

CheckStyle、FindBugs、SourceMonitor 使用心得

我把下载四个 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：

Warnings (100 of 329 items)				
Warning	'}' at column 17 should be on the same li	TCPCClient.java	/test/src	line 66
	Checkstyle P...			

双击一下，会跳转到第 17 行代码：

```
os.write("quit".getBytes());
break;
}
} else{
    System.out.println("Input error");
}
```

根据警告，我把这个 “}” 和 else 放到同一行：

```
os.write("quit".getBytes());
break;
}
} else{
    System.out.println("Input error");
}
```

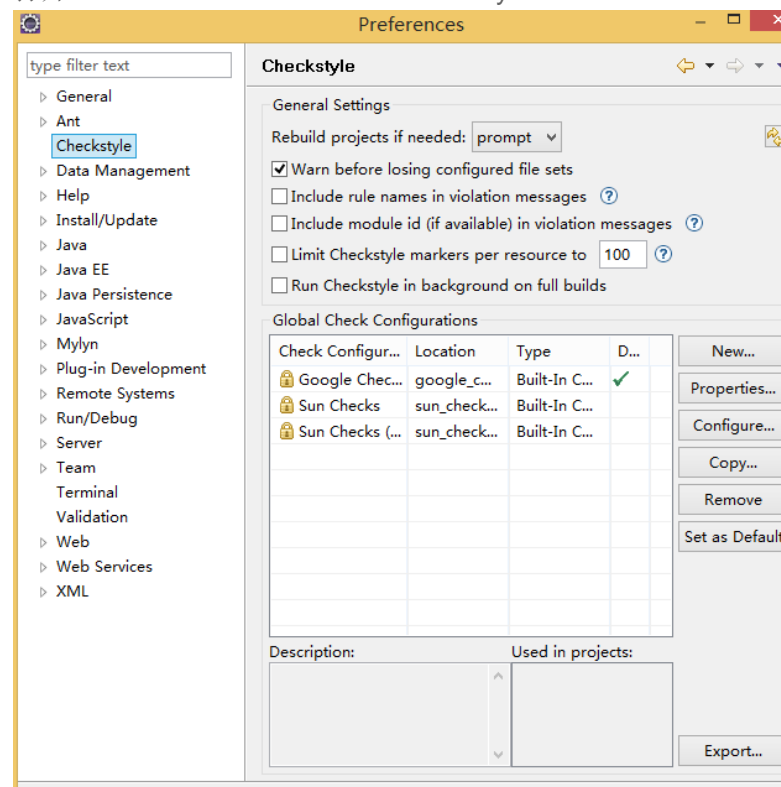
重新检查代码：

Warnings (100 of 327 items)				
Warning	'block' child have incorrect indentation le	TCPCClient.java	/test/src	line 49
	Checkstyle P...			
Warning	'block' child have incorrect indentation le	TCPCClient.java	/test/src	line 50
	Checkstyle P...			
Warning	'block' child have incorrect indentation le	TCPCClient.java	/test/src	line 52
	Checkstyle P...			
Warning	'block' child have incorrect indentation le	TCPCClient.java	/test/src	line 53
	Checkstyle P...			
Warning	'block' child have incorrect indentation le	TCPCClient.java	/test/src	line 55
	Checkstyle P...			

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)

