

丁老九 · May 17, 2016 · Last by 丁老九 replied at June 20, 2017 · Last modified by admin 思寒_seveniruby · 4263 hits

□ 本帖已被设为精华帖！

阅读该文章前，最好已经对PMD有了初步的认识和了解，可参考[静态分析工具PMD使用说明](#) □ 目录 ▾

准备工作

首先在[PMD官网](#)下载最新版本的文件，目前最新版本是5.4.1。

下载pmd-bin-5.4.1.zip和pmd-src-5.4.1.zip之后解压备用。

pmd-src-5.4.1是PMD源码包，是无法直接执行的。

pmd-bin-5.4.1是PMD的可执行包。

目录简介

- pmd-bin-5.4.1 【PMD可执行版本】
 - bin
 - designer.bat 【界面工具，能将java源代码转化为AST（抽象语法树），个人推荐使用】
 - bgastviewer.bat 【界面工具，与designer.bat功能相似】
 - cpd.bat 【用来查找重复代码的工具，命令行版】
 - cpdgui.bat 【用来查找重复代码的工具，GUI版】
 - pmd.bat 【Window平台下运行PMD需要使用的文件】
 - run.sh 【Linux平台下运行PMD需要使用的文件】
 - lib 【该目录存放PMD运行依赖的jar包，包括第三方jar包和各种语言的模块jar包】
- pmd-src-5.4.1 【PMD源代码版本】
 - pmd-core 【PMD的核心执行调度模块】
 - pmd-java 【针对java语言的检测模块】
 - src ->main
 - java -> net -> sourceforge -> pmd -> lang->java 【目录太深，在此处聚合】
 - rule 【该目录下存放已经编写好的java规则文件】

- **basic** 【基础类规则】
- **AvoidBranchingStatementAsLastInLoopRule.java** 【避免在循环的最后使用分支语句】
- **AvoidMultipleUnaryOperatorsRule.java** 【避免一元运算符的多重使用】
- ... 【其他基础类的规则文件】
- **codesize** 【代码体积类规则】
- ... 【各种规则类别的目录，包含该类别的java编写的规则文件】
- **resources**
 - **rulesets** 【java规则对应的xml文件】
 - **java**
 - **android.xml** 【PMD运行时使用该文件会调用安卓类规则进行扫描】
 - **basic.xml** 【PMD运行时使用该文件会调用基础类规则进行扫描】
 - ... 【其他类别的规则xml文件】
- **etc**
 - **grammar**
 - **Java.jjt** 【AST抽象语法树生成所需的语法文件】
- **pmd-java8** 【新增对java1.8版本的支持模块】
- **pmd-javascript** 【针对javascript语言的检测模块】
- **pmd-jsp** 【针对jsp语言的检测模块】
- ... 【其余的主要是针对不同语言实现的独立的检测模块】

自定义规则实现思路

1. 明确想要自定义的规则。
2. 列举会触犯这种规则的所有不同的写法。
3. 使用**designer.bat**分析所有写法的抽象语法树的特点。
4. 编写规则代码捕捉这种特点。
5. 创建自己的xml规则文件，内容包括规则的相关信息。
6. 运行PMD扫描错误代码，验证是否能触发自定义规则。

下面以一个比较简单的规则举例，详细的阐述一下实现这个规则的具体步骤，帮助大家快速上手。
目前PMD支持两种编写规则的方法：

1. 使用Java进行编写
2. 使用XPath表达式 我首先选择第一种Java编写方式进行讲解。 ****

1. 明确想要自定义的规则

需要自定义的规则：**While**循环必须使用括号，**While**循环没有括号很容易困惑代码结构。所以下面以“**While**循环必须使用括号”这条规则为例。

2. 列举会触犯这种规则的所有不同的写法

写出问题样例的代码写法。

```
class Example {
    void bar() {
        while (baz)
            buz.doSomething();
    }
}
```

弄清楚样例代码是什么样子的，就成功了一半。

3. 使用designer.bat分析所有写法的抽象语法树的特点

PMD扫描时并不是直接使用源码；它使用 `JavaCC` 生成解析器来解析源代码并生成AST(抽象语法树)。你可以使用PMD自带的designer工具进行解析代码。

