Software Verification

Junit, Eclipse, Static Analysis Tool

201011329 박대규
201111393 최서현
201111374 윤원경

Software Verification

INDEX

01 JUnit

What is Junit?

- JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.

-Especially, JUnit is a unit testing framework for the Java programming language.

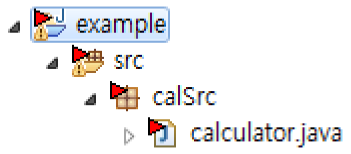
What is unit testing?

- Unit testing is a method by which individual units of source code that have possibilities to make troubles.

- Generally, you should test for public methods.

- Benefits are…

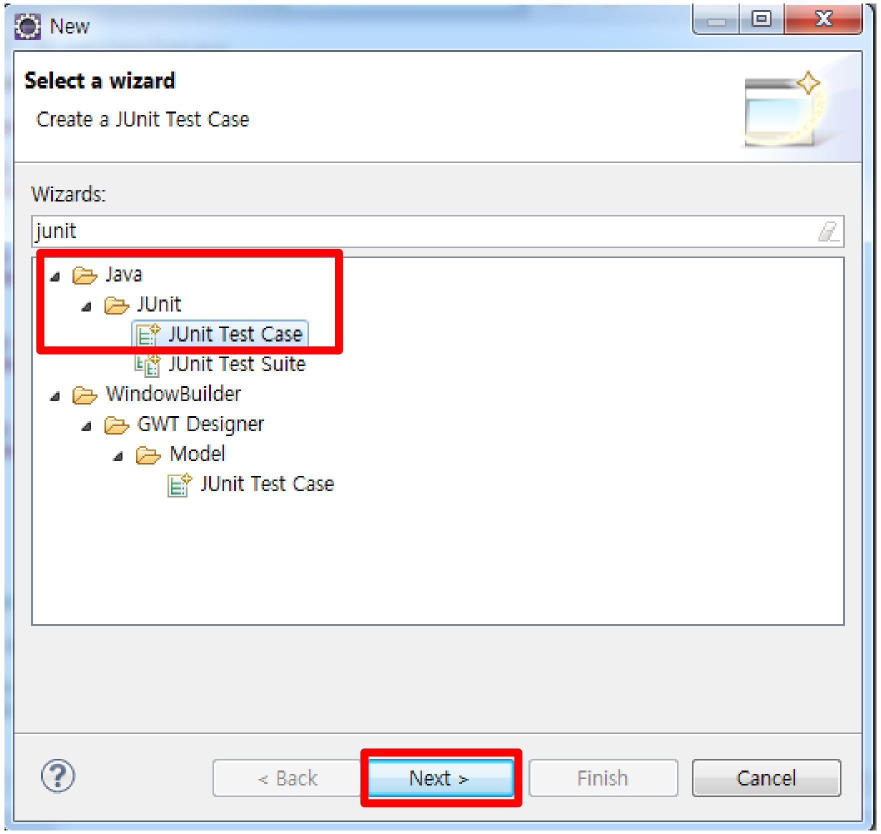
How to use JUnit



```
package calSrc;

public class calculator {

        public static int add(int a, int b){
         return a+b;
        }

        public static int subtract(int a, int b){
         return a-b;
        }

        public static int multiply(int a, int b){
         return a*b;
        }

        public static int divide(int a, int b){
         return a/b;
        }
}
```

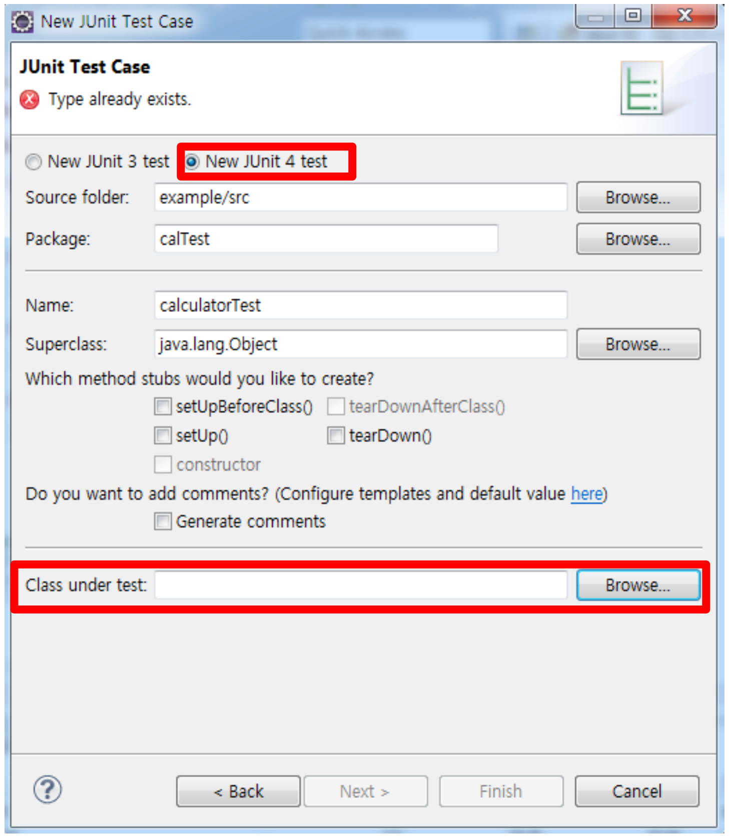
- make the java file for the example in new package

How to use JUnit(cont.)