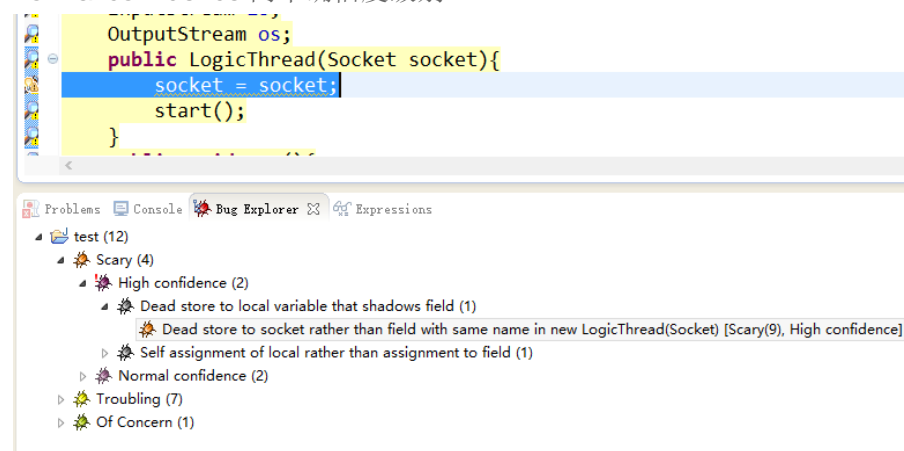
1. FindBugs 的安装:

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。

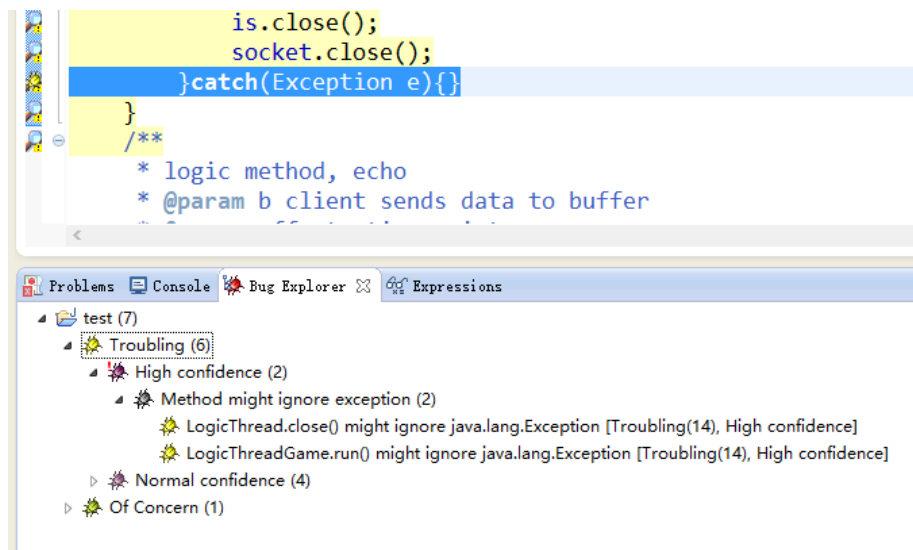
2.FindBugs 的使用:

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。

三. SourceMonitor 使用心得体会：（代码质量自动检查）

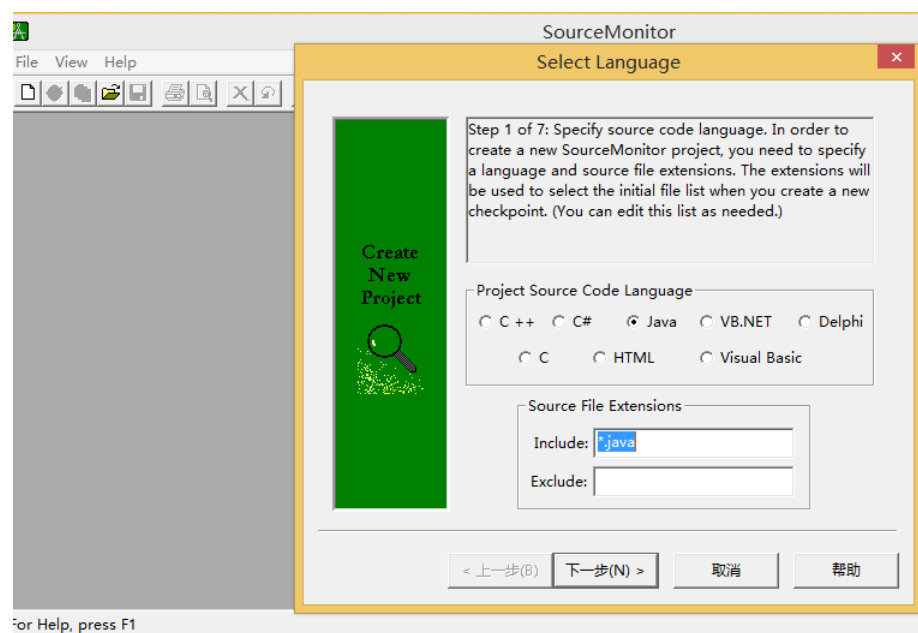
1. SourceMonitor 的安装

搜索 SourceMonitor 即可下载。安装即可。

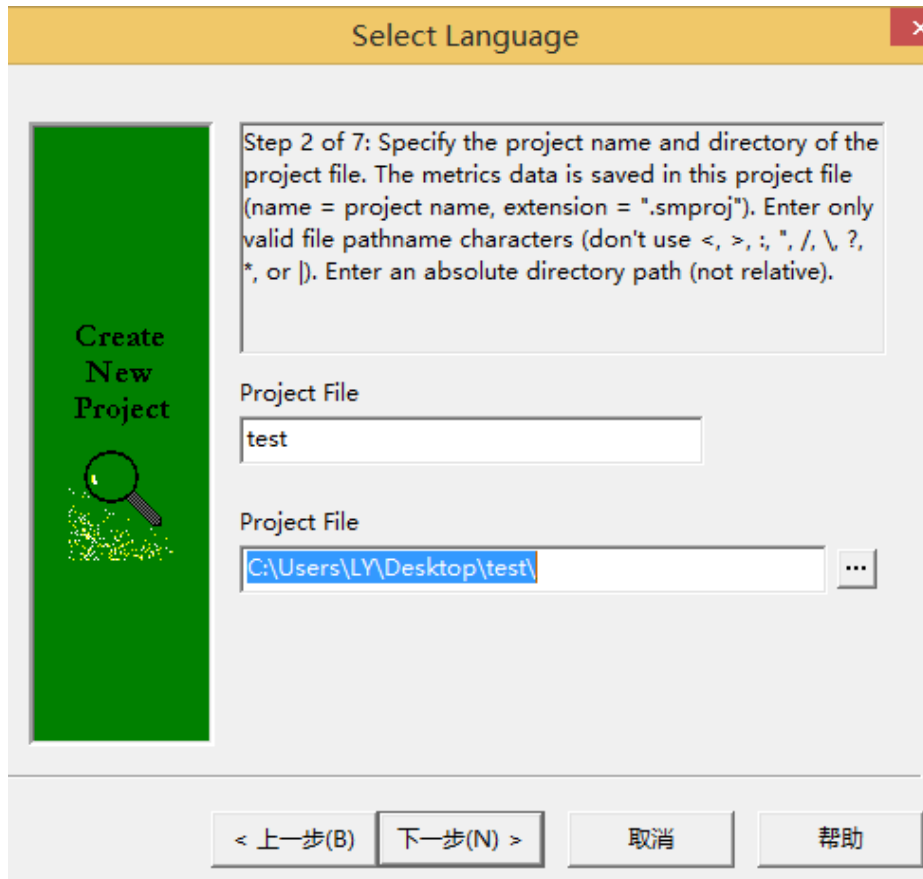
SourceMonitor 可以处理多种语言。最终结果可以绘成图，输出打印。

2. SourceMonitor 的使用：

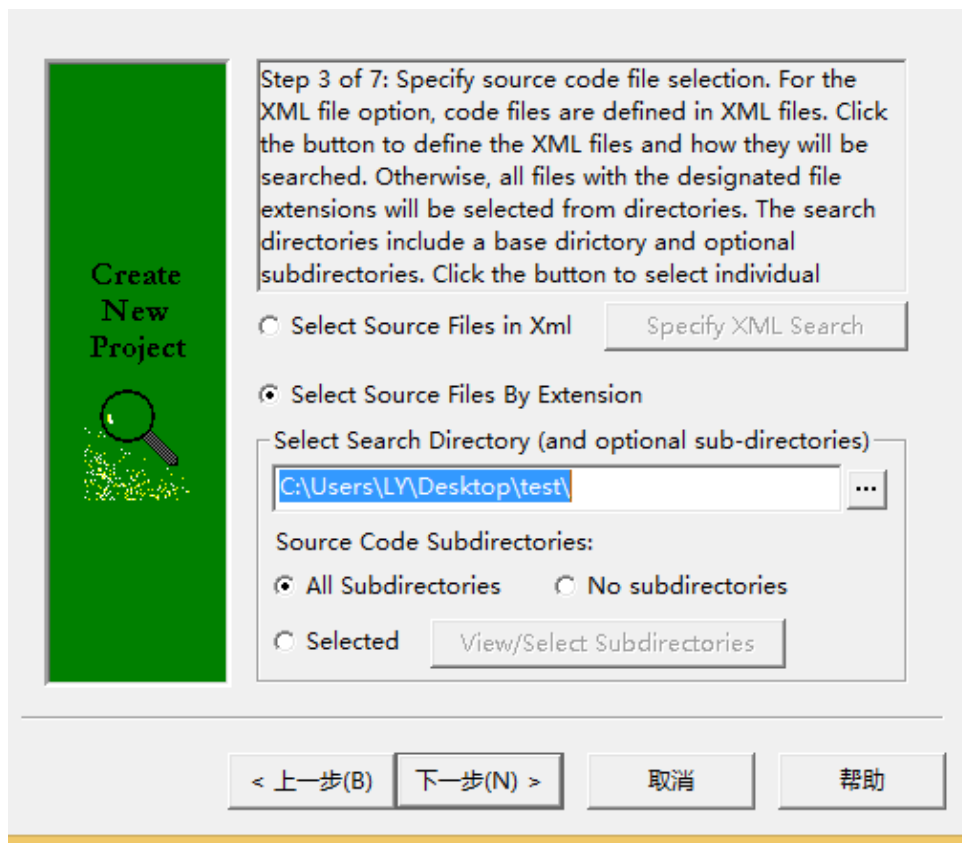
new 一个 project:



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”：

Create
New
Project



Step 4 of 7: Specify project options. Check the box to use modified complexity (each switch/case block - with all its cases - counts as 1 in the complexity metric). Check the box to ignore blank lines. Check the box to ignore contiguous comment lines at the beginning and end of source code files.

☐ Use Modified Complexity Metric

☒ Do not count blank lines

☐ Ignore Continuous Header and Footer Commen


< 上一步(B)

下一步(N) >

取消

帮助

Create
New
Project



Step 5 of 7: Specify the storage format to use when saving the project file. The new format uses about the same number of bytes, but supports lazy loading of the file metrics for each checkpoint.
If you make this format the default, then all projects - new and existing - will be saved with the new format.

Select Project File Storage Format

☐ Original SourceMonitor project format (extension .sm

☒ New SourceMonitor project format (extension .smprc

☒ Use this format when saving all projects

You can change the format used to save project files on the "General" tab in the "Opitions" dialog.

< 上一步(B)


下一步(N) >

取消

帮助

Specify Initial Checkpoint

Create New Project



Step 6 of 7: Specify an optional initial project checkpoint. The metrics for all of the source files in a project are saved in checkpoints so that you can track the progress of your work. If you wish, you can specify the name of an initial checkpoint so that, as soon as your project is created, you will see a display of the files in your first checkpoint.

☒ Create First Project Checkpoint

Checkpoint Name:

Note: The option to allow parsing of UTF-8 files applies to all projects. It can be changed on the "General" tab in the "Options" dialog. The current value is shown below.

