CheckStyle、FindBugs、SourceMonitor使用心得

我把下载的四个java文件放到了一个名字为test的文件夹下面。

一．CheckStyle使用心得体会：（静态语法检查）

CheckStyle是一款代码格式检查工具，关注代码格式带来的缺陷、风险和问题。

1. 根据设置好的编码规则来检查代码（定制化的）
2. 根据可以触发通知（notification），警告（warning）和错误（error）

③只做检查，不修改代码

1.CheckStyle的安装：

我在安装CheckStyle的过程中一开始是按照老师的方法：首先打开eclipse平台，Eclipse→Help→Install New Software→add，手动输入name和location。然后点击ok按照正常安装软件的步骤进行下去即可安装。但是我这样没有安装成功，原因是我的Eclipse版本过高。

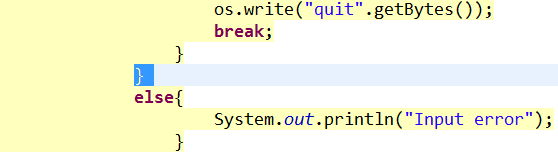
然后我选择了第二种方法：首先在这个网址<http://eclipse-cs.sourceforge.net/#!/>下载插件，下载后，把里面的两个文件夹plugins和features下面的文件分别拷贝到eclipse目录下面对应的plugins和features目录。

2.CheckStyle的使用：

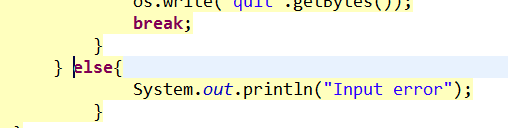
右键点击test项目找到CheckStyle，选择点击Check code with CheckStyle就会进行检查。警告里面的第一行如下所示：



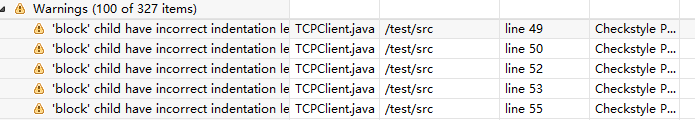
双击一下，会跳转到第17行代码：



根据警告，我把这个“}”和else放到同一行：



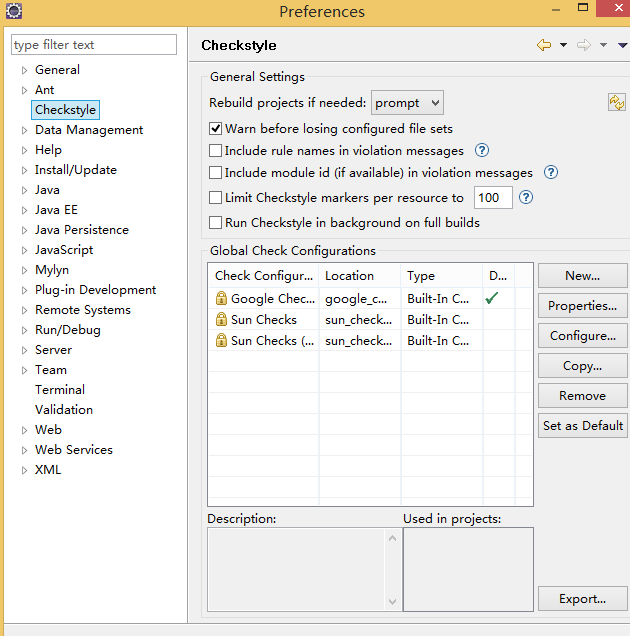
重新检查代码：



可以发现已经改好了。警告已经消失。

3.配置：

打开 Window→Preferences→CheckStyle:



二．FindBugs使用心得体会：（静态逻辑bug检查）

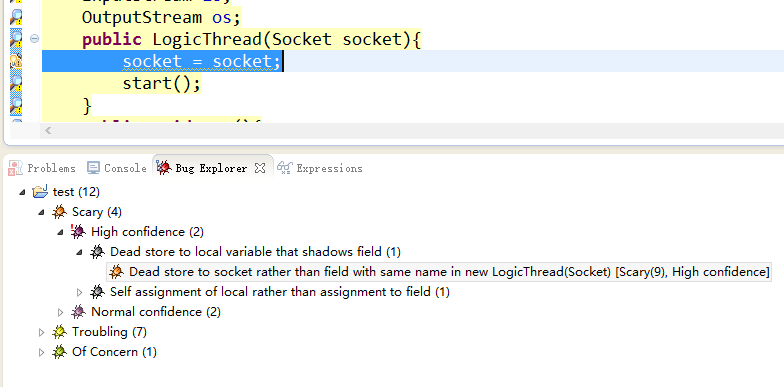
1. FindBugs的安装：

首先打开eclipse平台，点开help这个menu，点击Install New Software出现Available Software，然后点击add这个button添加想添加的Software，手动输入name和location。然后点击ok按照正常安装软件的步骤进行下去即可安装。