该工具所在目录：`pmd-bin-5.4.1/bin/designer.bat`

双击designer.bat后出现一个界面，在Source code中填入源代码，点击Go按钮：



以上样例代码解析成抽象语法树后如下：

```
CompilationUnit
  TypeDeclaration
    ClassDeclaration:(package private)
      UnmodifiedClassDeclaration(Example)
        ClassBody
          ClassBodyDeclaration
            MethodDeclaration:(package private)
              ResultType
                MethodDeclarator(bar)
                  FormalParameters
                    Block
                      BlockStatement
                        Statement
                          WhileStatement
                            Expression
                              PrimaryExpression
```

```
PrimaryPrefix
  Name: baz
Statement
  StatementExpression: null
PrimaryExpression
  PrimaryPrefix
    Name: buz.doSomething
  PrimarySuffix
    Arguments
```

图片中Abstract Syntax Tree/XPath/Symbol Table的位置就是抽象后的树形结构，这个树形结构和源代码是有对应关系的。

其中我们需要重点关注的 `WhileStatement` 的抽象树结构如下：

```
WhileStatement
  Expression
  Statement
  StatementExpression
```

这个是错误的代码示例的抽象树结构，如果While循环加上了括号，抽象树的结构就会变成：

```
WhileStatement
  Expression
  Statement
  Block
    BlockStatement
      Statement
      StatementExpression
```

这下能明显的看到了比之前多出了 `Block` 和 `BlockStatement` 这两个节点。

这样我们只需要写一个规则检查 `WhileStatement` 下有没有 `Block` 节点，如果没有 `Block` 节点，那就说明While语句后面没有大括号，就可以报警告知这里是有问题的。

顺便提一句，所有的结构信息，比如一个 `Statement` 节点后面可能跟着一个 `Block` 节点，这些都是 [在EBNF grammar](#)中定义的。比如在这个语法定义中，一个 `Statement` 的定义是这样的：

```
void Statement() :
{
  LOOKAHEAD( { isNextTokenAnAssert() } ) AssertStatement()
| LOOKAHEAD(2) LabeledStatement()
| Block()
| EmptyStatement()
| StatementExpression() ";"
| SwitchStatement()
| IfStatement()
| WhileStatement()
```

```
| DoStatement()  
| ForStatement()  
| BreakStatement()  
| ContinueStatement()  
| ReturnStatement()  
| ThrowStatement()  
| SynchronizedStatement()  
| TryStatement()  
}
```