☐ Allow parsing of UTF-8 files

< 上一步(B)


下一步(N) >

取消

帮助

Confirm New Project

Create New Project



Step 7 of 7: Confirm new project specifications:

Project Name: test
Source Code Language: Java
Select Source Code From XML File: No.
Source File Extensions:
Source Code Directory:
C:\Users\LY\Desktop\test\
Source Code Sub-directories: All.
Use Modified Complexity: No.
Ignore Blank Lines: Yes.
Ignore Header and Footer Comments: No.
Project File:
C:\Users\LY\Desktop\test\test.smproj
Create Checkpoint 'Baseline'

Click 'Finish' to create project as specified

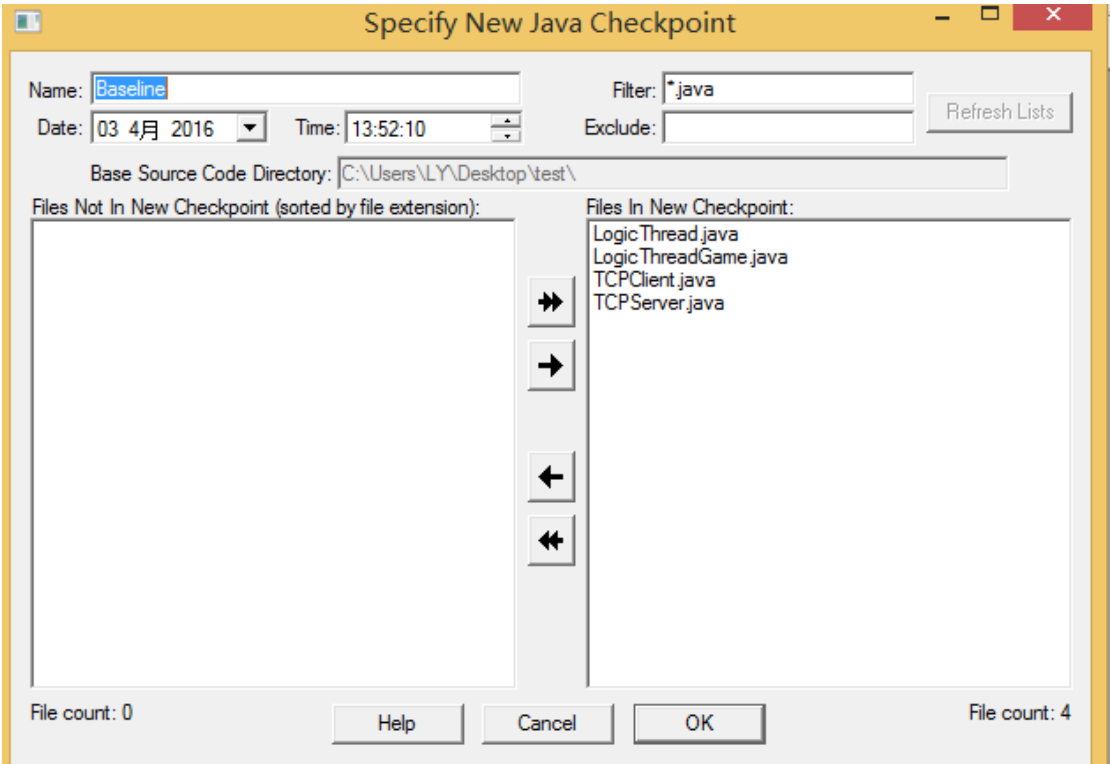
< 上一步(B)