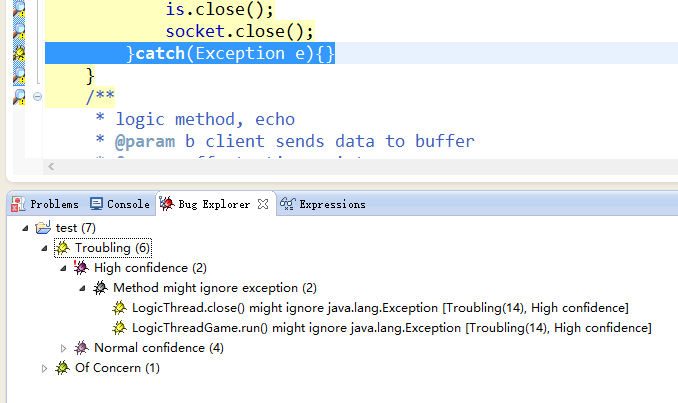
2.FindBugs的使用：

右击test项目，会有一个Find Bugs选项→Find Bugs，点击后开始进行扫描代码分析。（分析结果可以选择保存我xml文件。）对test静态逻辑结构检查结果如下：

检查后会出现scary，troubling和of concern三个级别的提示。同时有high confidence和normal confidence两个确信度级别。



根据Scary显示，把“socket=socket”改成“this.socket=socket”



Scary消失。

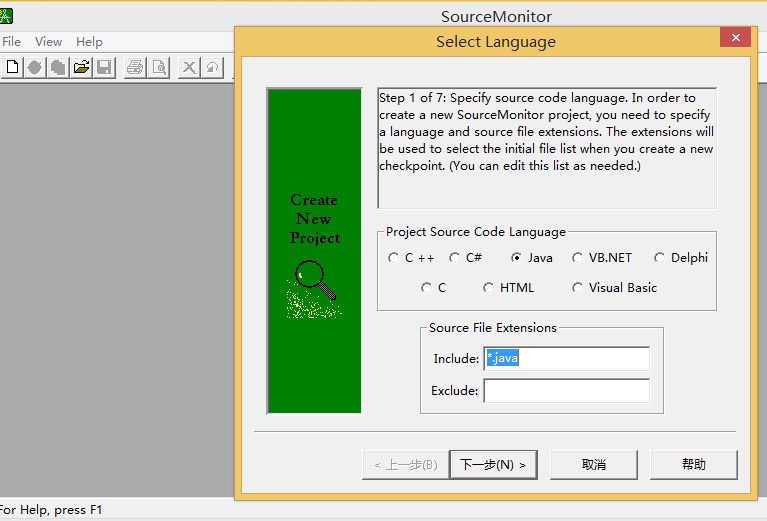
1. SourceMonitor使用心得体会：（代码质量自动检查）
2. SourceMonitor的安装

