

Checkstyle

A static analysis tool for Java

ASU CST216 Team 22



Contents

1. Originator	2
2. Purpose.....	2
3. Programming Language.....	2
4. Effective Use - What Phase?	2
5. Tool Usage & Installation.....	3
Installation.....	3
6. Configuration.....	4
7. How to use.....	6
8. Checkstyle Strengths	10
9. Checkstyle Limitations	10
Appendix A	11

Category: Test Tool, Unit Test, Checkstyle

1. Originator

The Checkstyle tool was developed by Oliver Burn and released to public domain for over two decades under GNU Public License (GPL.) The tool is downloadable through Eclipse marketplace (<https://marketplace.eclipse.org/content/checkstyle-plugin>) free of charge.

2. Purpose

The Eclipse Checkstyle Plugin (eclipse-cs) integrates the well-known source code analyzer Checkstyle into the Eclipse Integrated Development Environment (IDE.)

Checkstyle is a development tool to help Java programmers ensure that their Java code adheres to a set of coding standards. Checkstyle does this by inspecting their Java source code and pointing out items that deviate from a defined set of coding rules.

Checkstyle is easy to install, configure and use. The tool is very visual and can be used during development process to make corrections on the go. With the Eclipse Checkstyle Plugin the programmers' code is constantly inspected for problems.

Within the Eclipse workbench they are notified of problems via the Eclipse Problems View and source code annotations just as they would see with compiler errors or warnings.

A comfortable Checkstyle configuration editor helps Java programmers create and maintain their audit rule configurations by using filters and file sets which enable Java programmers to define which files get checked and which do not.

The primary use of this tool is to enforce coding rules defined by software development teams.

See more at: <https://marketplace.eclipse.org/content/checkstyle-plugin#sthash.z57MUmcE.dpuf>.

3. Programming Language

Currently Checkstyle only supports the Java programming language.

4. Effective Use - What Phase?

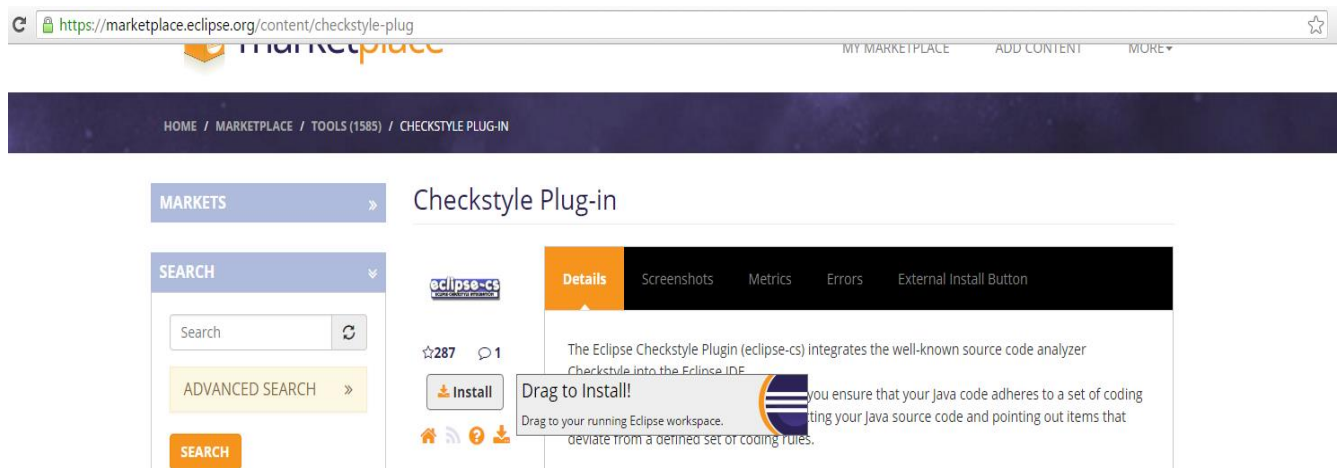
The Checkstyle tool is most effective during Unit Testing phase where it is a part of static analysis of

the code base.

