Lab 3.1 Using splint for C static analysis

1 Goals

- *Install splint;*
- *Finish code samples with 2 different kinds of problems which can be detected by Splint. You can choose any 2 of 11 problems as above.*
- *Use splint to detect the 2 kinds of problems. Descibe your observations in your report.*

2 Steps

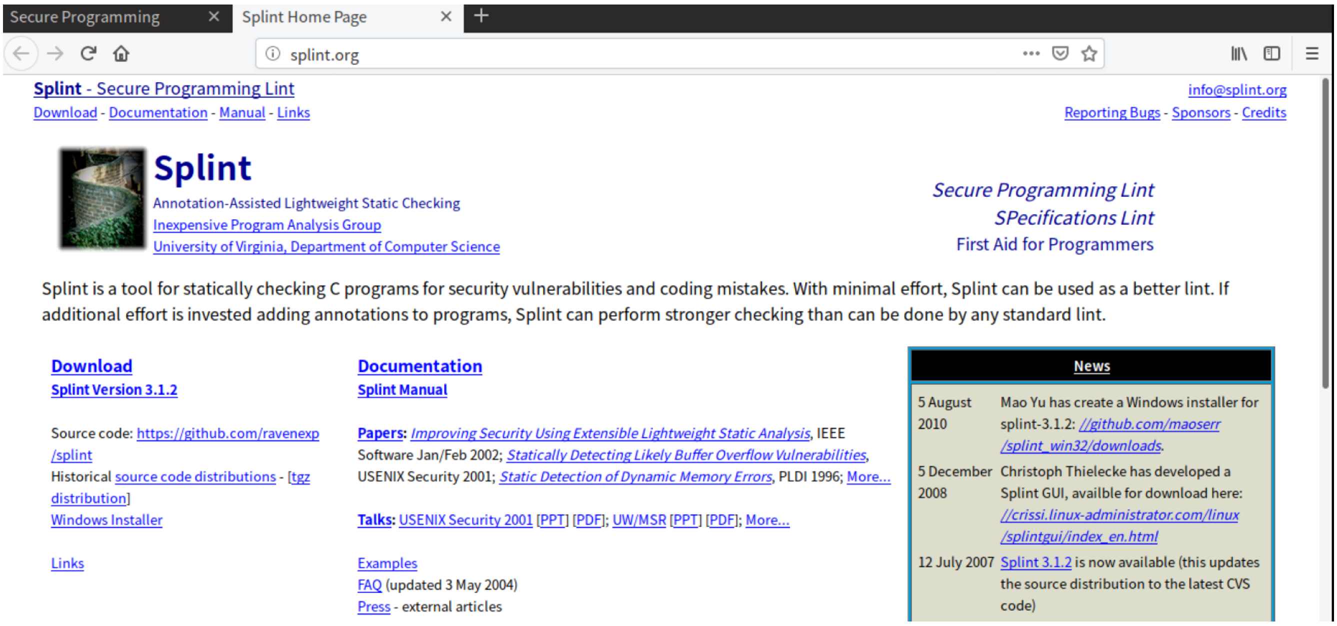
在网站中下载Splint：http://www.splint.org/downloads/splint-3.1.2.src.tgz

下载到本地后通过`tar zxvf splint-3.1.2.src.tgz`命令进行解压

然后通过`mkdir /usr/local/splint`创建一个新的文件夹用于安装splint

通过`./configure –prefix=/usr/local/splint`进行配置

通过`make install` 安装程序



Splint - Secure Programming Lint
Download - Documentation - Manual - Links

info@splint.org
Reporting Bugs - Sponsors - Credits

Splint
Annotation-Assisted Lightweight Static Checking
Inexpensive Program Analysis Group
University of Virginia, Department of Computer Science

Secure Programming Lint
SPecifications Lint
First Aid for Programmers

Splint is a tool for statically checking C programs for security vulnerabilities and coding mistakes. With minimal effort, Splint can be used as a better lint. If additional effort is invested adding annotations to programs, Splint can perform stronger checking than can be done by any standard lint.