搜素SourceMonitor即可下载。安装即可。

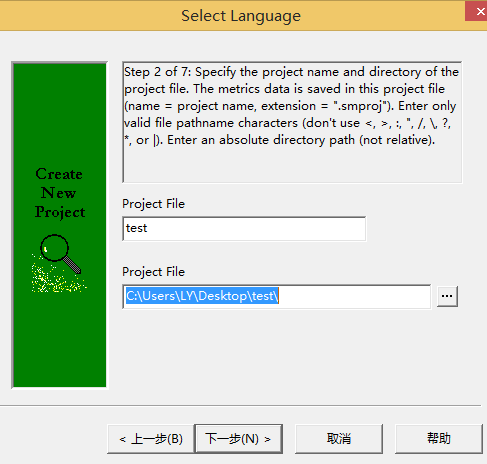
SourceMonitor可以处理多种语言。最终结果可以绘成图，输出打印。

1. SourceMonitor的使用：

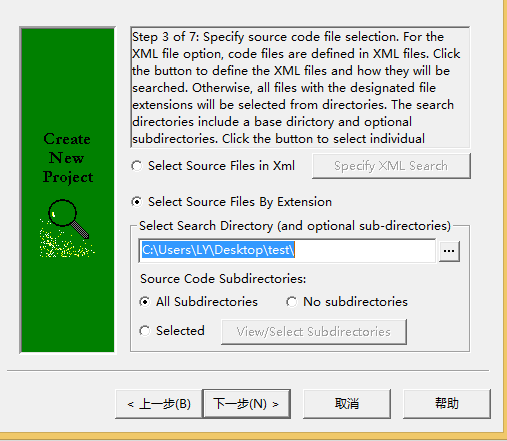
new一个project：



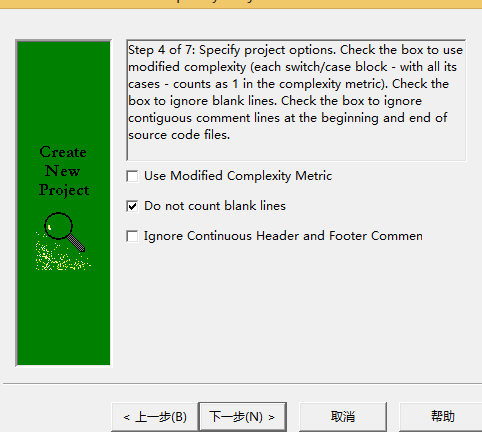
填好文件名和要保存的路径：

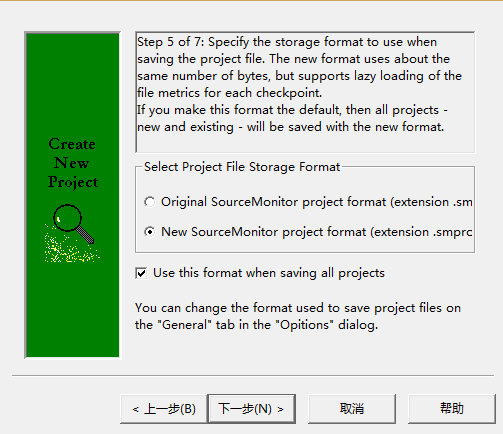


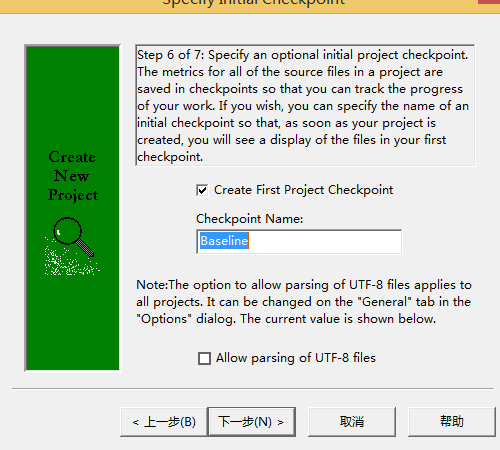
选择要检查的源文件路径：

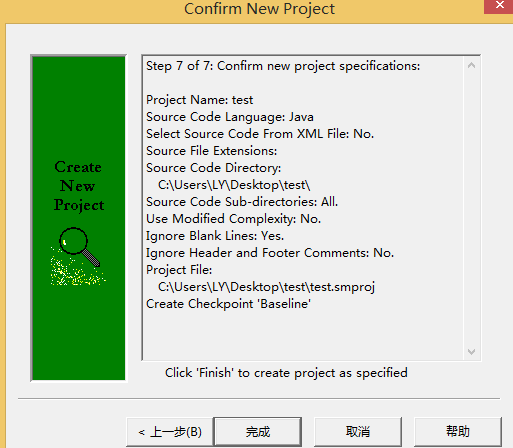


我选择了“不要算上空白行”：



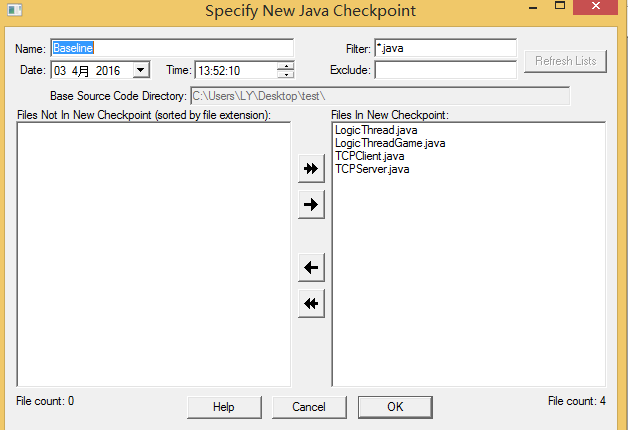




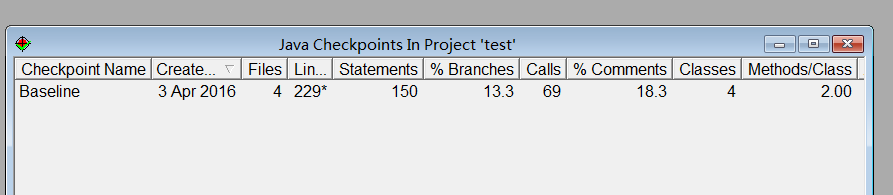


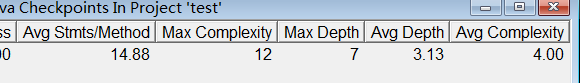
点击完成：

不检查的文件在左侧空白栏里，检查的文件在右侧。所以我要检查的4个java文件在右侧：