以上代码列出了一个 `Statement` 节点后面所有的可能的节点类型。

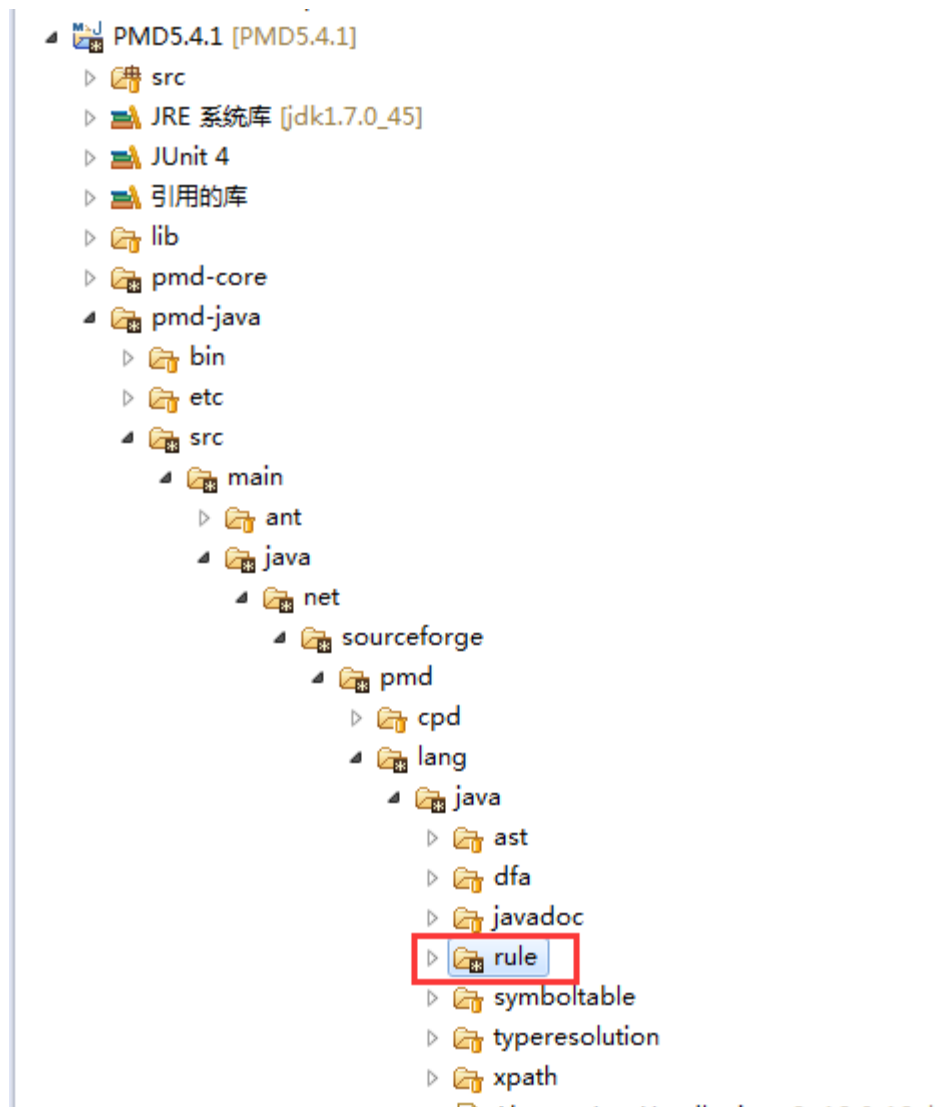
4. 编写规则代码捕捉这种特点

我们需要新建一个规则文件，也就是一个Java类，名称为WhileLoopsMustUseBracesRule.java。

新建的位置也是有要求的，以我在上文中介绍的目录结构为例：

新的规则类位置：pmd-src-5.4.1\pmd-java\src\main\java\net\sourceforge\pmd\lang\java\rule\这个目录下即可。

截图如下，标红处就是汇聚了所有规则文件的rule目录：



该类必须继承 `net.sourceforge.pmd.lang.java.rule.AbstractJavaRule` 。

```
import net.sourceforge.pmd.lang.java.rule.*;
public class WhileLoopsMustUseBracesRule extends AbstractJavaRule {
}
```

PMD工作原理就是在生成的抽象语法树中递归的遍历，直到找出要找的目标，然后返回结果。

接下来我们的目标就是在抽象语法树中找出 `WhileStatement` 节点下不存在 `Statement/Block` 这种结构的情况。

```
import net.sourceforge.pmd.lang.ast.*;
import net.sourceforge.pmd.lang.java.ast.*;
import net.sourceforge.pmd.lang.java.rule.*;

public class WhileLoopsMustUseBracesRule extends AbstractJavaRule {
    public Object visit(ASTWhileStatement node, Object data) {
        Node firstStmt = node.jjtGetChild(1);
        if (!hasBlockAsFirstChild(firstStmt)) {
```

```
        addViolation(data, node);
    }
    return super.visit(node, data);
}
private boolean hasBlockAsFirstChild(Node node) {
    return (node.jjtGetNumChildren() != 0 && (node.jjtGetChild(0) instanceof ASTBlock));
}
}
```

这段代码的主要意思：

1. 访问文件中的 `ASTWhileStatement` 节点
2. 获取 `ASTWhileStatement` 节点下第二个子节点
3. 判断第二个子节点的第一个子节点是不是 `ASTBlock` 节点
4. 如果不是 `ASTBlock` 节点，说明我们的目标实现，触犯规则
5. 使用 `addViolation(data, node);` 语句记录触犯该规则的节点相关数据

5. 创建自己的xml规则文件，内容包括规则的相关信息

现在规则已经写完了，我们需要告诉PMD运行时执行这条规则，就得将这个规则文件的相关信息放在XML规则集文件中。例如：`pmd-java/src/main/resources/rulesets/java/basic.xml`；这里面有很多规则的定义，复制粘贴一下，改成一个新的规则集文件，名字自己随便取：`mycustomrules.xml`，自己填充一下元素和属性。

name - `WhileLoopsMustUseBracesRule`

message - `Use braces for while loops`

class - 放哪都行. 注意，没有必要放在 `net.sourceforge.pmd` 目录下，可以放在 `com.yourcompany.util.pmd`

description - `Use braces for while loops`

example - 通过代码片段展示违反的规则样例

```
<?xml version="1.0"?>
<ruleset name="My custom rules"
  xmlns="http://pmd.sourceforge.net/ruleset/2.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://pmd.sourceforge.net/ruleset/2.0.0 http://pmd.sourceforge.net/ruleset_2_0_0.xsd">
  <rule name="WhileLoopsMustUseBracesRule"
    message="Avoid using 'while' statements without curly braces"
    class="WhileLoopsMustUseBracesRule">
    <description>
      Avoid using 'while' statements without using curly braces
    </description>
    <priority>3</priority>

    <example>
```

```
<![CDATA[
    public void doSomething() {
        while (true)
            x++;
    }
}]>
    </example>
</rule>
</ruleset>
```