完成

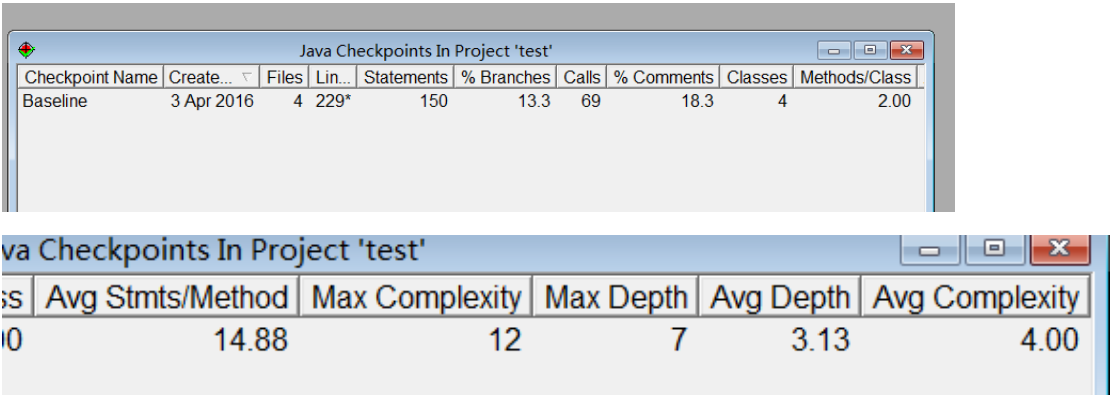
取消

帮助

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：

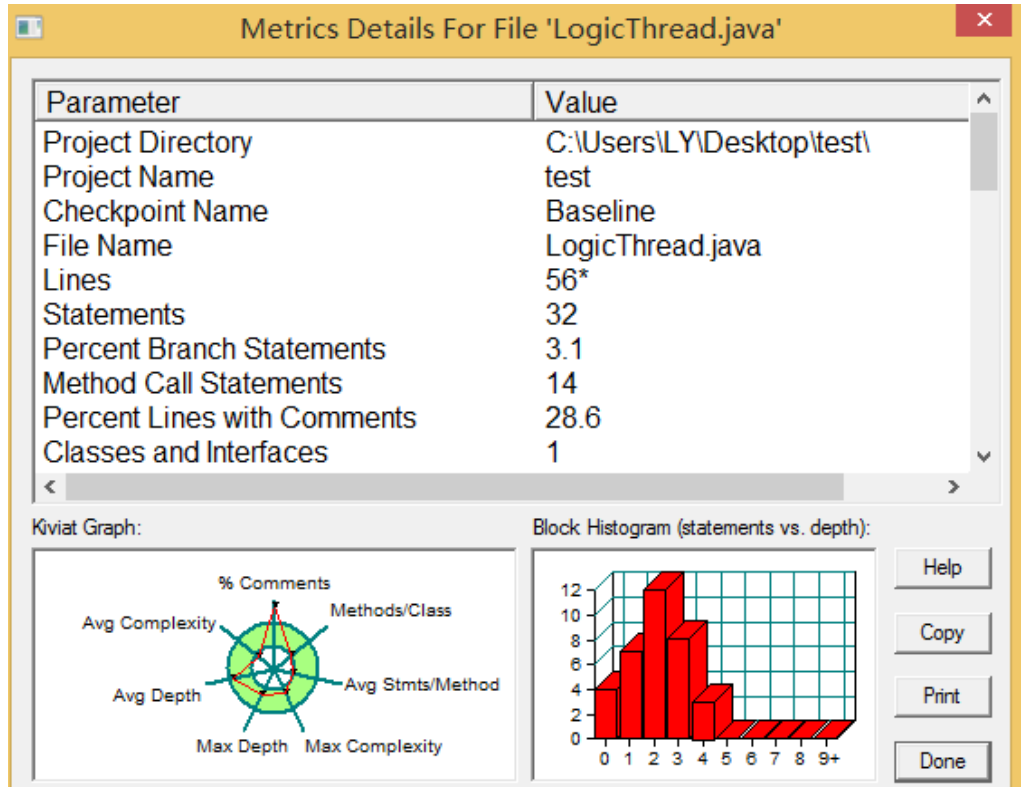


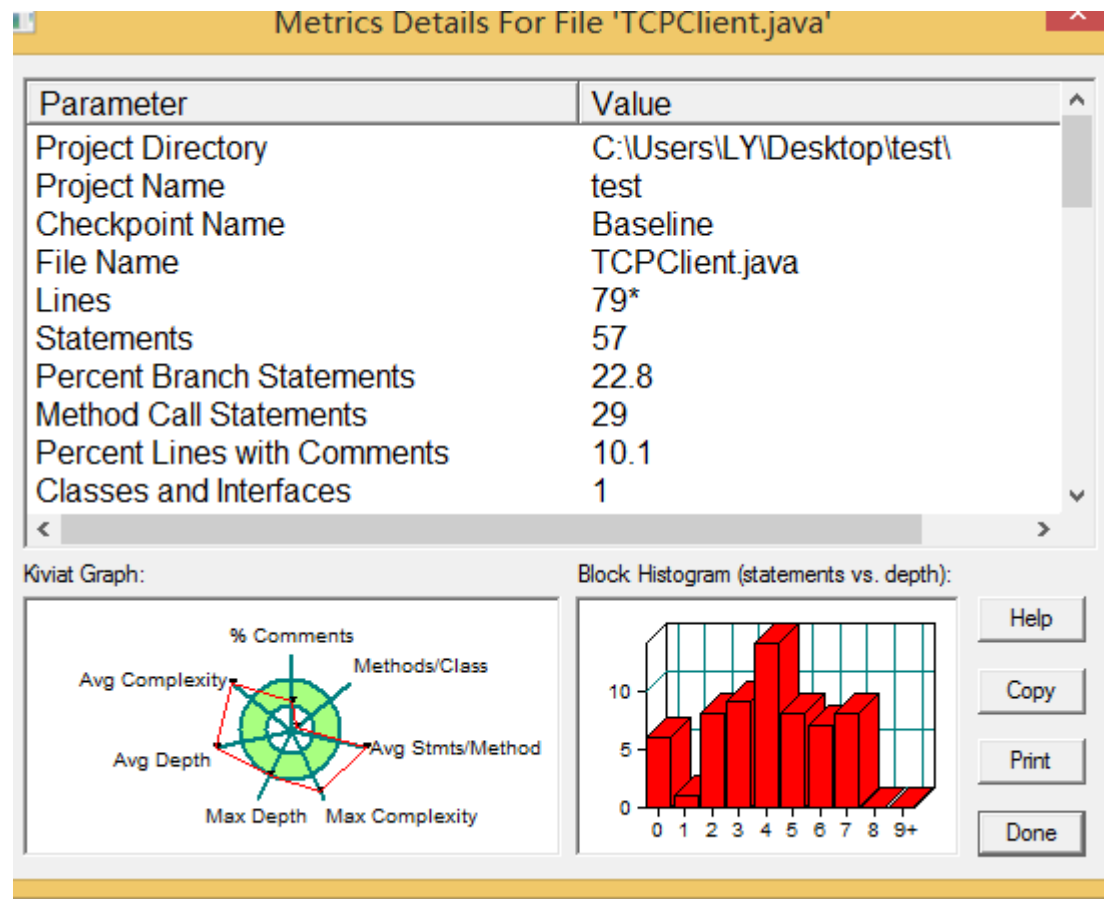
分析结果可以看出：我们有了 Baseline 第一个版本，也就是最初始的一个版本。Created On 是指日期时间：2016 年 4 月 3 号，Files 指文件个数即有多少个文件：4 个文件，Lines 指文件有多少行：229 行，Statements 指文件有多少个语句：150 个语句，Branches 指分支，Comments 指注释，后面还有类的声明以及方法等等一些总结性的数字。如果我们想要看一些细节性的东西，进行双击即可。

例如双击 4：

Files in Java Project 'test', Checkpoint 'Baseline' [Base Directory: 'C:\Users\LY\Desktop\test\']								
File Name	Lines	Statements	% Branches	Calls	% Comments	Classes	Methods/Class	Avg Stmt/Me
LogicThread.java	56*	32	3.1	14	28.6	1	4.00	
LogicThreadGame.java	70*	49	10.2	21	14.3	1	2.00	2
TCPClient.java	79*	57	22.8	29	10.1	1	1.00	5
TCPServer.java	24*	12	8.3	5	33.3	1	1.00	

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。