Java Code Review with CheckStyle

Tools to automate code review & analysis

http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis

- JAVA
  - CheckStyle (http://checkstyle.sourceforge.net/)

CheckStyle tool

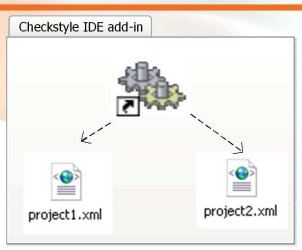
What is Checkstyle?

Checkstyle is an open-source development tool to help programmers write Java code that adheres to a coding standard

It automates the process of checking Java code to verify code following standard or not. This makes it ideal for projects that want to enforce a coding standard.

Official web site for releases and usage guide-line document http://checkstyle.sourceforge.net/

How Checkstyle works



As in the above figure:

- Checkstyle is add-ins component to IDE/Build tool
- Coding standard is user pre-defined, and embedded in each XML file
- Checkstyle will depend on XML file to parse code, then generate checking result to

How Checkstyle works

Checkstyle supports add-ins for various Java IDE and Build tools, such as:

Eclipse
NetBeans
IntelliJ
BlueJ
Borland JBuilder

Maven
Ant

Use CheckStyle with Eclipse

Sequential steps below help us to setup, define, and run checkstyle in Eclipse

Setup add-in to Eclipse:

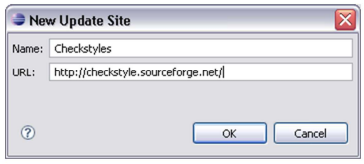
There are two ways to plug the add-in to Eclipse:

- ✓Manually download library from the official site, and copy to the Eclipse plug-ins folder

- ✓Make Eclipse to automatically download and update the library

Use CheckStyle with Eclipse

1.Open Eclipse
2.Go to Help→ Software updates → Find and install
3.Choose “Search for new features to install”, then click the “Next” button
4.Click the “New Remote Site” and input checkstyle website address, like



5.Click “OK” button, then wait for Eclipse to automatically update

Define a coding standard XML file for your own

We can create and edit your checkstyle by manually or use editor tool from Eclipse after Eclipse update checkstyle successfully

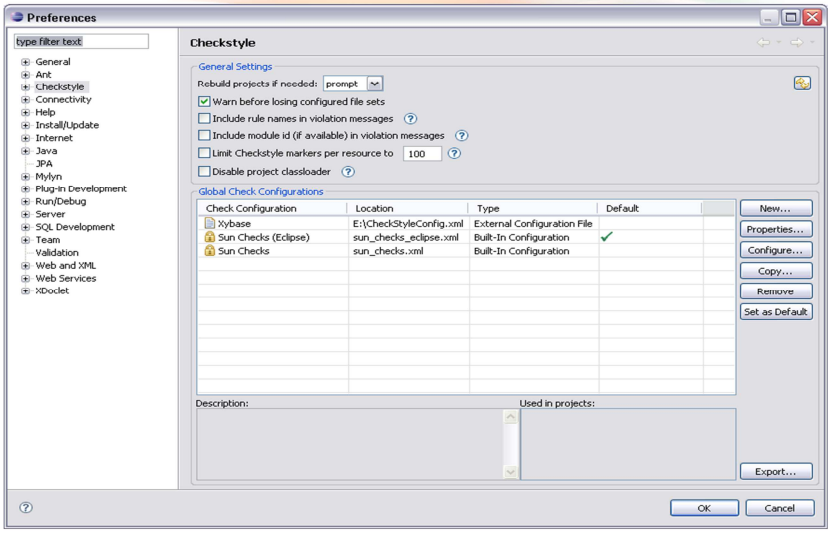
The steps here is example to use the editor

1.From Eclipse, go to Window->Preference, then click CheckStyle node on the left tree. You will see a list of defined CheckStyle