5. Tool Usage & Installation

Installation

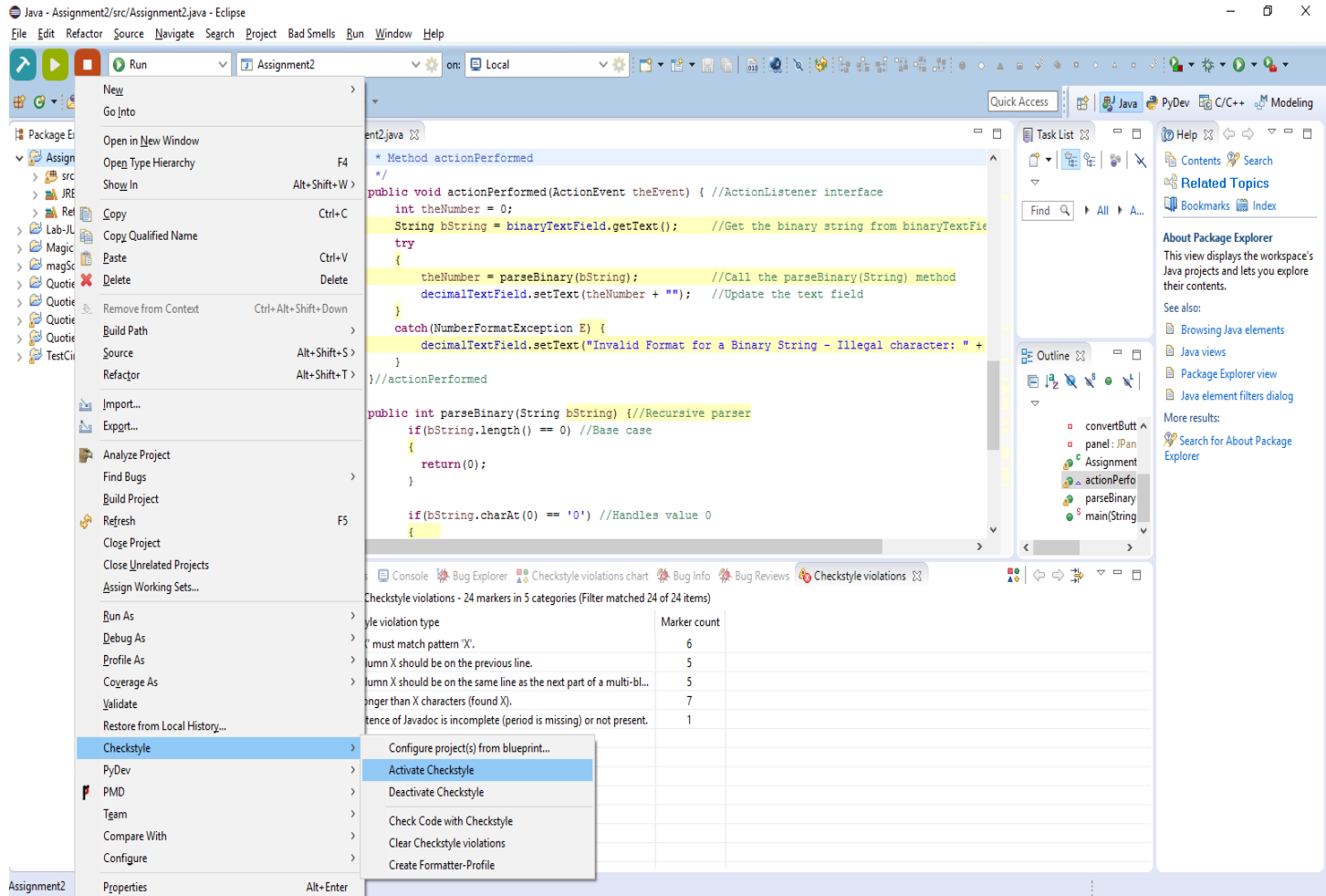
- Invoke the Eclipse Integrated Development and wait for it to completely open.
- Use any browser, navigate to <https://marketplace.eclipse.org/content/checkstyle-plugin> site.
- Mouse over to the [Install] button on the web site, single left-click and hold down the left mouse button to drag the [Install] button onto the Eclipse IDE. See picture below:



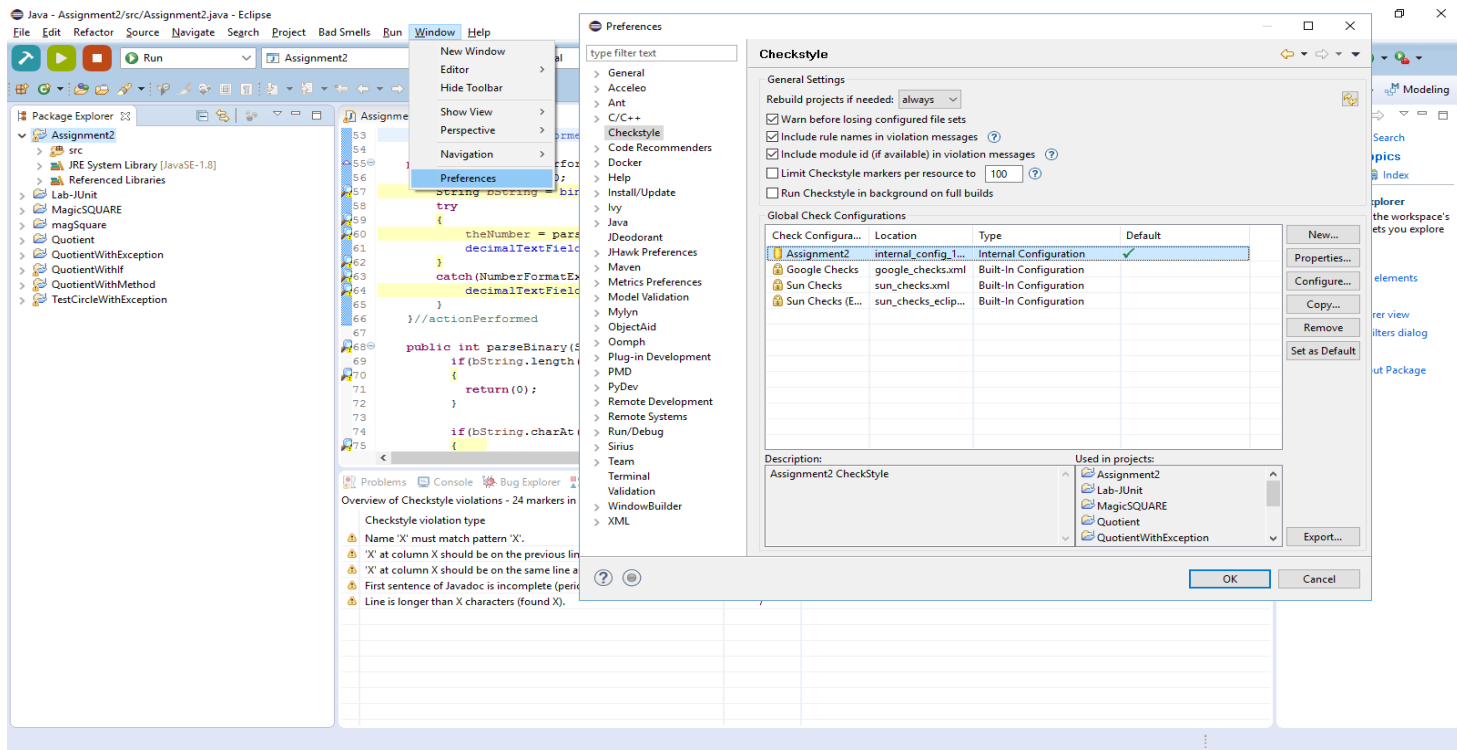
- Follow Eclipse IDE screen prompts to complete the installation. The Eclipse IDE will be restarted after the installation.

