

Developer WorkBench

PMD - Code Review Tool

# Lesson Objectives

- In this lesson, you will learn:
  - What is PMD
  - PMD Rulesets
  - PMD usage
  - PMD in Action

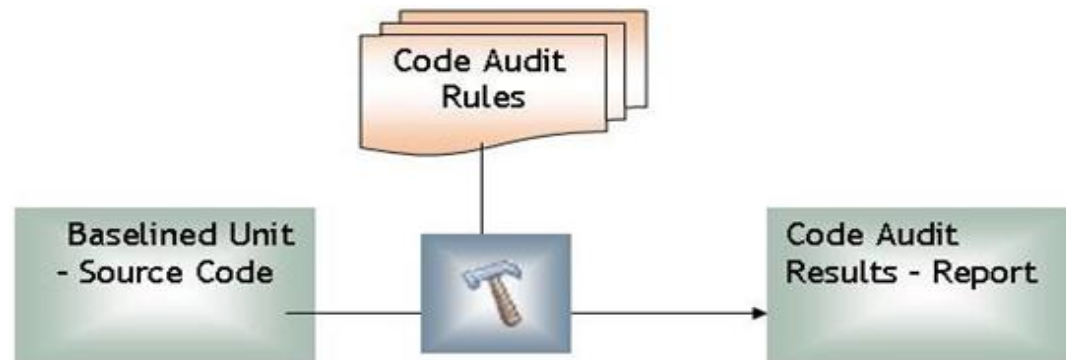


# What is PMD?

- Automated code review utility for java
- Static code analyzer
- Automatically detects a wide range of potential defects and unsafe or non-optimized code
- Focuses on preemptive defect detection

# What is PMD? (Contd...)

- Possible bugs: Empty try/catch/finally/switch statements
- Dead code: Unused local variables, parameters and private methods
- Suboptimal code: Wasteful String/String Buffer usage
- Overcomplicated expressions - Unnecessary if statements, for loops that could be while loops
- Duplicate code: Copied/pasted code means copied/pasted bugs



# PMD Rule Sets

- Rules are grouped together and form a Rule Set  
E.g. Basic Rules, Code Size Rules, Coupling Rules, Design Rules, JavaBeans Rules, Junit Rules, Migration Rules, Naming Rules, Optimization Rules, Security Code Guidelines, Unused Code Rules, etc.
- For more information visit:  
<http://pmd.sourceforge.net/rules/index.html>

# PMD Usage

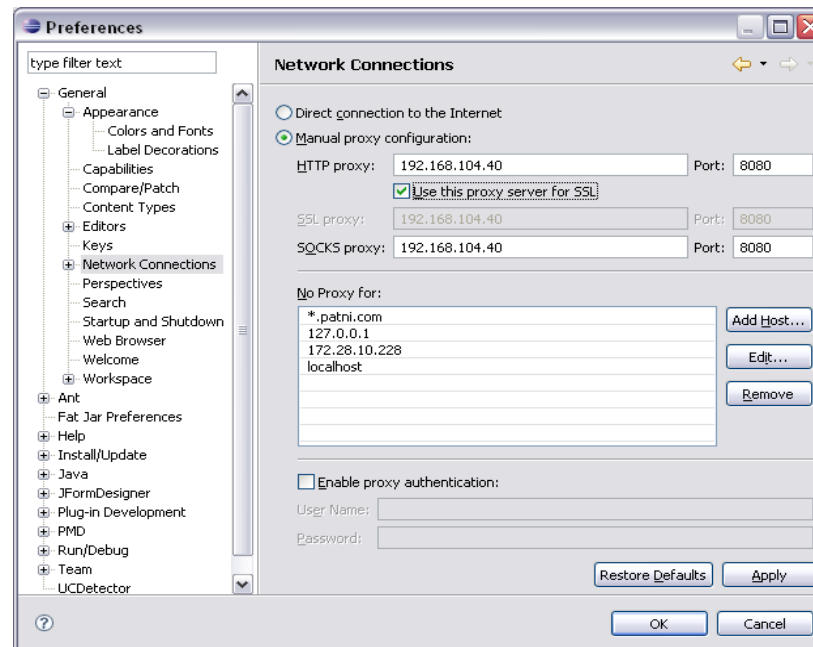
- Installation:
  - Windows and UNIX
- Prerequisites:
  - JDK 1.4 or higher
  - WinZip for Windows or "zip" utility InfoZip for UNIX
- Download the latest binary distribution - i.e., pmd-bin-x.xx.zip from <http://pmd.sourceforge.net/index.html>. Unzip it into any directory
  - \*Note: Please contact ITIMD Helpdesk to download the plug-in