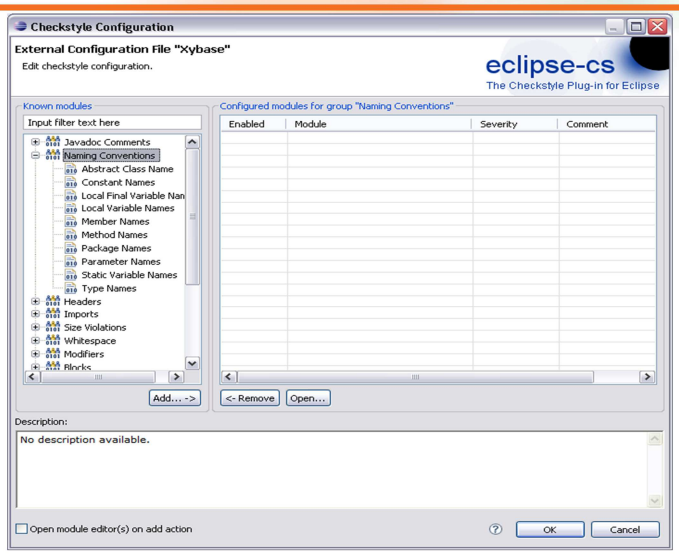
2. Click “New”, choose “Internal Configuration” and name to your checkstyle

3. The editor appeared, as below, and you can walk though sections on the left menu that checkstyle can support to check code

Define a coding standard XML file for your own



Define a coding standard XML file for your own



Active your CheckStyle

In the previous step, you already defined a checkstyle. In this section, it will show you the ways to active your checkstyle and apply the checkstyle into your project

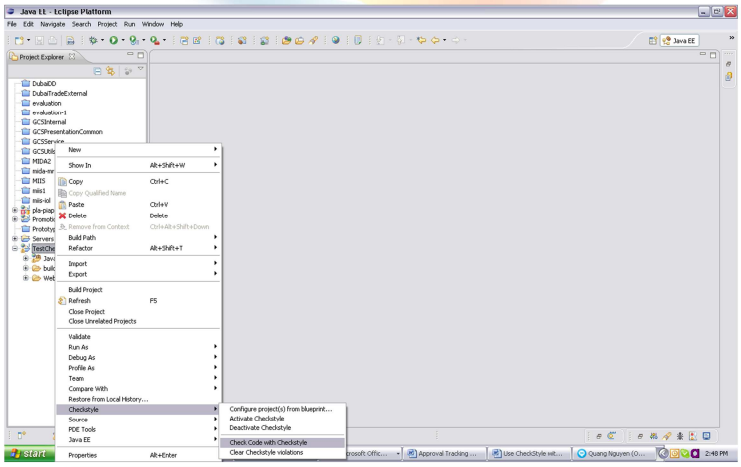
1.Right click to your project icon, and choose “Properties” menu, a dialog below appears

2.Click the “Checkstyle” menu and choose your own checkstyle from the available checkstyles list in combo-box

3. Click “OK” to finish

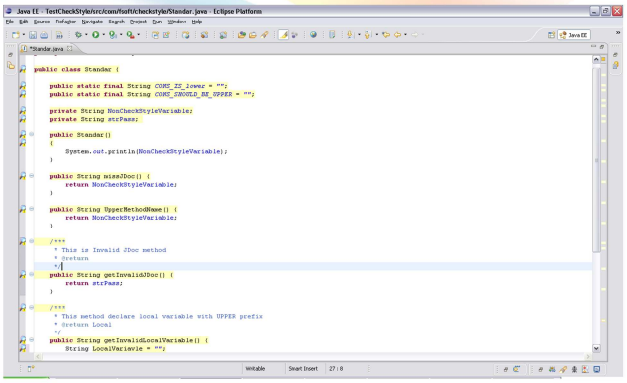
Active your CheckStyle

The screen below show us the bar we can execute checkstyle from Eclipse



Active your CheckStyle

After click the button “Check code with CheckStyle” you will see checkstyle mark YELLOW NOTE icon to class files



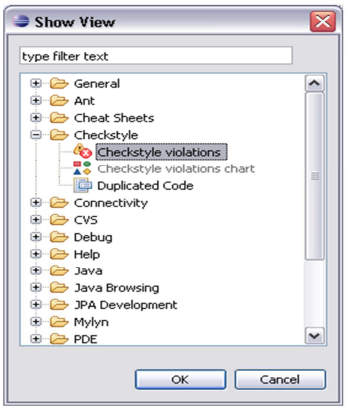
Move your mouse to the YELLOW ICON, you will receive a warning message about cause of invalid grammar. Follow that to fix and your invalid code grammar

Active your CheckStyle

Checkstyle plugin supports us two reports:

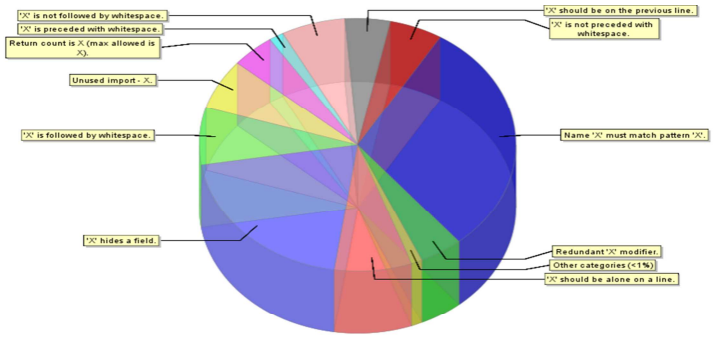
- Chart report
- Statistic report

To use two functions with Eclipse, you can go to Window->Show View->Other to see these both report types as beside:



Active your CheckStyle

Click one of them, you will see a dialog appeared with two REPORT GENERATE buttons on the left menu. Then, click to the REPORT button to see report
Followings are results of two reports

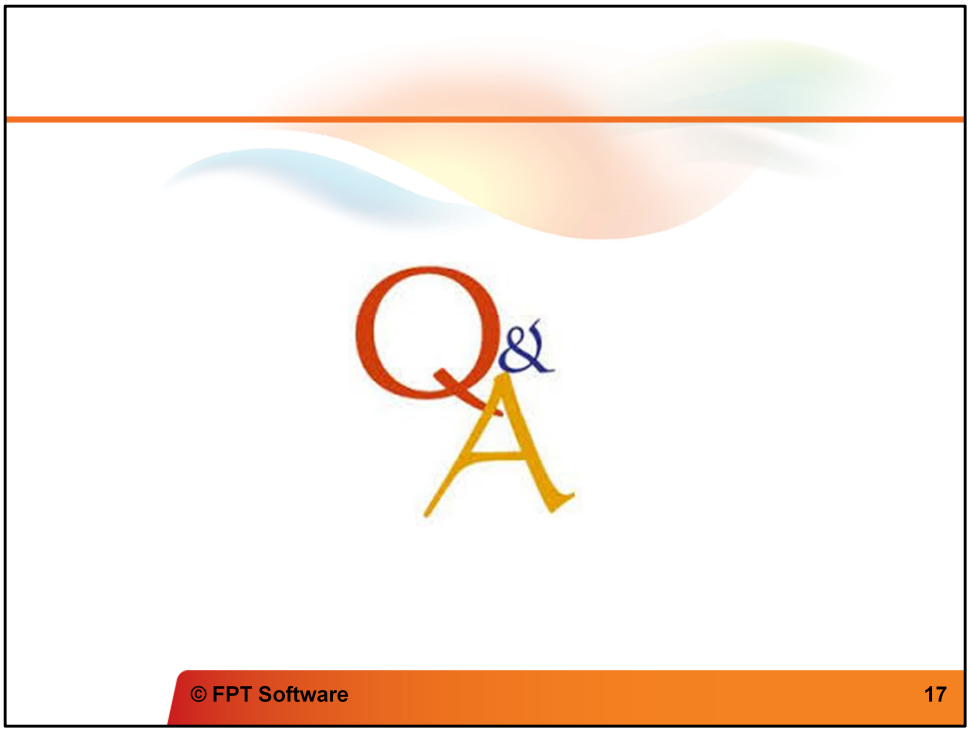


Active your CheckStyle

Checkstyle statistics

Overview of Checkstyle violations - 2205 markers in 16 categories (Filter matched 2205 of 2205 items)

Checkstyle violation type	Marker count
Name 'X' must match pattern 'X'.	665
'X' hides a field.	444
'X' should be alone on a line.	174
'X' is followed by whitespace.	162
'X' is not followed by whitespace.	144
Unused import - X.	142
'X' is not preceded with whitespace.	119
'X' should be on the previous line.	102
Return count is X (max allowed is X).	98
Redundant 'X' modifier.	97
'X' is preceded with whitespace.	33
Utility classes should not have a public or default constructor.	13
'X' modifier out of order with the JLS suggestions.	5
Each variable declaration must be in its own statement.	3


Q&A
© FPT Software
17