6. Configuration

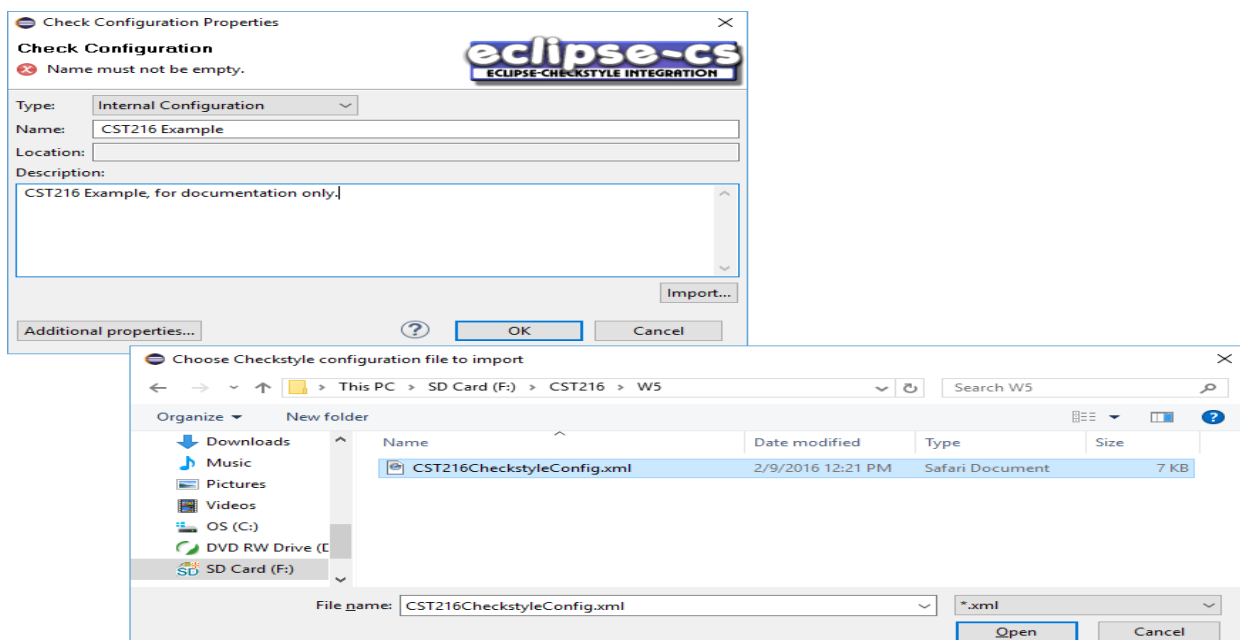
The Checkstyle tool works in concert with an external rule file written in Extensible Markup Language (XML) format which must be activated in Eclipse IDE. See picture below.