- [File]-[new]-others…
- Java-JUnit-Junit Test Case

How to use JUnit(cont.)



New JUnit Test Case

JUnit Test Case

Type already exists.

New JUnit 3 test [✓ New JUnit 4 test]

Source folder: example/src

Package: calTest

Name: calculatorTest

Superclass: java.lang.Object

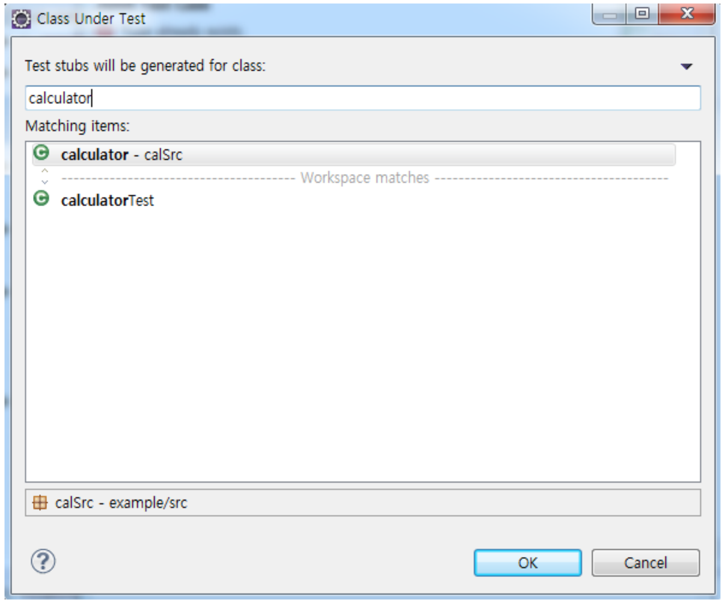
Which method stubs would you like to create?

- [ ] setUpBeforeClass()
- [ ] tearDownAfterClass()
- [ ] setUp()
- [ ] tearDown()
- [ ] constructor

Do you want to add comments? (Configure templates and default value here)

- [ ] Generate comments

Class under test:



Class Under Test

Test stubs will be generated for class:

calculator

Matching items:

calculator - calSrc

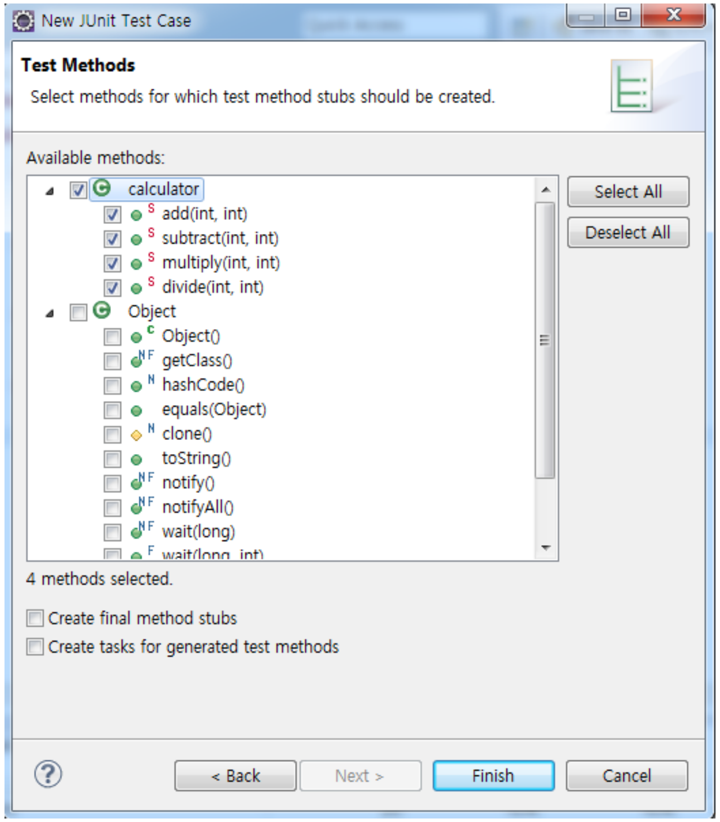
Workspace matches

calculatorTest

calSrc - example/src

- Choose New JUnit 4 Test, and put the new package and name. Then, click the browse button next to class under test and select the class to test.