# Integration with IDE's

- PMD can be integrated with following IDE's:
  - Eclipse
  - Jdeveloper
  - Jedit
  - Jbuilder
  - BlueJ
  - CodeGuide
  - NetBeans/Sun Java Studio Enterprise/Creator
  - IntelliJ IDEA
  - TextPad
  - Maven
  - Ant
  - Gel
  - Jcreator
  - Emacs

# Integration with Eclipse IDE

- Configure Eclipse:
  - Setup the network connection in Eclipse preferences. Go to Windows -> Preferences -> General -> Network Connection
  - Set your preferences as per your Internet Browser Connections Settings



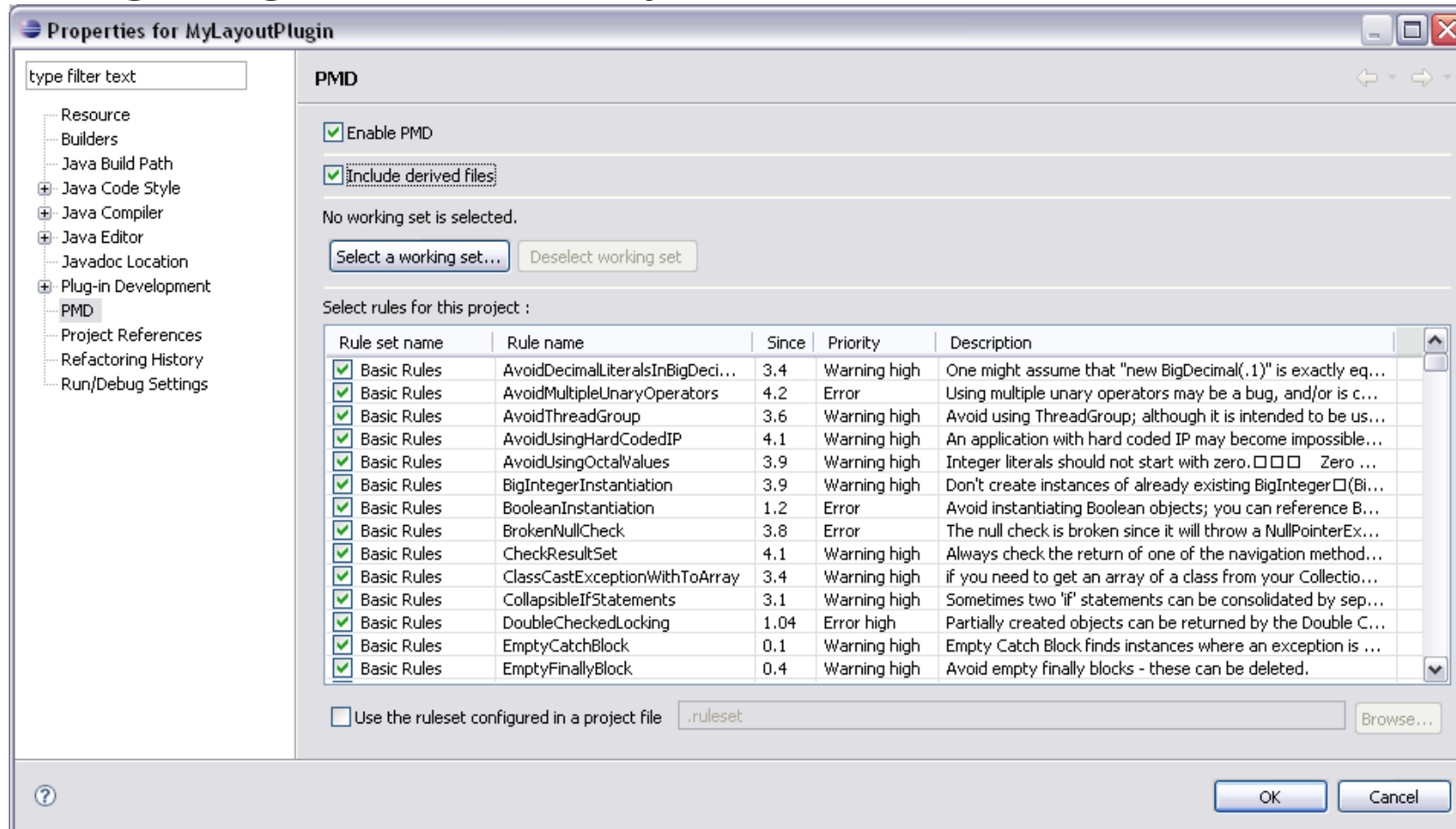


# Integration with Eclipse IDE (Contd...)

- To install PMD plug-in,
  - Start Eclipse and open a project
  - Select "Help"->"Software Updates"->"Find and Install"
  - Click "Next", then click "New remote site"
  - Enter "PMD" into the Name field and "<http://pmd.sf.net/eclipse>" into the URL field ( this downloads the plug in from the site)
  - Click through the rest of the dialog boxes to install the plug-in
- Alternatively, you can download the latest zip file and follow the above procedures except for using "New local site" and browsing to the downloaded zip file.

