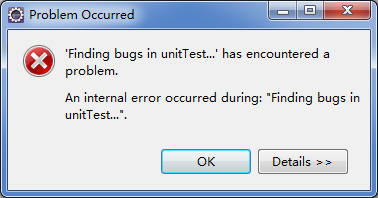
黄色的部分就显示为有问题，不符合规范。如果将鼠标移至放大镜图标上，就会显示出问题的详细描述，以便我们进行修改。

1. Findubugs

Findbugs总是在使用离线安装后，eclipse里出现了相应的选项，如下



但是按照老师视频里所讲的进行运行的时候就出现了问题，如下所示。



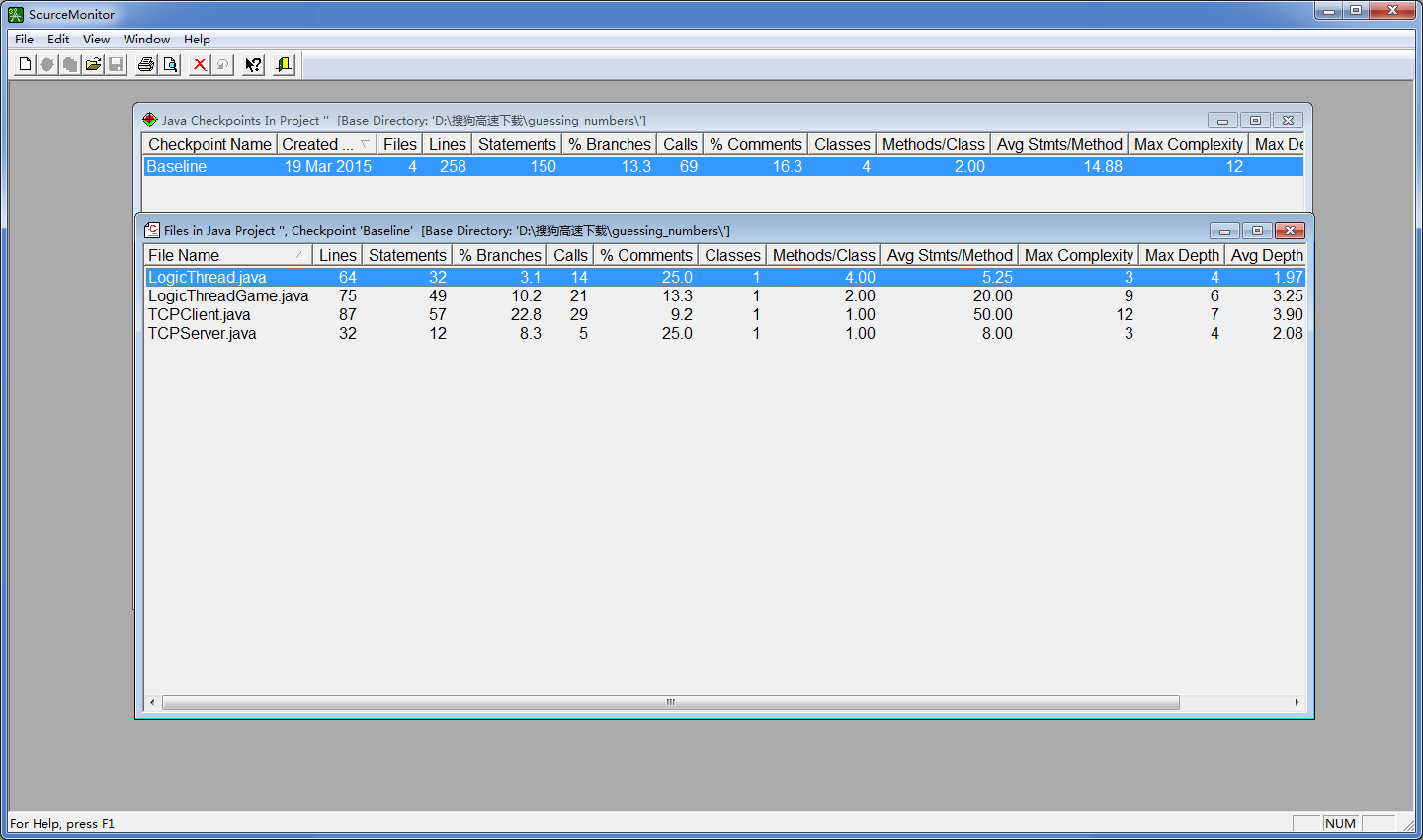
也不知道究竟是哪里错了。我就带着这个问题描述和错误代码去搜索引擎上搜索，也没有找到答案。所以就不能再进行下去了。