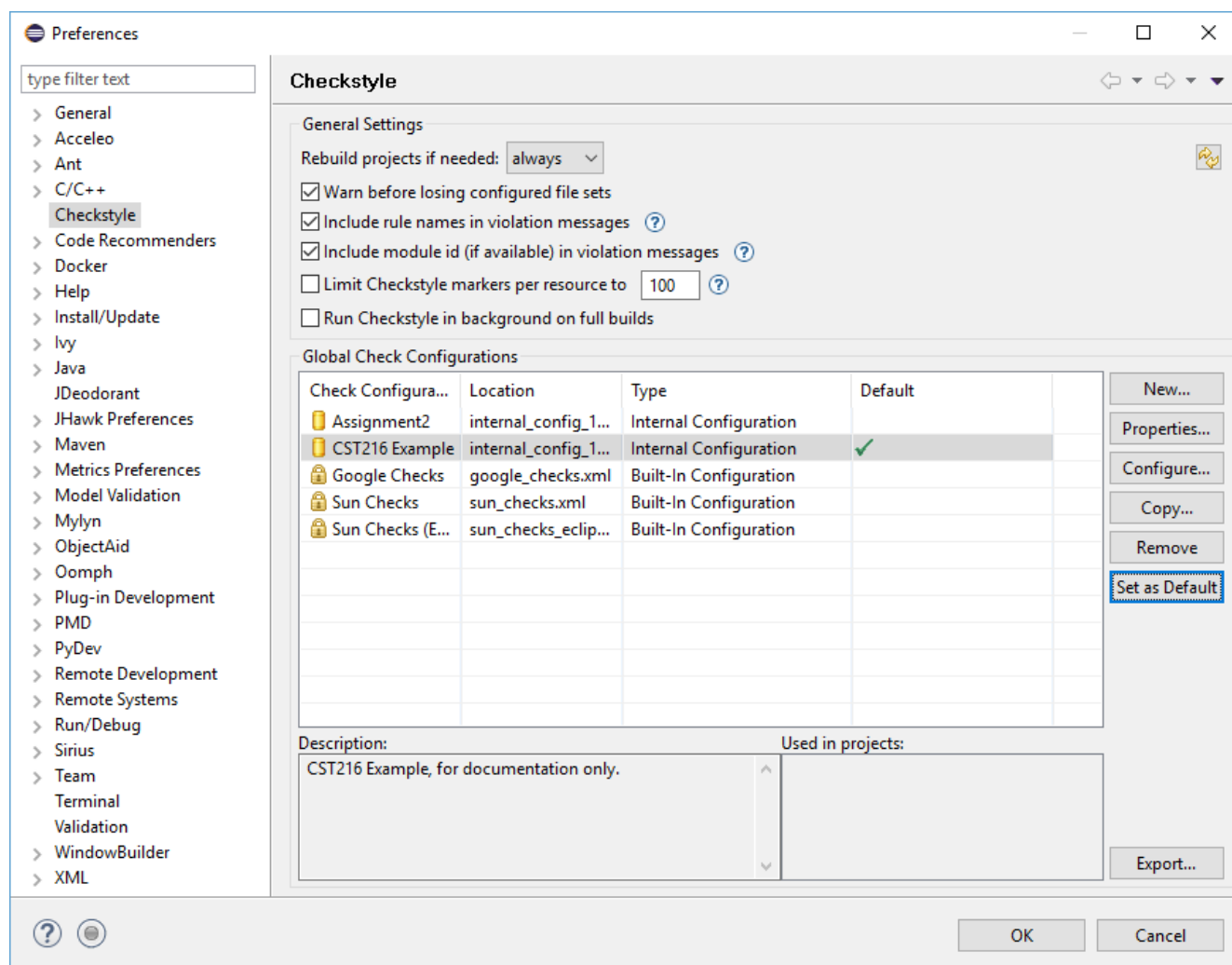
After activation, Checkstyle tool must be configured within Eclipse IDE. The picture below depicts Checkstyle already configured for use with quite a few projects.



To have a new configuration, click on the [New] button of this window, another window pops up. Leave the type as [Internal Configuration] at this point but enter a name in the [Name] box. Then click on [Import] button and navigate to where the rule (.xml) file resides and bring it in (assuming a natural GUI driven user without much difficulty with the Operating System.) See picture below.



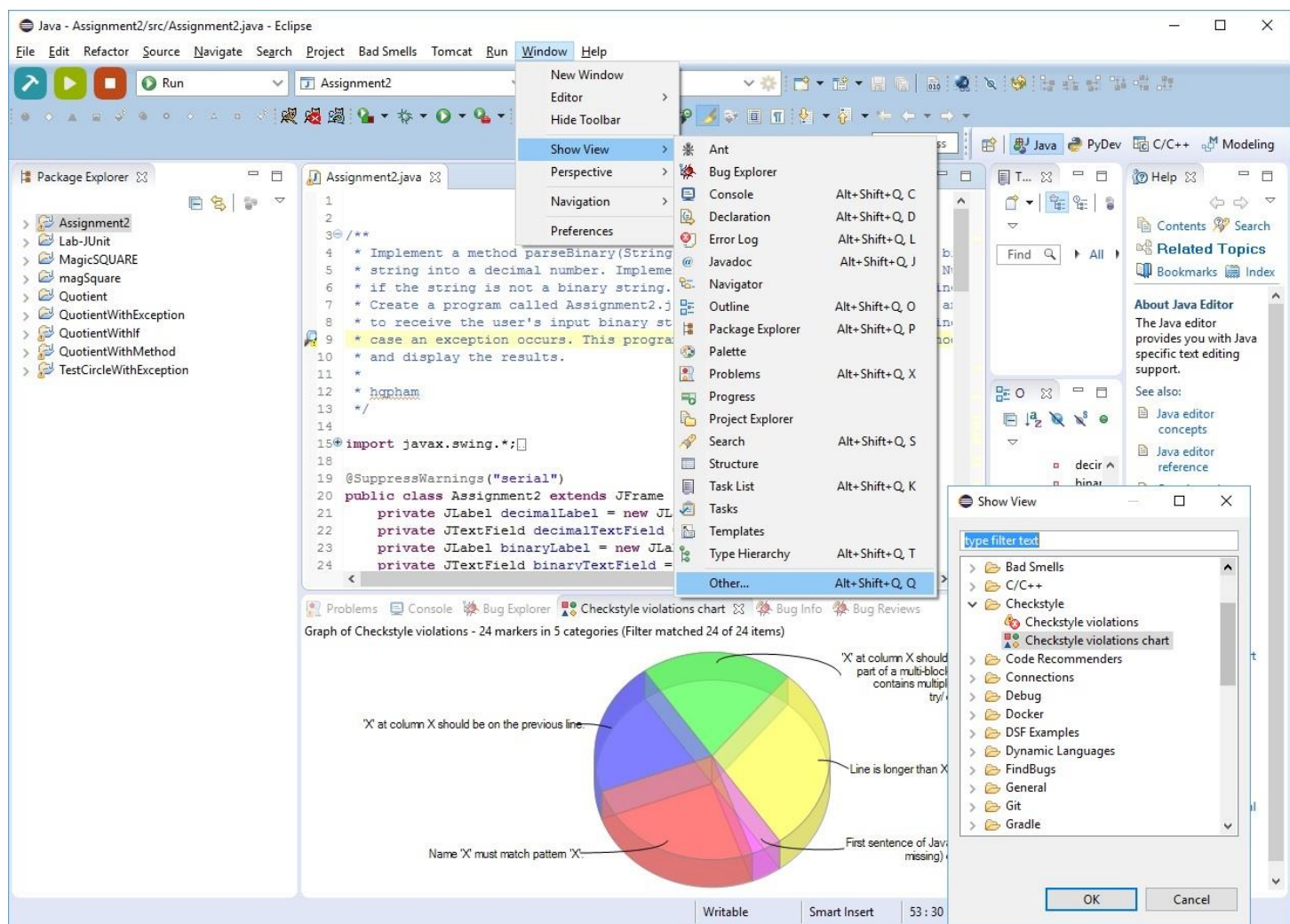
Now the new rule file has been brought in to Eclipse IDE. The user can make it the default rule file if so desired. See picture below.