Download
Splint Version 3.1.2

Source code: https://github.com/ravenexp/splint
Historical source code distributions - [tgz distribution]
Windows Installer

Links

Documentation
Splint Manual

Papers: *Improving Security Using Extensible Lightweight Static Analysis*, IEEE Software Jan/Feb 2002; *Statically Detecting Likely Buffer Overflow Vulnerabilities*, USENIX Security 2001; *Static Detection of Dynamic Memory Errors*, PLDI 1996; More...

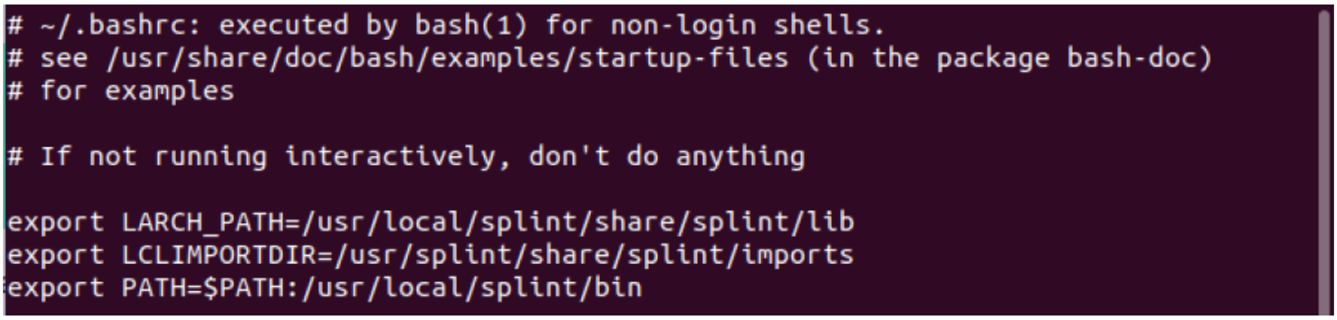
Talks: USENIX Security 2001 [PPT] [PDF]; UW/MSR [PPT] [PDF]; More...

Examples
FAQ (updated 3 May 2004)
Press - external articles

News	
5 August 2010	Mao Yu has create a Windows installer for splint-3.1.2: *//github.com/maoserr/splint_win32/downloads*.
5 December 2008	Christoph Thielecke has developed a Splint GUI, availble for download here: *//crissi.linux-administrator.com/linux/splintgui/index_en.html*
12 July 2007	Splint 3.1.2 is now available (this updates the source distribution to the latest CVS code)

配置环境变量`vi ~/.bashrc`，向文件中添加如下代码后，执行`source ~/.bashrc`

```
export LARCH_PATH=/usr/local/splint/share/splint/lib
export LCLIMPORTDIR=/usr/splint/share/splint/imports
export PATH=$PATH:/usr/local/splint/bin
```