1. Sourcemonitor

这个工具安装非常方便，使用安装包直接一路安装就可以了，不需要设置或者什么。

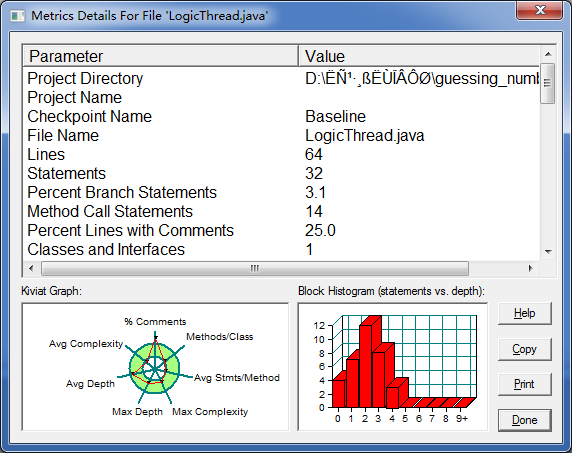
安装好后，我需要新建一个project，然后将老师给的代码添加进去。之后这个工具就会自动进行检测，将相关的参数显示出来，如代码复杂度，单文件代码行数，单函数代码行数等等。

截图如下

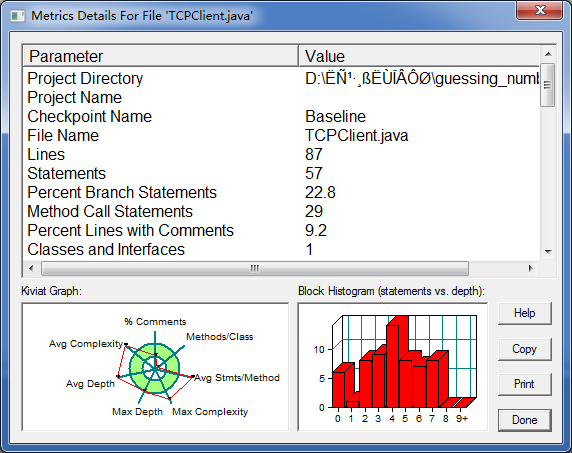


也可以点击某一个具体的文件，会以图表的形式显示出来

如LogicThread.java文件的截图如下



如TCPClient.java文件的截图如下



通过图表可以非常直观地看到平均复杂度，注释比例，平均函数深度等关键的指标信息，方便我们进行修改。

总结：

1. CheckStyle提供了大部分功能都是对于代码规范的检查，是对静态代码是否符合格式规范的检查，自己也可以自定义一个格式要求。
2. Findbugs专注于代码逻辑上是否有风险，它检查类或者 JAR 文件，将[字节码](http://baike.baidu.com/view/560330.htm)与一组缺陷模式进行对比以发现可能的问题。但是因为我这边安装好后，运行出现问题无法解决，所以就不能进行太详细的使用说明。
3. Sourcemonitor是对代码的数值化分析，从总行数(Lines) 语句数目(Statements) 、分支语句比例(Percent Branch Statements) 、注释比例(Percent Lines with Comments) 、函数数目(Functions)、平均每个函数包含的语句数目(Average Statements per Function)、函数圈复杂度(Function Complexity) 、函数深度(Block Depth)几个方面说明代码的情况，以便开发人员了解大概的情况，进行相应的重构和优化。

Github 测试