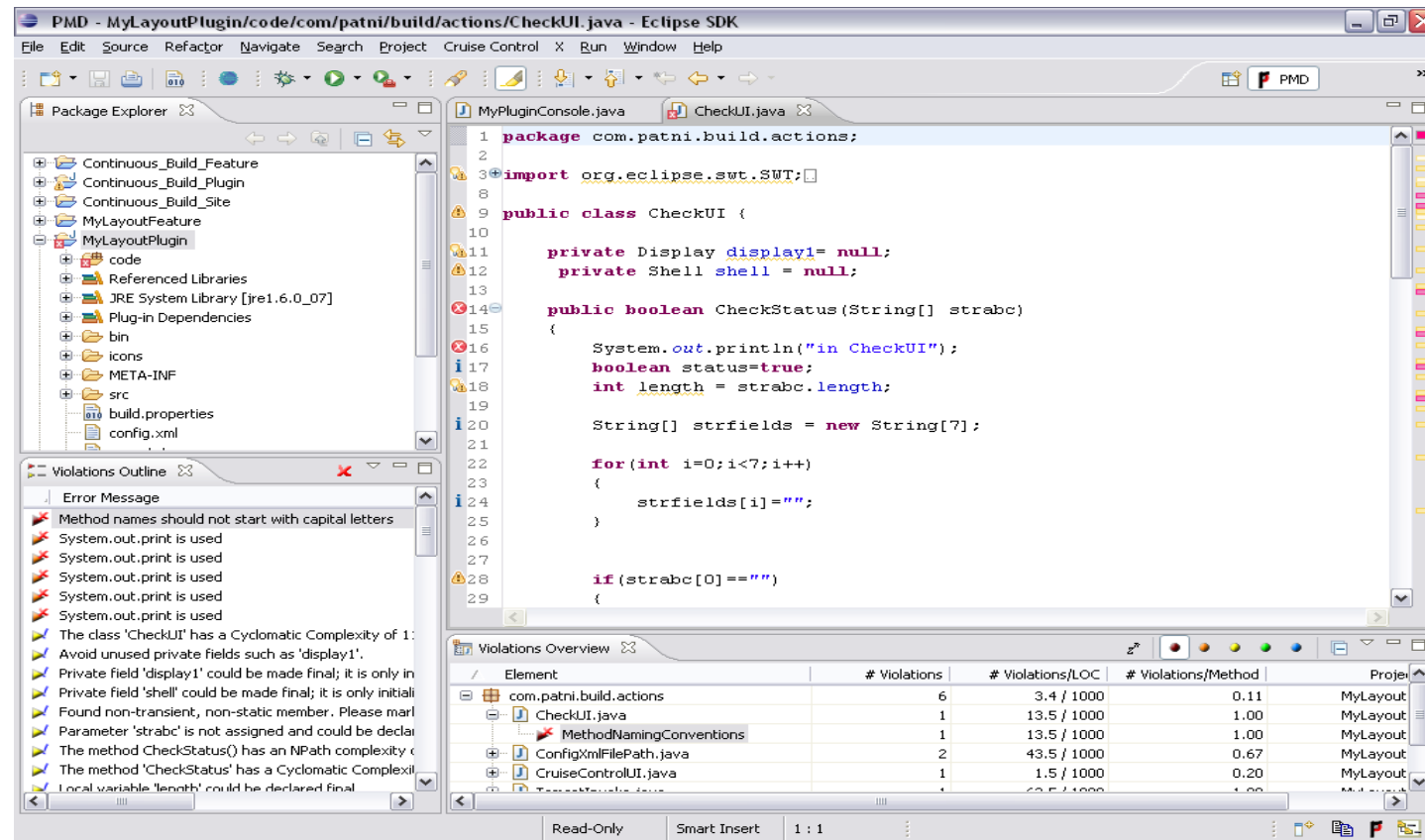
# Integration with Eclipse IDE (Contd...)