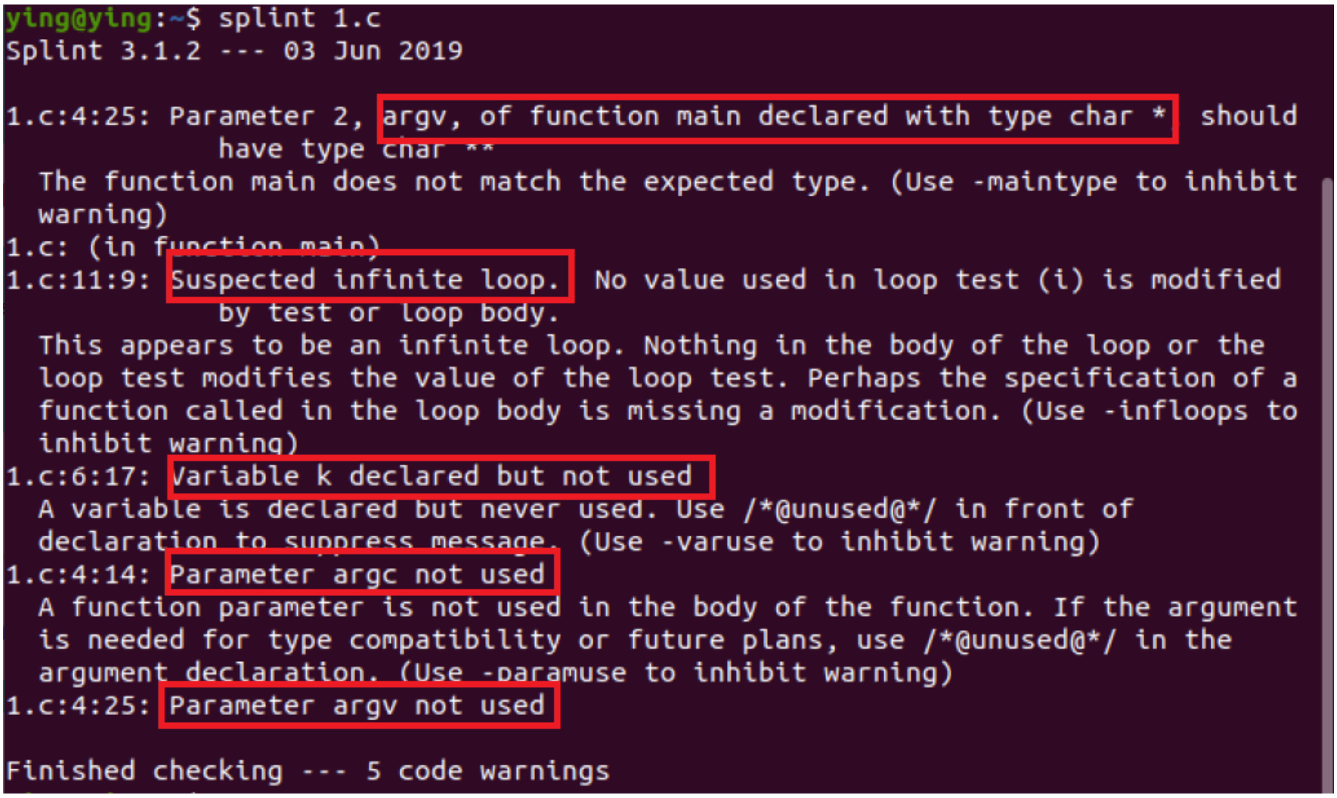
```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything

export LARCH_PATH=/usr/local/splint/share/splint/lib
export LCLIMPORTDIR=/usr/splint/share/splint/imports
export PATH=$PATH:/usr/local/splint/bin
```

然后准备一段有漏洞的C程序`1.c`如下：

```c
#include <stdio.h>
#include <string.h>

int main(int argc,char* argv) {  //pointer
    //unused declarations
    int i = 2,j,k;


    //likely infinite loop

    while (i==2) {
        j = i-1;
    }

    return 0;

}
```

运行`splint 1.c`得到如下结果：可以看到Splint指出了之前设定的问题：

- 死循环
- 定义了但是没有使用的变量
- 指针问题



```
ying@ying:~$ splint 1.c
Splint 3.1.2 --- 03 Jun 2019

1.c:4:25: Parameter 2, argv, of function main declared with type char *  should
             have type char **
  The function main does not match the expected type. (Use -maintype to inhibit
  warning)
1.c: (in function main)
1.c:11:9: Suspected infinite loop.  No value used in loop test (i) is modified
             by test or loop body.
  This appears to be an infinite loop. Nothing in the body of the loop or the
  loop test modifies the value of the loop test. Perhaps the specification of a
  function called in the loop body is missing a modification. (Use -infloops to
  inhibit warning)
1.c:6:17: Variable k declared but not used
  A variable is declared but never used. Use /*@unused@*/ in front of
  declaration to suppress message. (Use -varuse to inhibit warning)
1.c:4:14: Parameter argc not used
  A function parameter is not used in the body of the function. If the argument
  is needed for type compatibility or future plans, use /*@unused@*/ in the
  argument declaration. (Use -paramuse to inhibit warning)
1.c:4:25: Parameter argv not used

Finished checking --- 5 code warnings
```