How to use JUnit(cont.)



New JUnit Test Case

Test Methods

Select methods for which test method stubs should be created.

Available methods:

- [✓] calculator
  - [✓] add(int, int)
  - [✓] subtract(int, int)
  - [✓] multiply(int, int)
  - [✓] divide(int, int)
- [ ] Object
  - [ ] Object()
  - [ ] getClass()
  - [ ] hashCode()
  - [ ] equals(Object)
  - [ ] clone()
  - [ ] toString()
  - [ ] notify()
  - [ ] notifyAll()
  - [ ] wait(long)
  - [ ] wait(long, int)

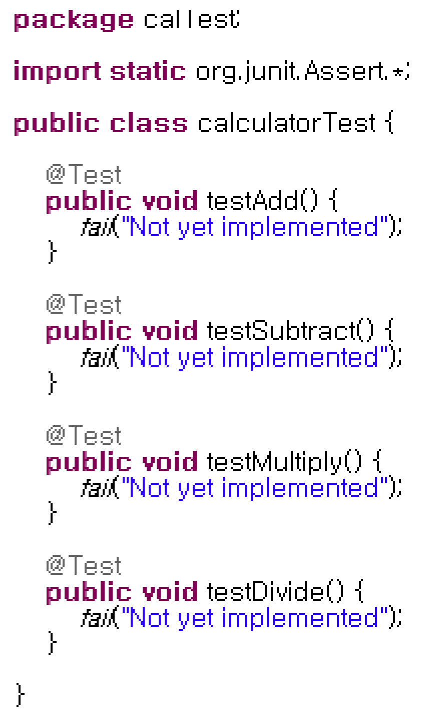
4 methods selected.

[ ] Create final method stubs

[ ] Create tasks for generated test methods

- Choose methods to test.

How to use JUnit(cont.)

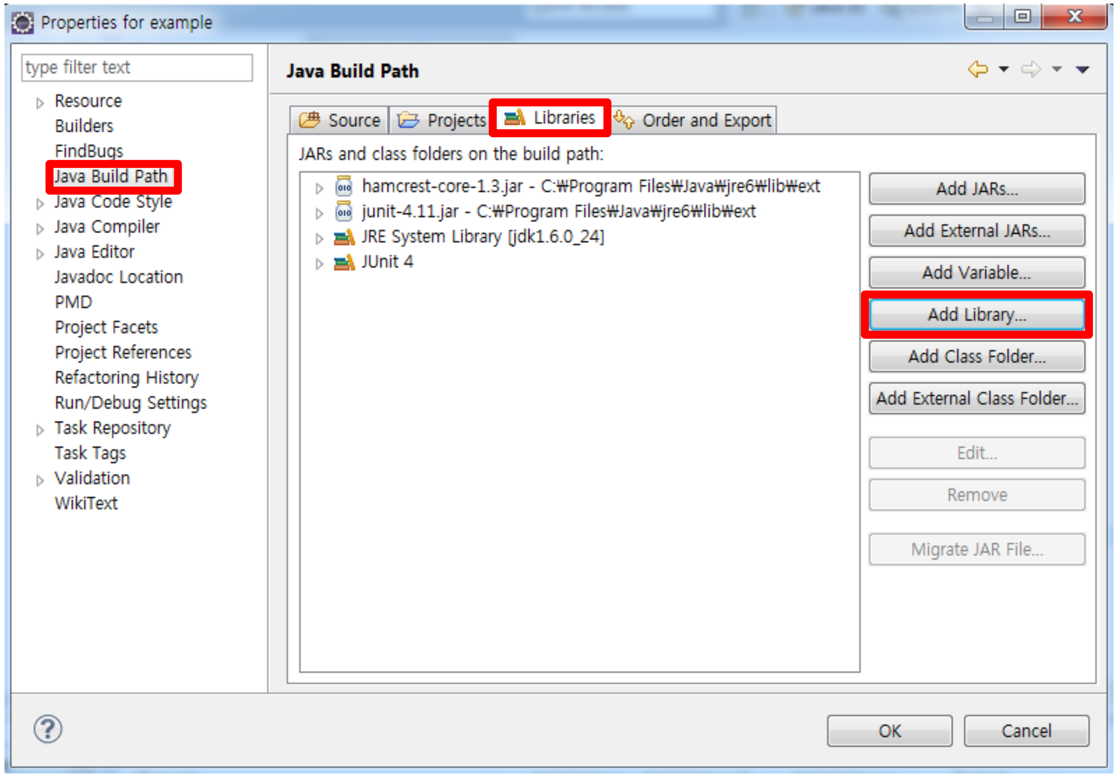


```
package calTest;

import static org.junit.Assert.*;

public class calculatorTest {

    @Test
    public void testAdd() {
        fail("Not yet implemented");
    }

    @Test
    public void testSubtract() {
        fail("Not yet implemented");
    }

    @Test
    public void testMultiply() {
        fail("Not yet implemented");
    }

    @Test
    public void testDivide() {
        fail("Not yet implemented");
    }

}
```

- TestCase is created.