- Configuring PMD in a Project:



# Integration with Eclipse IDE (Contd...)

- To run PMD with Eclipse – Right click on the project, select PMD – Check Code with PMD

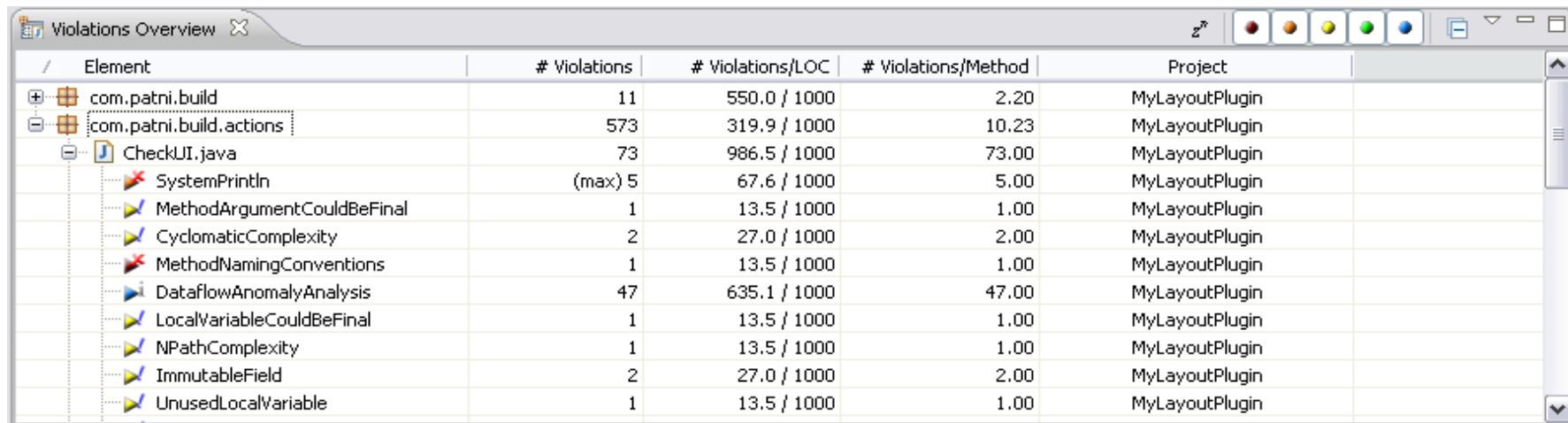


# PMD In Action

- PMD is quite flexible. You can use it in two basic ways:
  - Wait for PMD to automatically analyze a file each time it is saved or added to the project
  - Manually execute it against selected files, folders, or projects

# PMD In Action (Contd...)

- PMD violations have the following five levels of severity:
  - Error (high)
  - Error
  - Warning (high)
  - Warning
  - Information