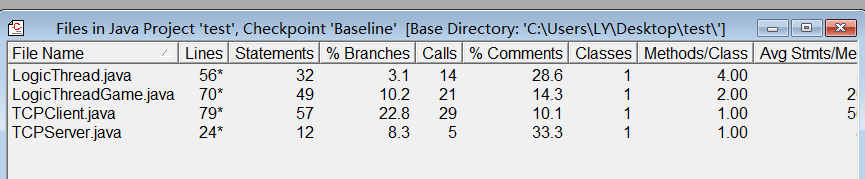
点击ok即开始扫描检查：



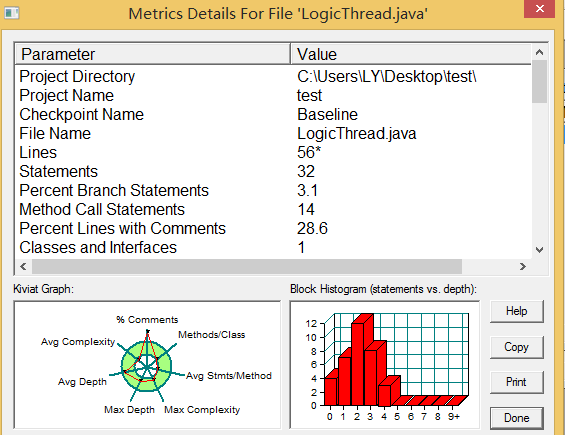


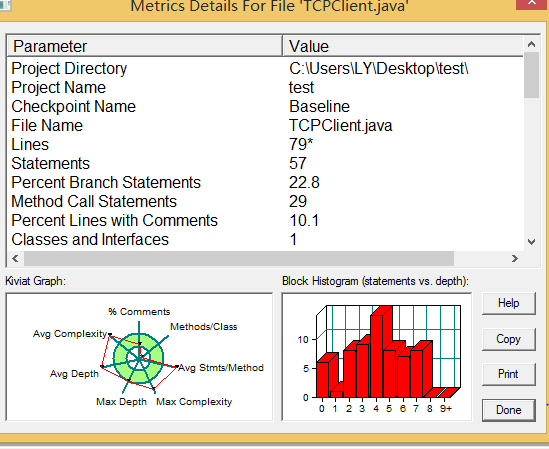
分析结果可以看出：我们有了Baseline第一个版本，也就是最初始的一个版本。Created On是指日期时间：2016年4月3号，Files指文件个数即有多少个文件：4个文件，Lines指文件有多少行：229行，Statements指文件有多少个语句：150个语句，Branches指分支，Comments指注释，后面还有类的声明以及方法等等一些总结性的数字。如果我们想要看一些细节性的东西，进行双击即可。

例如双击4：



再进行双击我们可以看见一些可视化的图表：





我截了两个图作为对比，根据结果可知这段代码的复杂度、深度等已经超出了正常范围。我们可以根据这一结果对代码进行重构，重构后再在此工程中新建 CheckPoint，进行新的分析。

图表里面会有一些详细的数据。这也是SourceMonitor最大的特点之一，他可以把这些结果print出来。同时可以把结果存储成xml格式，将来想要对结果进行再处理比较容易。