JUnit.org

How to use JUnit(cont.)



Properties for example

type filter text

Resource
Builders
FindBugs
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
PMD
Project Facets
Project References
Refactoring History
Run/Debug Settings
Task Repository
Task Tags
Validation
WikiText

Java Build Path

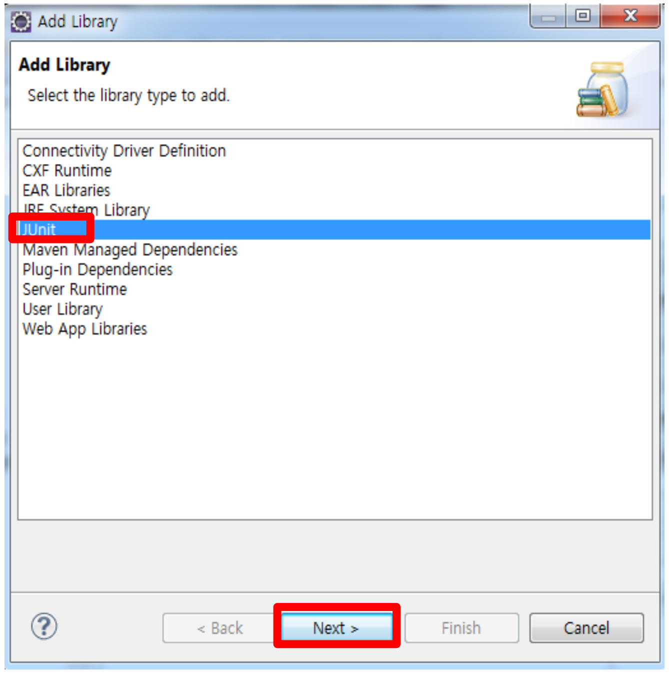
Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

hamcrest-core-1.3.jar - C:₩Program Files₩Java₩jre6₩lib₩ext
junit-4.11.jar - C:₩Program Files₩Java₩jre6₩lib₩ext
JRE System Library [jdk1.6.0_24]
JUnit 4

- right click your project.
- [Properties] – [Java Build Path] – [Library] – [Add Library]

How to use JUnit(cont.)

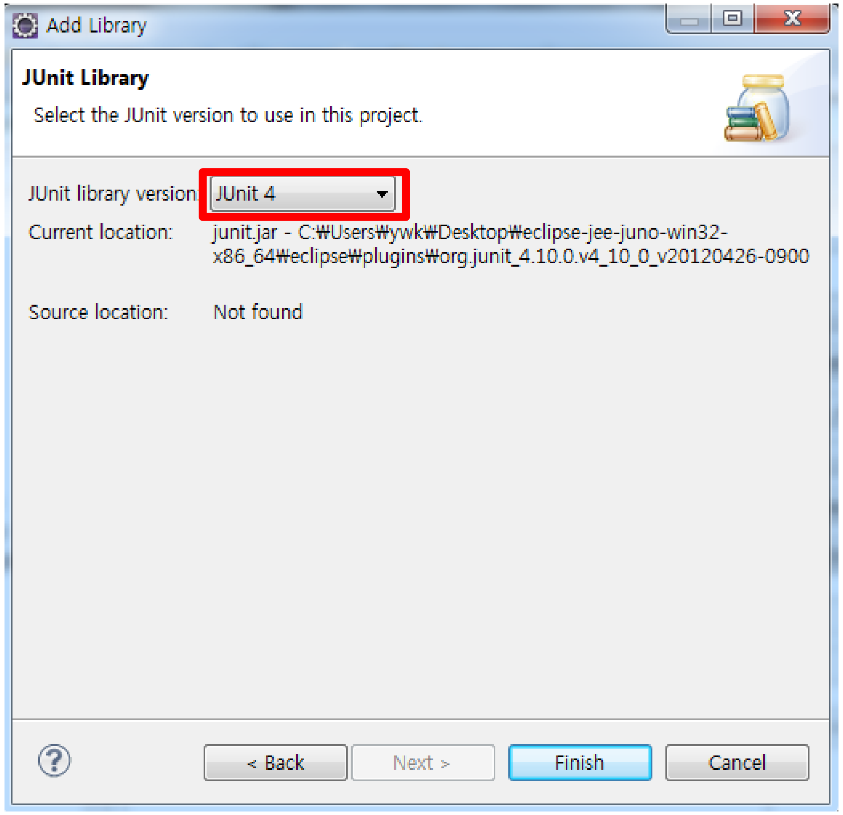


Add Library

Add Library

Select the library type to add.