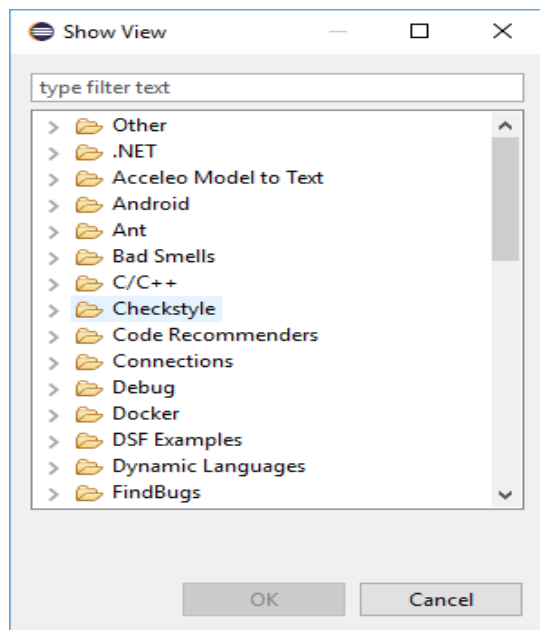
Click [OK] on this window to complete the configuration process.

7. How to use

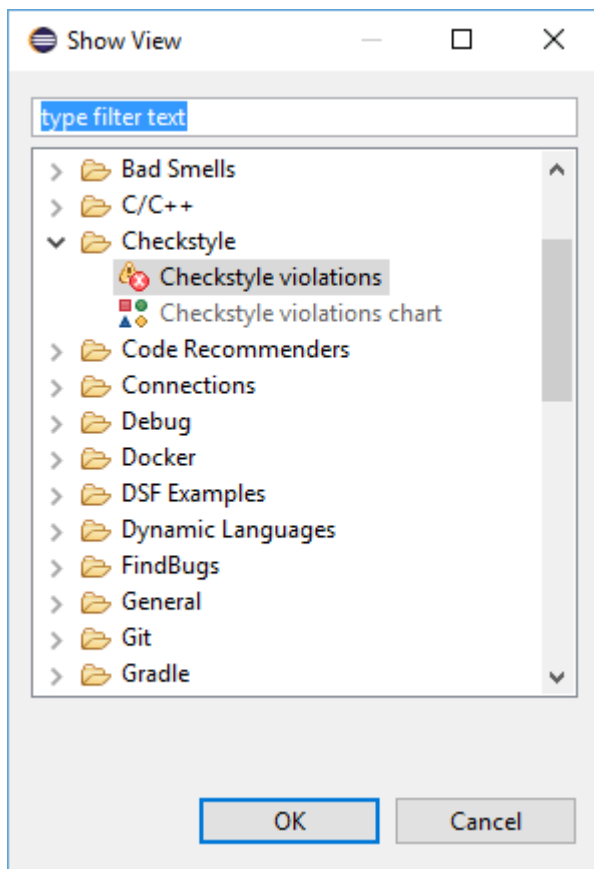
Once configured, Checkstyle takes action right after the project got compiled. To view the output, select [Window] and scroll down to [Show View] and then scroll down to [Other...]. See picture below.



Another window pops up. As shown below.



Click on [Checkstyle] and two forms of output are available there: Tablature form, and Pie-chart form. See picture below.



Select desired view and then click [OK] to continue. If the [Checkstyle violations] option was selected, then the output is displayed as shown below:

Java - Assignment2/src/Assignment2.java - Eclipse

File Edit Refactor Source Navigate Search Project Bad Smells Run Window Help

Run Assignment2 on: Local

Package Explorer

- Assignment2
 - src
 - JRE System Library [JavaSE-1.8]
 - Referenced Libraries
 - Lab-JUnit
 - MagicSquare
 - magSquare
 - Quotient
 - QuotientWithException
 - QuotientWithIf
 - QuotientWithMethod
 - TestCircleWithException

Assignment2.java

```

53  * Method actionPerformed
54  */
55  public void actionPerformed(ActionEvent theEvent) { //ActionListener interface
56      int theNumber = 0;
57      String bString = binaryTextField.getText(); //Get the binary string from binaryTextFie
58      try
59      {
60          theNumber = parseBinary(bString); //Call the parseBinary(String) method
61          decimalTextField.setText(theNumber + ""); //Update the text field
62      }
63      catch (NumberFormatException E) {
64          decimalTextField.setText("Invalid Format for a Binary String - Illegal character: " +
65      }
66  } //actionPerformed
67
68  public int parseBinary(String bString) { //Recursive parser
69      if (bString.length() == 0) //Base case
70      {
71          return (0);
72      }
73
74      if (bString.charAt(0) == '0') //Handles value 0
75  {

```

Task List

Find All A...