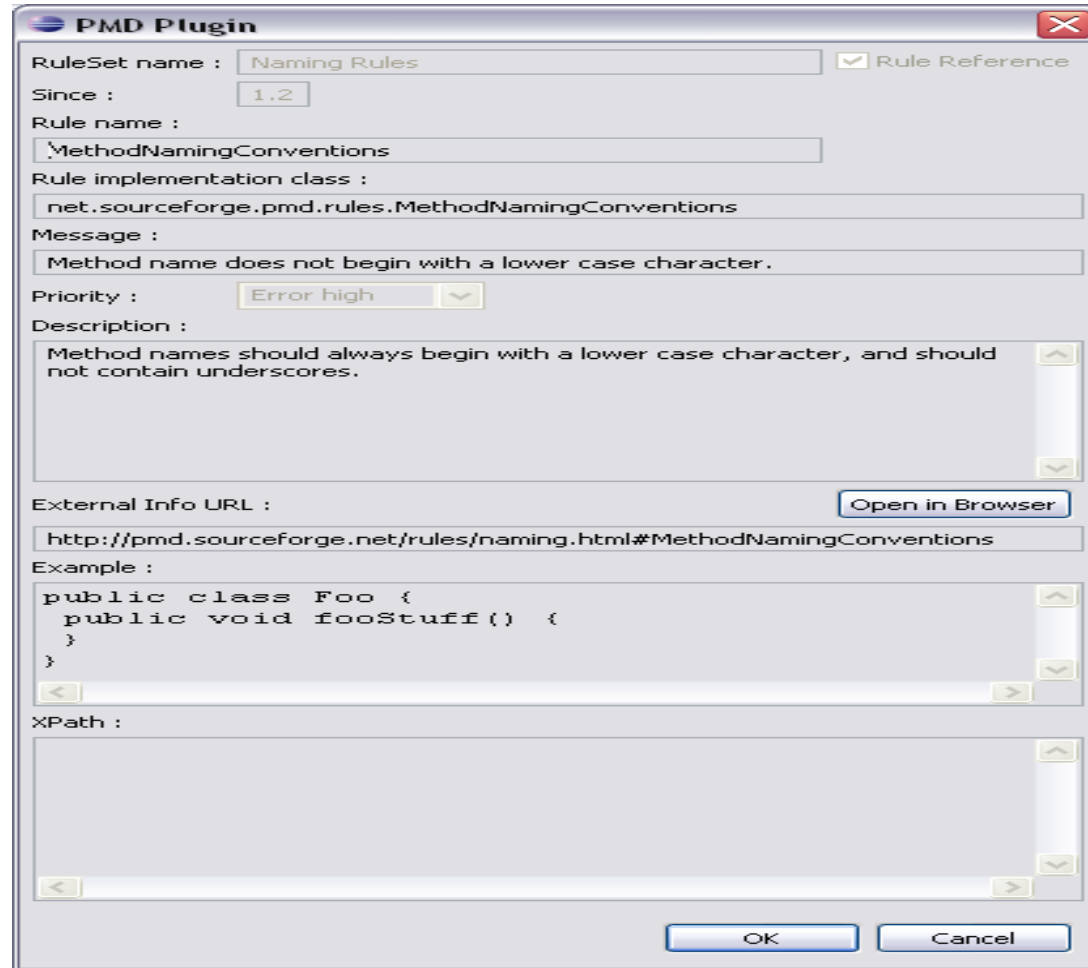


The screenshot shows the 'Violations Overview' window in PMD. It displays a table with columns: Element, # Violations, # Violations/LOC, # Violations/Method, and Project. The table lists violations for the project 'MyLayoutPlugin' across various elements and rules.

Element	# Violations	# Violations/LOC	# Violations/Method	Project
com.patni.build	11	550.0 / 1000	2.20	MyLayoutPlugin
com.patni.build.actions	573	319.9 / 1000	10.23	MyLayoutPlugin
CheckUI.java	73	986.5 / 1000	73.00	MyLayoutPlugin
SystemPrintln	(max) 5	67.6 / 1000	5.00	MyLayoutPlugin
MethodArgumentCouldBeFinal	1	13.5 / 1000	1.00	MyLayoutPlugin
CyclomaticComplexity	2	27.0 / 1000	2.00	MyLayoutPlugin
MethodNamingConventions	1	13.5 / 1000	1.00	MyLayoutPlugin
DataflowAnomalyAnalysis	47	635.1 / 1000	47.00	MyLayoutPlugin
LocalVariableCouldBeFinal	1	13.5 / 1000	1.00	MyLayoutPlugin
NPathComplexity	1	13.5 / 1000	1.00	MyLayoutPlugin
ImmutableField	2	27.0 / 1000	2.00	MyLayoutPlugin
UnusedLocalVariable	1	13.5 / 1000	1.00	MyLayoutPlugin

# PMD In Action (Contd...)

- Details of the Rule



The screenshot shows the 'PMD Plugin' configuration window. The 'RuleSet name' is 'Naming Rules', and 'Rule Reference' is checked. The 'Since' version is '1.2'. The 'Rule name' is 'MethodNamingConventions', and the 'Rule implementation class' is 'net.sourceforge.pmd.rules.MethodNamingConventions'. The 'Message' is 'Method name does not begin with a lower case character.'. The 'Priority' is set to 'Error high'. The 'Description' text area contains: 'Method names should always begin with a lower case character, and should not contain underscores.'. The 'External Info URL' is 'http://pmd.sourceforge.net/rules/naming.html#MethodNamingConventions', with an 'Open in Browser' button. The 'Example' text area shows a code snippet: 

```
public class Foo {  
    public void fooStuff() {  
    }  
}
```

. The 'XPath' field is empty. 'OK' and 'Cancel' buttons are at the bottom.

**PMD Plugin**

RuleSet name :  ☒ Rule Reference

Since :

Rule name :

Rule implementation class :

Message :

Priority :  ▼

Description : 

Method names should always begin with a lower case character, and should not contain underscores.

External Info URL :