Lab 3.2 Using eclipse for java static analysis

1 Goals

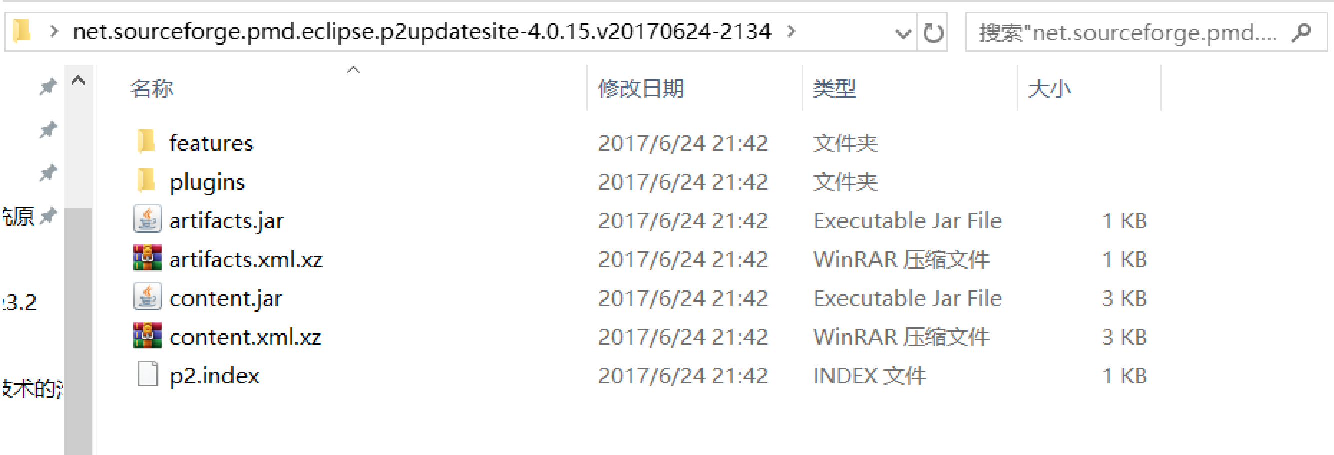
- *Install plugins in Java;*
- *Learn to check Java code by using static code analyzers in Eclipse. Descibe your observations in your report.*

2 Steps

本实验选用了PMD插件。通过GitHub的链接下载离线版的插件：

https://github.com/pmd/pmd-eclipse-plugin/releases/tag/4.4.0.v20190526-1012

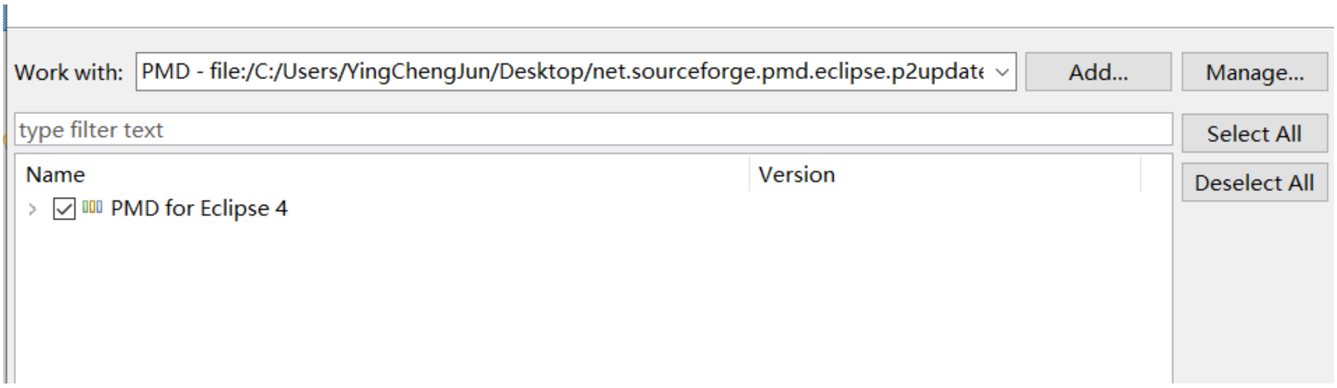
下载完成后解压文件：



› net.sourceforge.pmd.eclipse.p2updatesite-4.0.15.v20170624-2134 ›
搜索"net.sourceforge.pmd....