Outline

- convertButt
- panel: JPan
- Assignment
- actionPerfo
- parseBinary
- main(String

Help

Contents Search

Related Topics

Bookmarks Index

About Package Explorer

This view displays the workspace's Java projects and lets you explore their contents.

See also:

- Browsing Java elements
- Java views
- Package Explorer view
- Java element filters dialog

More results:

Search for About Package Explorer

Problems Console Bug Explorer Checkstyle violations chart Bug Info Bug Reviews Checkstyle violations

Overview of Checkstyle violations - 24 markers in 5 categories (Filter matched 24 of 24 items)

Checkstyle violation type	Marker count
Name 'X' must match pattern 'X'.	6
'X' at column X should be on the previous line.	5
'X' at column X should be on the same line as the next part of a multi-bl...	5
Line is longer than X characters (found X).	7
First sentence of Javadoc is incomplete (period is missing) or not present.	1

If the [Checkstyle violations chart] option was selected, then the output is displayed as shown below,

Java - Assignment2/src/Assignment2.java - Eclipse

File Edit Refactor Source Navigate Search Project Bad Smells Run Window Help

Run Assignment2 on: Local

Package Explorer

- Assignment2
 - src
 - JRE System Library [JavaSE-1.8]
 - Referenced Libraries
 - Lab-JUnit
 - MagicSquare
 - magSquare
 - Quotient
 - QuotientWithException
 - QuotientWithIf
 - QuotientWithMethod
 - TestCircleWithException

Assignment2.java

```

53  * Method actionPerformed
54  */
55  public void actionPerformed(ActionEvent theEvent) { //ActionListener interface
56      int theNumber = 0;
57      String bString = binaryTextField.getText(); //Get the binary string from binaryTextFie
58      try
59      {
60          theNumber = parseBinary(bString); //Call the parseBinary(String) method
61          decimalTextField.setText(theNumber + ""); //Update the text field
62      }
63      catch (NumberFormatException E) {
64          decimalTextField.setText("Invalid Format for a Binary String - Illegal character: " +
65      }
66  } //actionPerformed
67
68  public int parseBinary(String bString) { //Recursive parser
69      if (bString.length() == 0) //Base case
70      {
71          return (0);
72      }
73
74      if (bString.charAt(0) == '0') //Handles value 0
75  {

```

Task List

Find All A...

Outline

- convertButt
- panel: JPan
- Assignment
- actionPerfo
- parseBinary
- main(String

Help

Contents Search

Related Topics

Bookmarks Index

About Checkstyle violations chart

Each workbench window contains one or more perspectives, which are made up of various views and editors.

See also:

- Workbench
- Perspectives
- Customizing the Workbench

More results:

Search for Checkstyle violations chart view

Problems Console Bug Explorer Checkstyle violations chart Bug Info Bug Reviews Checkstyle violations

Graph of Checkstyle violations - 24 markers in 5 categories (Filter matched 24 of 24 items)

'X' at column X should be on the previous line.

'X' at column X should be on the same line as the next part of a multi-block statement (one that directly contains multiple blocks: if else-if else or try/ catch/ finally).

Line is longer than X characters (found X).

Name 'X' must match pattern 'X'.

First sentence of Javadoc is incomplete (period is missing) or not present.

8. Checkstyle Strengths

The Checkstyle tool strengths are as follows:

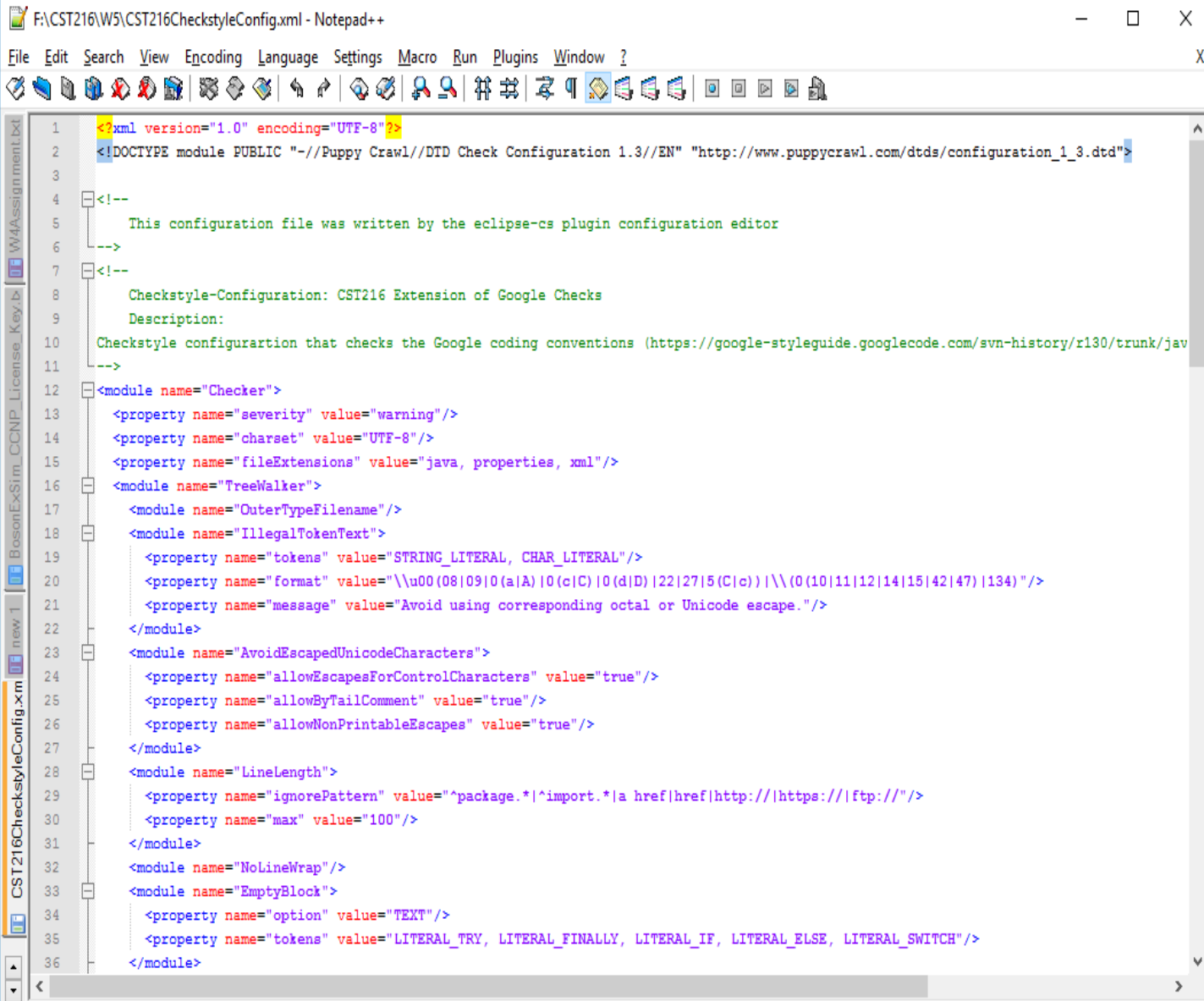
- The programming style adopted by a software development project can help to comply with good programming practices which improve the code quality, readability, re-usability, and reduce the cost of development.