Connectivity Driver Definition
CXF Runtime
EAR Libraries
JRE System Library
JUnit
Maven Managed Dependencies
Plug-in Dependencies
Server Runtime
User Library
Web App Libraries



Add Library

JUnit Library

Select the JUnit version to use in this project.

JUnit library version: JUnit 4

Current location: junit.jar - C:₩Users₩ywk₩Desktop₩eclipse-jee-juno-win32-x86_64₩eclipse₩plugins₩org.junit_4.10.0.v4_10_0_v20120426-0900

Source location: Not found

- Choose the JUnit 4 and Finish.

JUnit.org

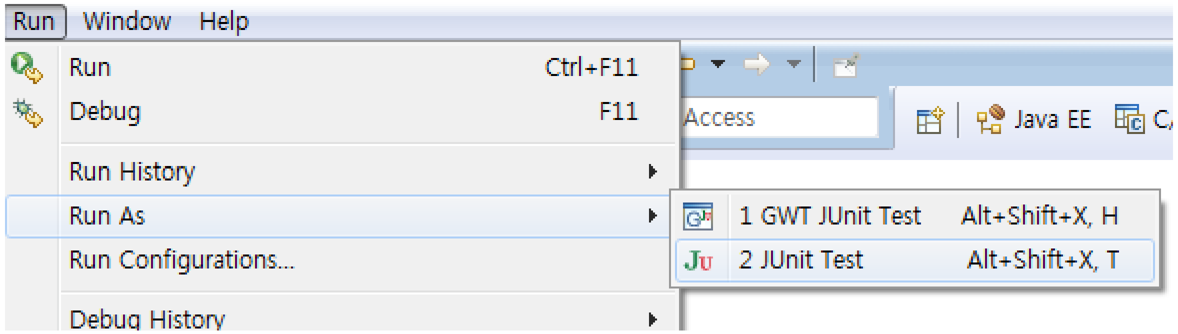
How to use JUnit(cont.)

```
package calTest;

import static org.junit.Assert.*;

import org.junit.Test;

public class calculatorTest {

    @Test
    public void test() {
        fail("Not yet implemented");
    }

}
```

```
package calTest;

import calSrc.calculator;
import static org.junit.Assert.*;

import org.junit.*;

public class calculatorTest {

    @Test
    public void testAdd() {
        int result=calculator.add(1, 2);
        assertEquals("ADD FAIL",result,1);
    }

    @Test
    public void testSubtract() {
        int result=calculator.subtract(2, 1);
        assertEquals(result,1);
    }

    @Test
    public void testMultiply() {
        int result=calculator.multiply(1, 2);
        assertEquals(result,2);
    }

    @Test
    public void testDivide() {
        int result=calculator.divide(2, 1);
        assertEquals(result,2);
    }
}
```

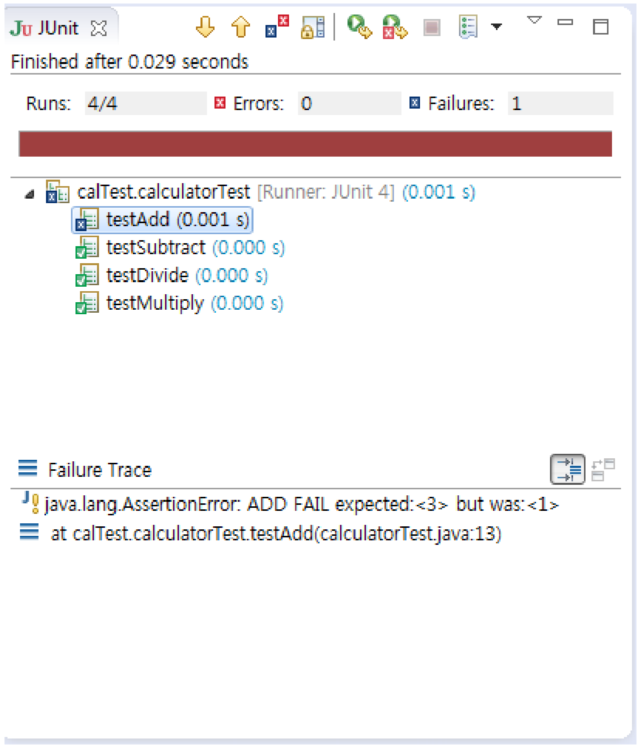
- TestCase is created.
Fill some methods to test.

How to use JUnit(cont.)