名称	修改日期	类型	大小
features	2017/6/24 21:42	文件夹	
plugins	2017/6/24 21:42	文件夹	
artifacts.jar	2017/6/24 21:42	Executable Jar File	1 KB
artifacts.xml.xz	2017/6/24 21:42	WinRAR 压缩文件	1 KB
content.jar	2017/6/24 21:42	Executable Jar File	3 KB
content.xml.xz	2017/6/24 21:42	WinRAR 压缩文件	3 KB
p2.index	2017/6/24 21:42	INDEX 文件	1 KB

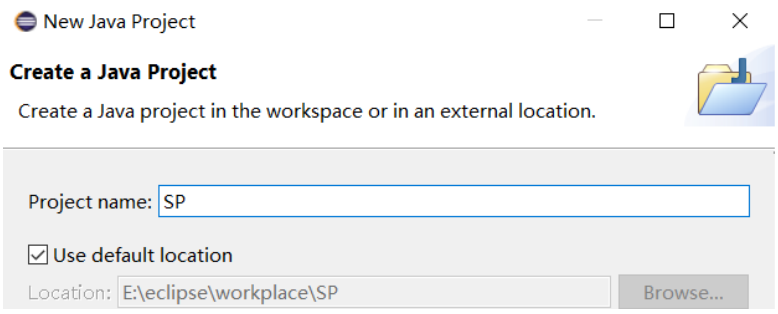
在Eclipse界面中选择Help->Install new software，然后点击Add将解压后的文件夹导入，接下来一路点击Next，安装完成后重启。



Work with: PMD - file:/C:/Users/YingChengJun/Desktop/net.sourceforge.pmd.eclipse.p2updat‹

Name	Version
☑ PMD for Eclipse 4	

创建一个Java Project



New Java Project

Create a Java Project

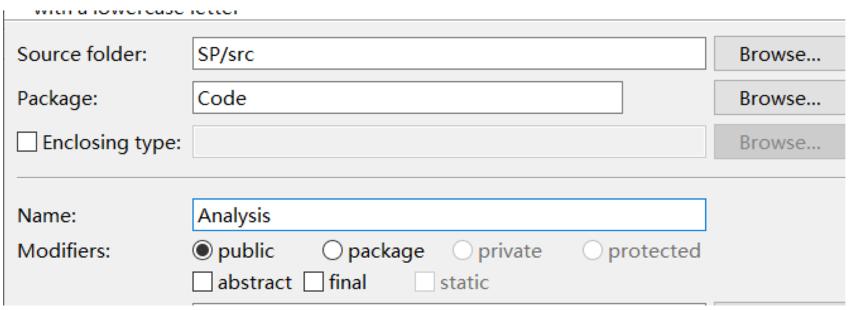
Create a Java project in the workspace or in an external location.

Project name: SP

☑ Use default location

Location: E:\eclipse\workplace\SP

创建一个包以及一个文件



with a lowercase letter

Source folder: SP/src

Package: Code

□ Enclosing type:

Name: Analysis

Modifiers: ● public ○ package ○ private ○ protected

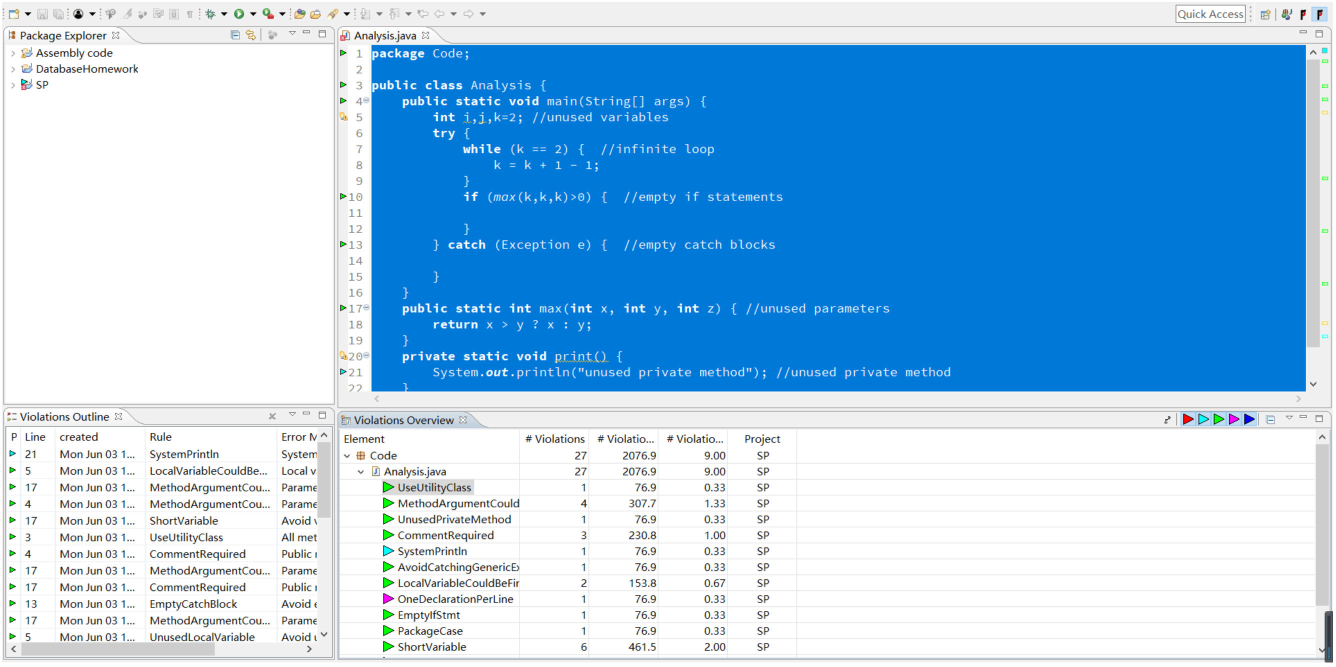
□ abstract □ final □ static

编辑一段有问题的代码

```java
package Code;
public class Analysis {
    public static void main(String[] args) {
        int i,j,k=2; //unused variables
        try {
            while (k == 2) {  //infinite loop
                k = k + 1 - 1;
            }
            if (max(k,k,k)>0) {  //empty if statements

            }
        } catch (Exception e) {  //empty catch blocks

        }
    }
    public static int max(int x, int y, int z) { //unused parameters
```

```
        return x > y ? x : y;
    }
    private static void print() {
        System.out.println("unused private method"); //unused private method
    }
}
```

右键选择PMD->CheckCode，如下图所示：可以看到下面有对源代码的静态分析



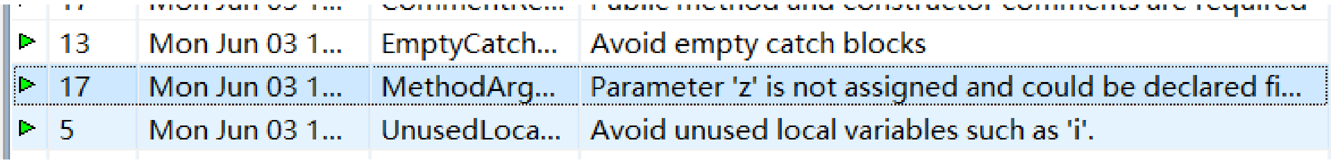
```
package Code;

public class Analysis {
    public static void main(String[] args) {
        int i,j,k=2; //unused variables
        try {
            while (k == 2) {  //infinite loop
                k = k + 1 - 1;
            }
            if (max(k,k,k)>0) {  //empty if statements

            }
        } catch (Exception e) {  //empty catch blocks

        }
    }
    public static int max(int x, int y, int z) { //unused parameters
        return x > y ? x : y;
    }
    private static void print() {
        System.out.println("unused private method"); //unused private method
    }
```