9. Checkstyle Limitations

The Checkstyle tool has the following limitations:

- The performed checks mainly limit themselves to the presentation and don't analyze content.
- The performed checks do not confirm the correctness or completeness of the program.
- In practice, it can be tedious to comply with all the style constraints, some of which could possibly harm the programming stages' dynamic; so, it may be useful to determine which level of check is needed for a certain type of program.

Appendix A



The screenshot shows a Notepad++ window titled "F:\CST216\W5\CST216CheckstyleConfig.xml - Notepad++". The window contains an XML file with the following content:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE module PUBLIC "-//Puppy Crawl//DTD Check Configuration 1.3//EN" "http://www.puppycrawl.com/dtds/configuration_1_3.dtd">
3
4 <!--
5     This configuration file was written by the eclipse-cs plugin configuration editor
6 -->
7 <!--
8     Checkstyle-Configuration: CST216 Extension of Google Checks
9     Description:
10    Checkstyle configuration that checks the Google coding conventions (https://google-styleguide.googlecode.com/svn-history/r130/trunk/jav
11 -->
12 <module name="Checker">
13     <property name="severity" value="warning"/>
14     <property name="charset" value="UTF-8"/>
15     <property name="fileExtensions" value="java, properties, xml"/>
16     <module name="TreeWalker">
17         <module name="OuterTypeFilename"/>
18         <module name="IllegalTokenText">
19             <property name="tokens" value="STRING_LITERAL, CHAR_LITERAL"/>
20             <property name="format" value="\\u00(08|09|0(a|A)|0(c|C)|0(d|D)|22|27|5(C|c))|\\(0(10|11|12|14|15|42|47)|134)"/>
21             <property name="message" value="Avoid using corresponding octal or Unicode escape."/>
22         </module>
23         <module name="AvoidEscapedUnicodeCharacters">
24             <property name="allowEscapesForControlCharacters" value="true"/>
25             <property name="allowByTailComment" value="true"/>
26             <property name="allowNonPrintableEscapes" value="true"/>
27         </module>
28         <module name="LineLength">
29             <property name="ignorePattern" value="^package.*|^import.*|a href|href|http://|https://|ftp://" />
30             <property name="max" value="100"/>
31         </module>
32         <module name="NoLineWrap"/>
33         <module name="EmptyBlock">
34             <property name="option" value="TEXT"/>
35             <property name="tokens" value="LITERAL_TRY, LITERAL_FINALLY, LITERAL_IF, LITERAL_ELSE, LITERAL_SWITCH"/>
36         </module>
```

The status bar at the bottom indicates: "eXtensible Markup Language file", "length: 6166 lines: 128", "Ln: 62 Col: 102 Sel: 0|0", "UNIX", "ANSI", and "INS".

F:\CST216\W5\CST216CheckstyleConfig.xml - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

W4Assignment.txt
BosonExSim_CNLP_License_Key.b
new 1
CST216CheckstyleConfig.xml