- [Run]-[Run as]-[JUnit Test]

How to use JUnit(cont.)



```
Finished after 0.029 seconds
calTest.calculatorTest [Runner: JUnit 4] (0.001 s)
    testAdd (0.001 s)
    testSubtract (0.000 s)
    testDivide (0.000 s)
    testMultiply (0.000 s)

java.lang.AssertionError: ADD FAIL expected:<3> but was:<1>
at calTest.calculatorTest.testAdd(calculatorTest.java:13)
```

- You can see the results.

Annotations of JUnit

Annotation	Explanation
@Test @Test(timeout=ms) @Test(expected=class)	Written for the test method. When timeout paramater exists, the test would fail if the test takes longer than defined time. When expected parameter exists, the test would fail if the test doesn't make defined exception.
@Ignore	If you want to see the report but test code is not implemented yet, you can use this annotation.

@After @Before	It's same as setUp() and tearDown() in JUnit3. It is executed before and after each test methods.
@AfterClass @BeforeClass	It's same as setUpBeforeClass() and tearDownAfterClass() in JUnit3. It is executed before and after the class.

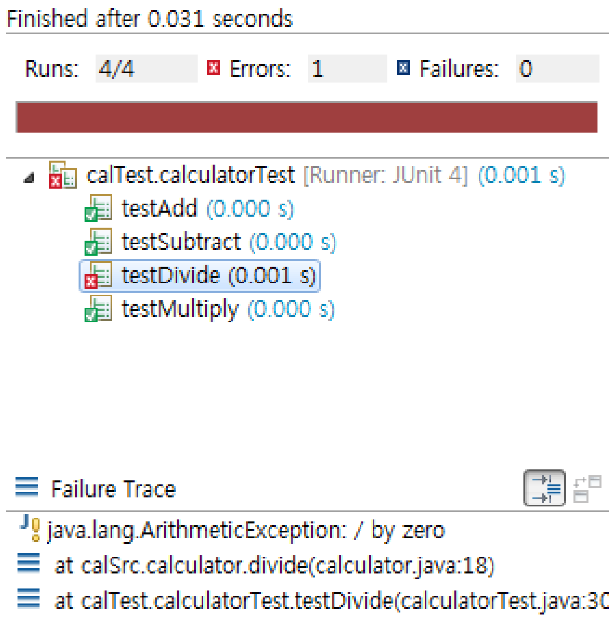
Annotations of JUnit(cont.)

Annotation	Explanation
@RunWith(value=class)	It is used to decide test runner.
@SuiteClasses(value={classes})	It is used to define tests which would be included in test suites.
@Parameters	It is used for multiple tests for several parameters.

Examples for using Annotation

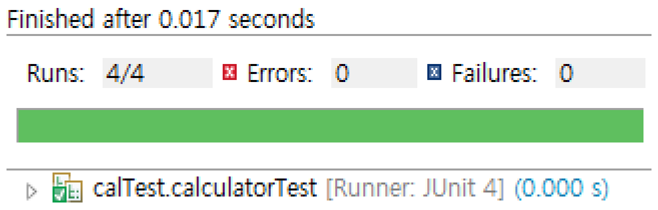
@Test(expected=exception class)

```
@Test
public void testDivide() {
    int result=calculator.divide(2, 0);
    assertEquals(result,2);
}
```



```
Finished after 0.031 seconds
Runs: 4/4   Errors: 1   Failures: 0

calTest.calculatorTest [Runner: JUnit 4] (0.001 s)
    testAdd (0.000 s)
    testSubtract (0.000 s)
    testDivide (0.001 s)
    testMultiply (0.000 s)

Failure Trace
java.lang.ArithmeticException: / by zero
at calSrc.calculator.divide(calculator.java:18)
at calTest.calculatorTest.testDivide(calculatorTest.java:3C
```

```
@Test(expected=ArithmeticException.class)
public void testDivide() {
    int result=calculator.divide(2, 0);
    assertEquals(result,2);
}
```

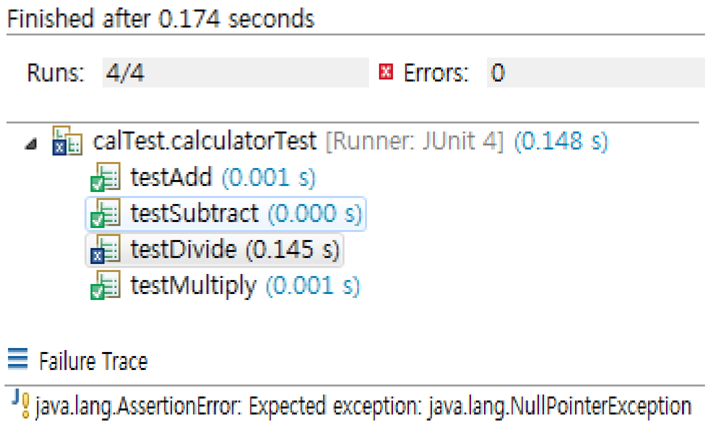


```
Finished after 0.017 seconds
Runs: 4/4   Errors: 0   Failures: 0

calTest.calculatorTest [Runner: JUnit 4] (0.000 s)
```