P	Line	created	Rule	Error M
▷	21	Mon Jun 03 1...	SystemPrintln	System
▷	5	Mon Jun 03 1...	LocalVariableCouldBe...	Local v
▷	17	Mon Jun 03 1...	MethodArgumentCou...	Parame
▷	4	Mon Jun 03 1...	MethodArgumentCou...	Parame
▷	17	Mon Jun 03 1...	ShortVariable	Avoid v
▷	3	Mon Jun 03 1...	UseUtilityClass	All met
▷	4	Mon Jun 03 1...	CommentRequired	Public r
▷	17	Mon Jun 03 1...	MethodArgumentCou...	Parame
▷	17	Mon Jun 03 1...	CommentRequired	Public r
▷	13	Mon Jun 03 1...	EmptyCatchBlock	Avoid e
▷	17	Mon Jun 03 1...	MethodArgumentCou...	Parame
▷	5	Mon Jun 03 1...	UnusedLocalVariable	Avoid u

Element	# Violations	# Violatio...	# Violatio...	Project
Code	27	2076.9	9.00	SP
Analysis.java	27	2076.9	9.00	SP
▷ UseUtilityClass	1	76.9	0.33	SP
▷ MethodArgumentCould	4	307.7	1.33	SP
▷ UnusedPrivateMethod	1	76.9	0.33	SP
▷ CommentRequired	3	230.8	1.00	SP
▷ SystemPrintln	1	76.9	0.33	SP
▷ AvoidCatchingGenericE	1	76.9	0.33	SP
▷ LocalVariableCouldBeFir	2	153.8	0.67	SP
▷ OneDeclarationPerLine	1	76.9	0.33	SP
▷ EmptyIfStmt	1	76.9	0.33	SP
▷ PackageCase	1	76.9	0.33	SP
▷ ShortVariable	6	461.5	2.00	SP

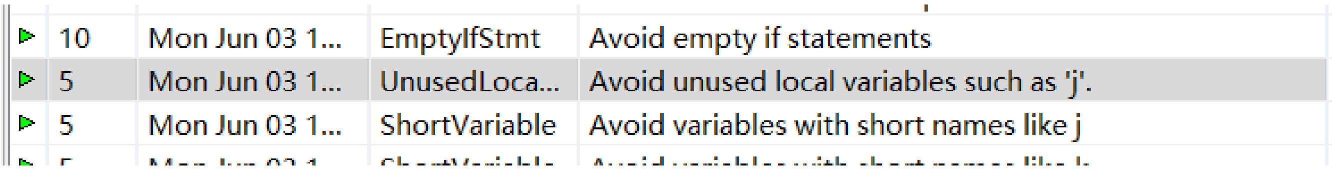
PMD插件对该代码的检测完成了以下测试点：

- 未使用的函数形参：



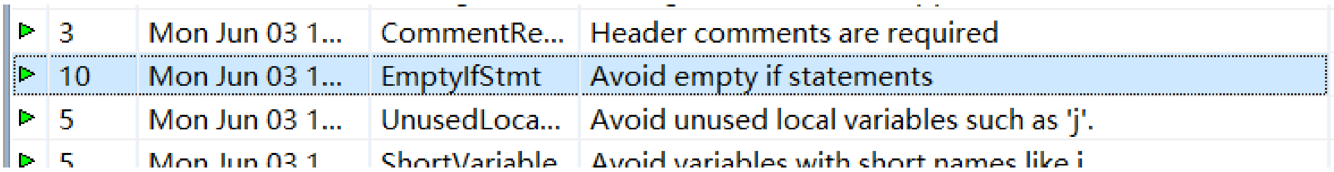
▷	17	Mon Jun 03 1...	CommentRe...	Public method and constructor comments are required
▷	13	Mon Jun 03 1...	EmptyCatch...	Avoid empty catch blocks
▷	17	Mon Jun 03 1...	MethodArg...	Parameter 'z' is not assigned and could be declared fi...
▷	5	Mon Jun 03 1...	UnusedLoca...	Avoid unused local variables such as 'i'.