6. 运行PMD扫描错误代码，验证是否能触发自定义规则

在执行前需要把你修改后的pmd-java重新打包：

在命令行中进入 `pmd-src-5.4.1\pmd-java` 目录中，执行 `mvn clean package`。

打包成功后，将 `pmd-src-5.4.1\pmd-java\target` 中的 `pmd-java-5.4.1.jar` 替换 `pmd-bin-5.4.1\lib` 目录中对应的jar包。

最后在 `pmd-bin-5.4.1\bin` 目录中执行

```
pmd.bat -d c:\path\to\my\src -f xml -R c:\path\to\mycustomrules.xml
```

可在命令行界面中查看结果。



成功！使用Java编写的自定义规则完成！

使用XPath表达式编写该规则

PMD是支持XPath引擎的，这条“While循环必须使用括号”也可以使用XPath表达式实现。

```
//WhileStatement[not(Statement/Block)]
```

意思是匹配查找整个抽象语法树中 `WhileStatement` 节点下不存在 `Statement/Block` 这种结构的情况。

XPath表达式完成后，不需要写java代码，只写一个xml规则文件就行了。

```
<?xml version="1.0"?>
<ruleset name="My XPathRule rules"
  xmlns="http://pmd.sourceforge.net/ruleset/2.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://pmd.sourceforge.net/ruleset/2.0.0 http://pmd.sourceforge.net/ruleset_2_0_0.xsd">
  <rule name="WhileLoopsMustUseBracesRule"
    language="java"
    message="Avoid using 'while' statements without curly braces"
    class="net.sourceforge.pmd.lang.rule.XPathRule">
    <description>
      Avoid using 'while' statements without using curly braces
    </description>
    <properties>
```



```
<property name="xpath">
  <value>
<![CDATA[
//WhileStatement[not(Statement/Block)]
]]>
  </value>
</property>
</properties>
<priority>3</priority>
<example>
<![CDATA[
class Example {
  void bar() {
    while (baz)
      buz.doSomething();
  }
}
]]>
  </example>
</rule>
</ruleset>
```

运行时指向这个新编写的xml，查看结果：



成功！

小技巧：PMD自带的designer.bat工具可以快速生成一个xpath rule xml。

1. 打开designer界面工具，输入源代码，输入XPath表达式，点击Go按钮，确认右下方的结果输出正确。
2. 点击左上方Actions->Create rule XML
3. 在新的页面输入Rule name,Rule msg,Rule desc后，点击Create rule XML按钮，查看输出的结果。注意：目前版本的designer有一个小BUG，需要自己在XML中的rule标签中指定被测代码的属性language="java"

这篇文章只是教大家快速的上手自定义PMD规则，下一篇文章将重点分析复杂规则如何编写，敬请期待。

参考文献

PMD site. [How to write a PMD rule](#)

ONJava.com.[Custom PMD Rules](#)

CSDN.[静态分析工具PMD使用说明](#)

团队公众号

关注公众号，第一时间收到我们推送的新文章~



8 个赞



转载文章时务必注明原作者及原始链接，并注明「发表于 TesterHome」，并不得对作品进行修改。

共收到 12 条回复

时间

点赞



陈恒捷 #1 · May 17, 2016



正文不要出现公众号的二维码吧



思寒_seveniruby #2 · May 17, 2016



#1楼 @chenhengjie123 入驻企业专区的正规公司可以带这个. 他应该是360QA中心的同学, 这篇文章写的不错. PMD的规则定义是个不错的领域. 有助于解决一些难搞的空指针问题. 我前阵子也在研究这个工具. 还在学习如何使用自定义规则