Examples for using Annotation

@Test(expected=exception class)

```
@Test(expected=NullPointerException.class)
public void testDivide() {
    int result=calculator.divide(2, 1);
    assertEquals(result,2);
}
```

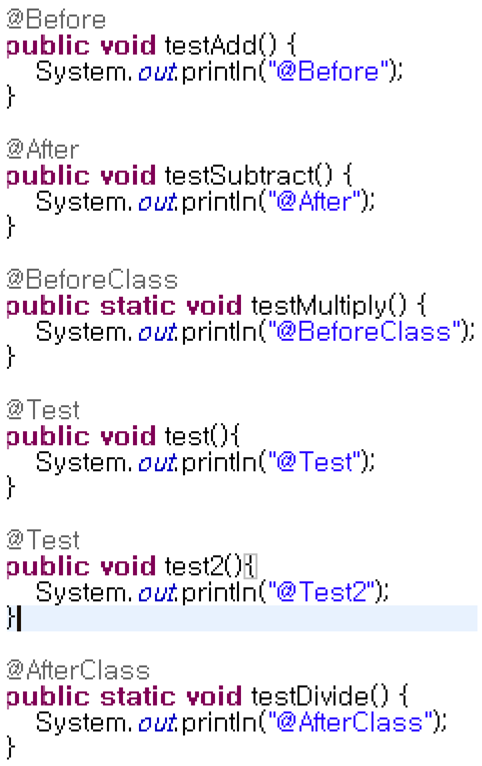


```
Finished after 0.174 seconds

Runs: 4/4        Errors: 0

calTest.calculatorTest [Runner: JUnit 4] (0.148 s)
    testAdd (0.001 s)
    testSubtract (0.000 s)
    testDivide (0.145 s)
    testMultiply (0.001 s)

Failure Trace
java.lang.AssertionError: Expected exception: java.lang.NullPointerException
```

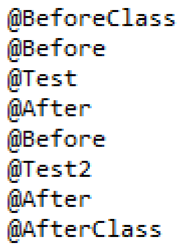
JUnit.org

Examples for using Annotation

@Before,After,BeforeClass,AfterClass



```
@Before
public void testAdd() {
    System.out.println("@Before");
}

@After
public void testSubtract() {
    System.out.println("@After");
}

@BeforeClass
public static void testMultiply() {
    System.out.println("@BeforeClass");
}

@Test
public void test(){
    System.out.println("@Test");
}

@Test
public void test2(){
    System.out.println("@Test2");
}

@AfterClass
public static void testDivide() {
    System.out.println("@AfterClass");
}
```

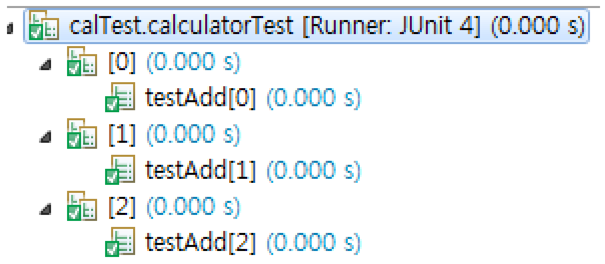


```
@BeforeClass
@Before
@Test
@After
@Before
@Test2
@After
@AfterClass
```

Examples for using Annotation

@Parameters,RunWith

```
@RunWith(value=Parameterized.class)
public class calculatorTest {

    int a,b,expected;

    @Parameters
    public static Collection params()
    {
        return Arrays.asList(new Object[] [] {
            {1,2,3}, {1,3,4}, {2,3,5}
        });
    }

    public calculatorTest(int a, int b,int expected)
    {
        this.a=a;
        this.b=b;
        this.expected=expected;
    }

    @Test
    public void testAdd() {
        calculator cal=new calculator();
        assertEquals(cal.add(a,b),expected);
    }
}
```



```
calTest.calculatorTest [Runner: JUnit 4] (0.000 s)
    [0] (0.000 s)
        testAdd[0] (0.000 s)
    [1] (0.000 s)
        testAdd[1] (0.000 s)
    [2] (0.000 s)
        testAdd[2] (0.000 s)
```

JUnit.org

Methods of JUnit

Method	Explanation
assertEquals (expected type, result type)	After comparing expected value with result value, returns success if those are same.
assertEquals (expected object, result object)	After comparing expected object's value with result object's value, returns success if those are same.
assertSame (expected object, result object)	After comparing expected object's memory address and result object's memory address, returns success if those are same.

assertNull(object)
 	
assertNotNull(object)
	After testing object is null or not, returns success if object is null.
	