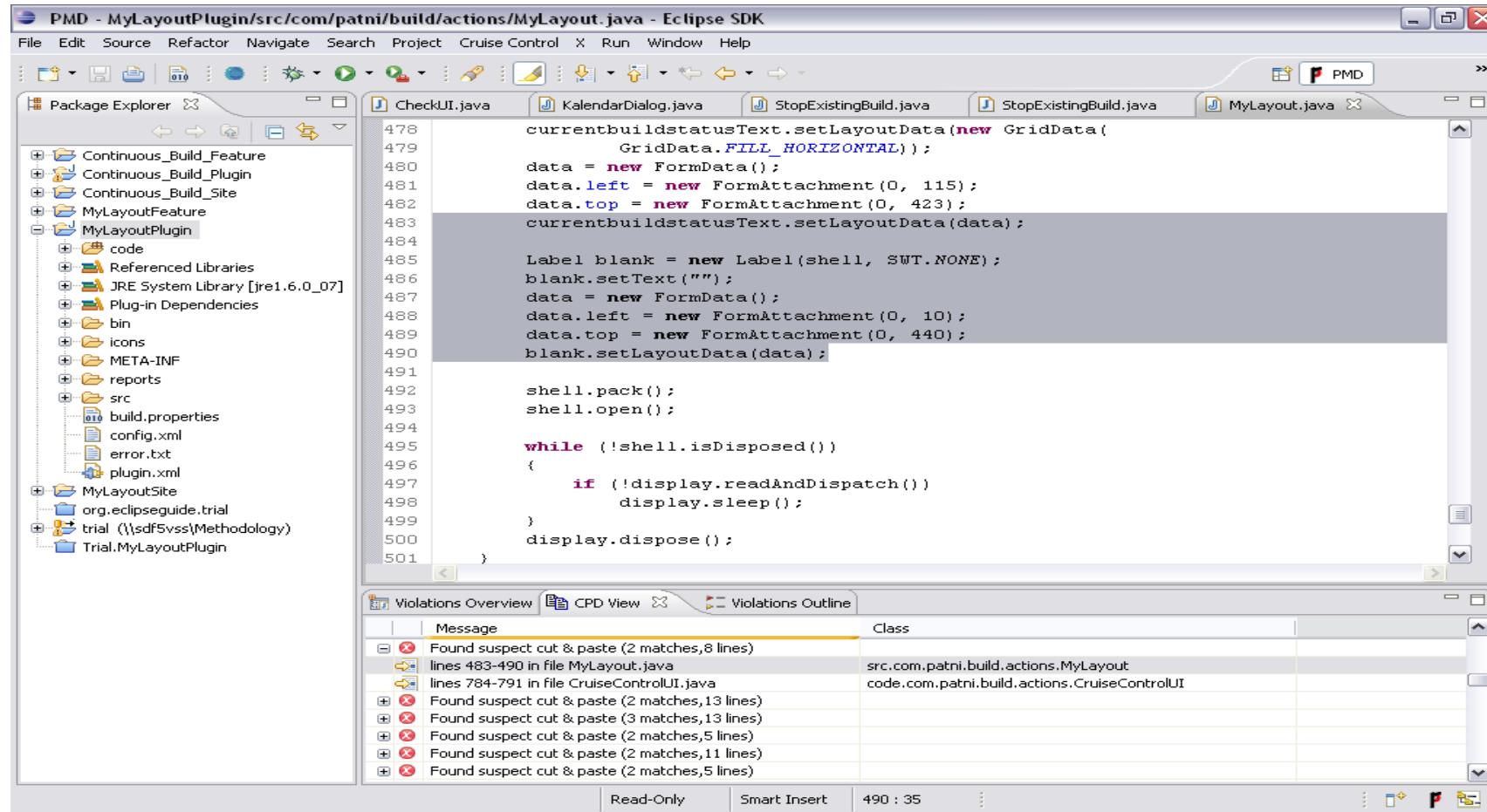
Example : 

```
public class Foo {  
    public void fooStuff() {  
    }  
}
```

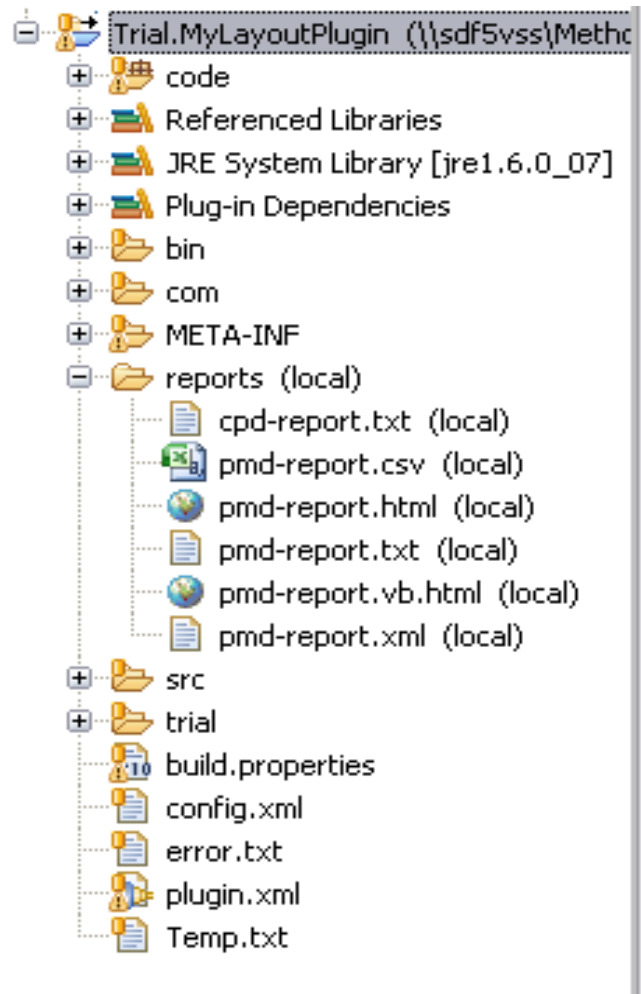
XPath :