```
37 <module name="NeedBraces"/>
38 <module name="LeftCurly">
39   <property name="maxLength" value="100"/>
40 </module>
41 <module name="RightCurly"/>
42 <module name="RightCurly">
43   <property name="option" value="alone"/>
44   <property name="tokens" value="CLASS_DEF, METHOD_DEF, CTOR_DEF, LITERAL_FOR, LITERAL_WHILE, LITERAL_DO, STATIC_INIT, INSTANCE_INIT"/>
45 </module>
46 <module name="OneStatementPerLine"/>
47 <module name="MultipleVariableDeclarations"/>
48 <module name="MissingSwitchDefault"/>
49 <module name="PackageName">
50   <property name="format" value="^[a-z]+(\.[a-z][a-z0-9]*)*$"/>
51   <message key="name.invalidPattern" value="Package name ''{0}'' must match pattern ''{1}''."/>
52 </module>
53 <module name="TypeName">
54   <message key="name.invalidPattern" value="Type name ''{0}'' must match pattern ''{1}''."/>
55 </module>
56 <module name="MemberName">
57   <property name="format" value="^[a-z][a-z0-9][a-zA-Z0-9]*$"/>
58   <message key="name.invalidPattern" value="Member name ''{0}'' must match pattern ''{1}''."/>
59 </module>
60 <module name="ParameterName">
61   <property name="format" value="^[a-z][a-z0-9][a-zA-Z0-9]*$"/>
62   <message key="name.invalidPattern" value="Parameter name ''{0}'' must match pattern ''{1}''."/>
63 </module>
64 <module name="LocalVariableName">
65   <property name="format" value="^[a-z][a-z0-9][a-zA-Z0-9]*$"/>
66   <property name="allowOneCharVarInForLoop" value="true"/>
67   <property name="tokens" value="VARIABLE_DEF"/>
68   <message key="name.invalidPattern" value="Local variable name ''{0}'' must match pattern ''{1}''."/>
69 </module>
70 <module name="ClassTypeParameterName">
71   <property name="format" value="^[A-Z][0-9]?$|([A-Z][a-zA-Z0-9]*[T])$"/>
72   <message key="name.invalidPattern" value="Class type name ''{0}'' must match pattern ''{1}''."/>
```

eXtensible Markup Language file length : 6166 lines : 128 Ln : 62 Col : 102 Sel : 0 | 0 UNIX ANSI INS

F:\CST216\W5\CST216CheckstyleConfig.xml - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

```
73 </module>
74 <module name="MethodTypeParameterName">
75     <property name="format" value="^[A-Z][0-9]?([A-Z][a-zA-Z0-9]*[T])$"/>
76     <message key="name.invalidPattern" value="Method type name '{0}' must match pattern '{1}'."/>
77 </module>
78 <module name="GenericWhitespace">
79     <message key="ws.notPreceded" value="GenericWhitespace '{0}' is not preceded with whitespace."/>
80     <message key="ws.followed" value="GenericWhitespace '{0}' is followed by whitespace."/>
81     <message key="ws.preceded" value="GenericWhitespace '{0}' is preceded with whitespace."/>
82     <message key="ws.illegalFollow" value="GenericWhitespace '{0}' should followed by whitespace."/>
83 </module>
84 <module name="AbbreviationAsWordInName">
85     <property name="allowedAbbreviationLength" value="1"/>
86     <property name="ignoreFinal" value="false"/>
87 </module>
88 <module name="OperatorWrap">
89     <property name="option" value="NL"/>
90     <property name="tokens" value="BAND, BOR, BSR, BXOR, DIV, EQUAL, GE, GT, LAND, LE, LITERAL_INSTANCEOF, LOR, LT, MINUS, MOD, NOT_EQ
91 </module>
92 <module name="SummaryJavadocCheck"/>
93 <module name="MethodName">
94     <property name="format" value="^[a-z][a-z0-9][a-zA-Z0-9_]*$"/>
95     <message key="name.invalidPattern" value="Method name '{0}' must match pattern '{1}'."/>
96 </module>
97 <module name="EmptyCatchBlock">
98     <property name="exceptionVariableName" value="expected"/>
99 </module>
100 <module name="CommentsIndentation"/>
101 <module name="AbstractClassName"/>
102 <module name="ClassTypeParameterName"/>
103 <module name="ConstantName"/>
104 <module name="LocalFinalVariableName"/>
105 <module name="LocalVariableName"/>
106 <module name="MemberName"/>
107 <module name="StaticVariableName"/>
108 <module name="TypeName"/>
```

eXtensible Markup Language file length: 6166 lines: 128 Ln: 69 Col: 14 Sel: 0|0 UNIX ANSI INS

F:\CST216\W5\CST216CheckstyleConfig.xml - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

```
93 <module name="MethodName">
94   <property name="format" value="^[a-z][a-z0-9][a-zA-Z0-9_]*$"/>
95   <message key="name.invalidPattern" value="Method name ''{0}'' must match pattern ''{1}''."/>
96 </module>
97 <module name="EmptyCatchBlock">
98   <property name="exceptionVariableName" value="expected"/>
99 </module>
100 <module name="CommentsIndentation"/>
101 <module name="AbstractClassName"/>
102 <module name="ClassTypeParameterName"/>
103 <module name="ConstantName"/>
104 <module name="LocalFinalVariableName"/>
105 <module name="LocalVariableName"/>
106 <module name="MemberName"/>
107 <module name="StaticVariableName"/>
108 <module name="TypeName"/>
109 <module name="AbbreviationAsWordInName"/>
110 <module name="CatchParameterName"/>
111 <module name="InterfaceTypeParameterName"/>
112 <module name="EmptyForInitializerPad"/>
113 <module name="EmptyForIteratorPad"/>
114 <module name="OperatorWrap"/>
115 <module name="TypecastParenPad">
116   <property name="tokens" value="RPAREN,TYPECAST"/>
117 </module>
118 <module name="SeparatorWrap"/>
119 <module name="AvoidNestedBlocks"/>
120 <module name="DefaultComesLast"/>
121 <module name="DeclarationOrder"/>
122 <module name="EmptyStatement"/>
123 <module name="TodoComment"/>
124 <module name="UpperEll"/>
125 <module name="RedundantModifier"/>
126 </module>
127 </module>
128
```

eXtensible Markup Language file length : 6166 lines : 128 Ln : 99 Col : 14 Sel : 0 | 0 UNIX ANSI INS