- 未使用的局部变量



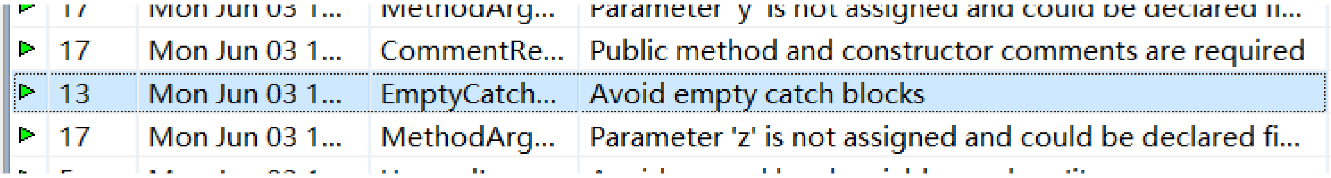
▷	10	Mon Jun 03 1...	EmptyIfStmt	Avoid empty if statements
▷	5	Mon Jun 03 1...	UnusedLoca...	Avoid unused local variables such as 'j'.
▷	5	Mon Jun 03 1...	ShortVariable	Avoid variables with short names like j
▷	5	Mon Jun 03 1...	ShortVariable	Avoid variables with short names like k

- 空if块



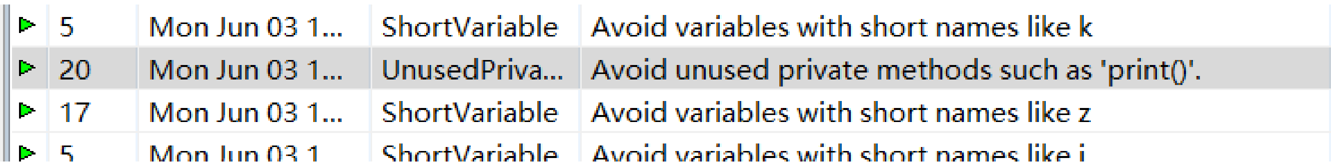
▷	3	Mon Jun 03 1...	CommentRe...	Header comments are required
▷	10	Mon Jun 03 1...	EmptyIfStmt	Avoid empty if statements
▷	5	Mon Jun 03 1...	UnusedLoca...	Avoid unused local variables such as 'j'.
▷	5	Mon Jun 03 1...	ShortVariable	Avoid variables with short names like i

- 空catch块



▶	17	Mon Jun 03 1...	MethodArg...	Parameter 'y' is not assigned and could be declared fi...
▶	17	Mon Jun 03 1...	CommentRe...	Public method and constructor comments are required
▶	13	Mon Jun 03 1...	EmptyCatch...	Avoid empty catch blocks
▶	17	Mon Jun 03 1...	MethodArg...	Parameter 'z' is not assigned and could be declared fi...

- 没有使用的私有方法



▶	5	Mon Jun 03 1...	ShortVariable	Avoid variables with short names like k
▶	20	Mon Jun 03 1...	UnusedPriva...	Avoid unused private methods such as 'print()'.
▶	17	Mon Jun 03 1...	ShortVariable	Avoid variables with short names like z
▶	5	Mon Jun 03 1...	ShortVariable	Avoid variables with short names like i

但是PMD插件没有发现第6~8行处的死循环问题，说明插件还是存在一定的局限性

```
while (k == 2) {
    k = k + 1 - 1;
}
```