# PMD In Action (Contd...)

- Detecting Cut Pasted Code



# PMD In Action (Contd...)



• Reporting Feature: PMD generates reports in following formats:

- .txt
- .csv
- .html
- .vb.html
- .xml



# PMD In Action (Contd...)

- HTML Reporting Format:

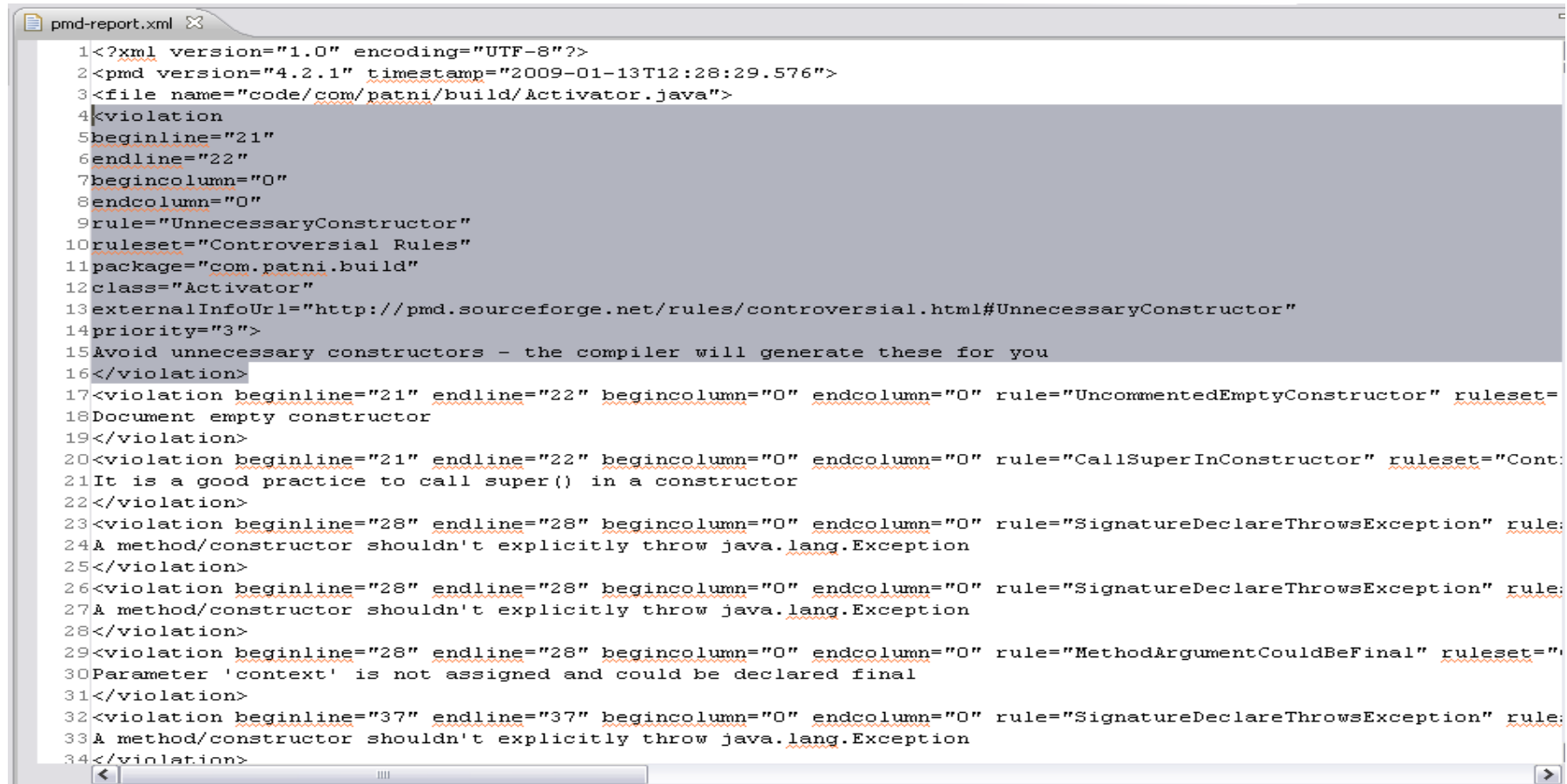
## PMD report

### Problems found

#	File	Line	Problem
1	code/com/patni/build/Activator.java	21	<a href="#">Avoid unnecessary constructors - the compiler will generate these for you</a>
2	code/com/patni/build/Activator.java	21	<a href="#">Document empty constructor</a>
3	code/com/patni/build/Activator.java	21	<a href="#">It is a good practice to call super() in a constructor</a>
4	code/com/patni/build/Activator.java	28	<a href="#">A method/constructor shouldn't explicitly throw java.lang.Exception</a>
5	code/com/patni/build/Activator.java	28	<a href="#">A method/constructor shouldn't explicitly throw java.lang.Exception</a>
6	code/com/patni/build/Activator.java	28	<a href="#">Parameter 'context' is not assigned and could be declared final</a>
7	code/com/patni/build/Activator.java	37	<a href="#">A method/constructor shouldn't explicitly throw java.lang.Exception</a>
8	code/com/patni/build/Activator.java	37	<a href="#">A method/constructor shouldn't explicitly throw java.lang.Exception</a>
9	code/com/patni/build/Activator.java	37	<a href="#">Parameter 'context' is not assigned and could be declared final</a>
10	code/com/patni/build/Activator.java	38	<a href="#">Assigning an Object to null is a code smell. Consider refactoring.</a>
11	code/com/patni/build/Activator.java	58	<a href="#">Parameter 'path' is not assigned and could be declared final</a>
12	code/com/patni/build/actions/CheckUI1.java	7	<a href="#">The class 'CheckUI1' has a Cyclomatic Complexity of 11 (Highest = 10).</a>
13	code/com/patni/build/actions/CheckUI1.java	9	<a href="#">Avoid unused private fields such as 'display1'.</a>
14	code/com/patni/build/actions/CheckUI1.java	9	<a href="#">This final field could be made static</a>
15	code/com/patni/build/actions/CheckUI1.java	10	<a href="#">This final field could be made static</a>
16	code/com/patni/build/actions/CheckUI1.java	12	<a href="#">The method 'checkStatus' has a Cyclomatic Complexity of 10.</a>
17	code/com/patni/build/actions/CheckUI1.java	12	<a href="#">The method checkStatus() has an NPath complexity of 512</a>

# PMD In Action (Contd...)

- XML Reporting Format:



```