恒温 #3 · May 17, 2016



#2楼 @seveniruby 关于文章中是否带有宣传性质, 比如公众号之类的, 麻烦 @seveniruby 定义下规则, 发下申明吧。



丁老九 #4 · May 17, 2016 作者



#2楼 @seveniruby 大神好, 这里有一些误会需要解释一下。

1. 我是属于360Qtest团队的, 跟360QA中心目前是相互独立的。
2. 希望能在公司专栏处新建一个360Qtest的位置, 就可以把之前俊哥的火线帖子也放在该专

栏下。

3. PMD的原理是基于模式匹配，目前还不擅长解决难搞的空指针问题，findbugs比较擅长扫描空指针问题，我会逐渐的更新静态扫描系列文章，以后都会提到的。



丁老九 #5 · May 17, 2016 作者



#3楼 @lihuazhang 嗯嗯，如果有明确的版规说明不让发公众号二维码，我会及时删除。



思寒_seveniruby #6 · May 17, 2016



#4楼 @oggbboy 因为你们都同属于360公司. 那你们就统一发到360公司QA专区如何?
我在那个专栏里面做了介绍

奇虎360公司多个QA团队的文章专区.

包括并不限于360QA中心和360Qtest等技术卓越的测试团队.

TesterHome

社区

招聘

公开课

Wiki

臭虫广播站

360公司QA专区 共有 1 个讨论主题

奇虎360公司多个QA团队的文章专区.

包括并不限于360QA中心和360Qtest等技术卓越的测试团队.



静态代码扫描——PMD 自定义规则实践 (一) ★

360QA中心 · oggbboy 于 4 小时前发布 · 最后由 oggbboy 于 1 小时前回复



丁老九 #7 · May 17, 2016 作者



#6楼 @seveniruby 嗯嗯，好的，非常感谢🙏



simple #8 · May 18, 2016



文章范围缩小到一定的领域，面向的读者肯定是目前正在从事这方面工作的人，不做静态代码分析工作的人可能没有耐心读完。不过后面我们还是会持续的写类似的文章，目的是跟大家一起探讨基于代码层面的检测，让测试工作upstream，目前PMD规则自定义，业内都是写个如何入门，比较浅，但是PMD是个好东西，套用monkey大神的话，金箍棒在龙王那里就是个定海神针，在孙悟空那里就是能捅破天的东西。



大风 #9 · June 17, 2016



SonarQube的Java自定义规则也是用AST来做的，并且已逐渐不再支持XPath了，看来还是PMD更开放。

 丁老九 在 [Topic was deleted] 中提及了此贴 08 Jul 16:24




丁老九 #11 · April 25, 2017 作者




根据用户反馈，修改优化该文档不易理解的部分。

 丁老九 在 静态代码扫描 (二)——PMD 自定义规则实践 中提及了此贴 05 May 15:58

 丁老九 在 静态代码扫描 (三)——FindBugs 自定义规则入门 中提及了此贴 08 May 11:53

 丁老九 在 静态代码扫描 (四)——Java 资源关闭研究 中提及了此贴 08 May 11:53

 丁老九 在 静态代码扫描 (五)——Java 资源关闭的特殊场景 中提及了此贴 15 May 11:31



summe #16 · June 13, 2017



请问，designer.bat 执行后没有界面出现，这是为什么？



丁老九 #17 · June 20, 2017 作者



对  summe #16 回复

电脑上有JAVA环境么？

需要

Sign In

后方可回复, 如果你还没有账号请点击这里

Sign Up

。

相关话题

[腾讯 TMQ] 管中窥豹：结合 NewApi 实践来了解 Lint 代码扫描

静态代码扫描 (二)——PMD 自定义规则实践

静态代码扫描 (五)——Java 资源关闭的特殊场景

Sonar Java 规则插件开发 (基于阿里开发手册)

python 静态代码扫描工具 sonarqube 入门 (自定义扫描规则)

作者



oggboy (丁老九)

第 1064 位Users / 2014-06-12

VIP

“这家伙很懒，什么个性签名都没有留下。”



[关于](#) / [活跃用户](#) / [中国移动互联网测试技术大会](#) / [反馈](#) / [Github](#) / [API](#) / [帮助推广](#)

TesterHome 移动测试社区，由众多移动测试工作者维护，致力于推进国内测试技术。Inspired by RubyChina

友情链接 [WeTest](#)[腾讯质量开放平台](#) / [InfoQ](#) / [测试之道](#) / [测试窝](#) / [百度测试吧](#) / [IT大咖说](#)

[简体中文](#) / [正體中文](#) / [English](#)