1<?xml version="1.0" encoding="UTF-8"?>
2<pmd version="4.2.1" timestamp="2009-01-13T12:28:29.576">
3<file name="code/com/patni/build/Activator.java">
4<violation
5beginline="21"
6endline="22"
7begincolumn="0"
8endcolumn="0"
9rule="UnnecessaryConstructor"
10ruleset="Controversial Rules"
11package="com.patni.build"
12class="Activator"
13externalInfoUrl="http://pmd.sourceforge.net/rules/controversial.html#UnnecessaryConstructor"
14priority="3">
15Avoid unnecessary constructors - the compiler will generate these for you
16</violation>
17<violation beginline="21" endline="22" begincolumn="0" endcolumn="0" rule="UncommentedEmptyConstructor" ruleset="
18Document empty constructor
19</violation>
20<violation beginline="21" endline="22" begincolumn="0" endcolumn="0" rule="CallSuperInConstructor" ruleset="Cont:
21It is a good practice to call super() in a constructor
22</violation>
23<violation beginline="28" endline="28" begincolumn="0" endcolumn="0" rule="SignatureDeclareThrowsException" rule:
24A method/constructor shouldn't explicitly throw java.lang.Exception
25</violation>
26<violation beginline="28" endline="28" begincolumn="0" endcolumn="0" rule="SignatureDeclareThrowsException" rule:
27A method/constructor shouldn't explicitly throw java.lang.Exception
28</violation>
29<violation beginline="28" endline="28" begincolumn="0" endcolumn="0" rule="MethodArgumentCouldBeFinal" ruleset="
30Parameter 'context' is not assigned and could be declared final
31</violation>
32<violation beginline="37" endline="37" begincolumn="0" endcolumn="0" rule="SignatureDeclareThrowsException" rule:
33A method/constructor shouldn't explicitly throw java.lang.Exception
34</violation>
```

# Customizing PMD

- It is possible to create your own rules and rule sets
- For additional information and detailed description on customizing PMD visit,
  - [http://www.jacoozi.com/index.php?option=com\\_content&task=view&id=121&Itemid=134](http://www.jacoozi.com/index.php?option=com_content&task=view&id=121&Itemid=134)
  - <http://pmd.sourceforge.net/>

# Lab: 3

- 2.1: Installing PMD
- 2.2 Using PMD with Eclipse



# Summary

- Learned the usage of PMD
- Understood how to Integrate PMD with Eclipse IDE



# Review Question

- Question 1: PMD is a
  - Review tool
  - Testing Tool
  - Design Tool
- Question 2: PMD violations has \_\_\_\_\_ level of severity
  - Error (high)
  - Fatal
  - Warning (high)
  - Information