	After testing object is null or not, returns success if object is not null.

assertTrue(boolean)
 	
assertFalse(boolean)
	After condition is true or not, returns success if condition is true.
	
	After condition is true, or not returns success if condition is false.

Solutions for error

- InitializationError

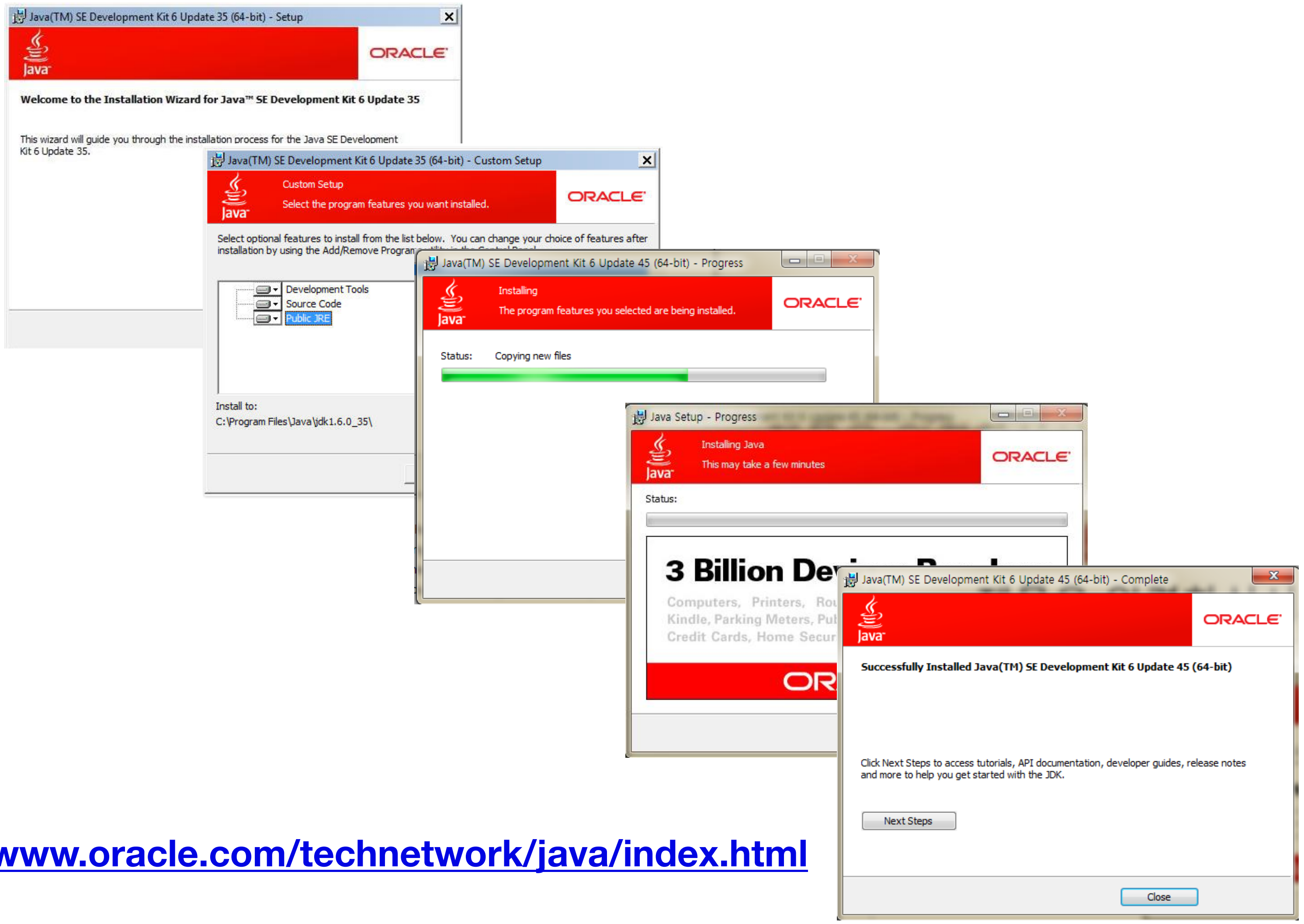
java.lang.noclassDefFoundError: org/hamcrest/selfDescribing

-> add the hamcrests.jar as the external library…

02 Eclipse

How to install JDK

- 01 JUnit
- **02 Eclipse**
- 03 CheckStyle
- 04 PMD
- 05 FindBugs
- 06 SONAR


Java(TM) SE Development Kit 6 Update 35 (64-bit) - Setup

Welcome to the Installation Wizard for Java™ SE Development Kit 6 Update 35

This wizard will guide you through the installation process for the Java SE Development Kit 6 Update 35.

Java(TM) SE Development Kit 6 Update 35 (64-bit) - Custom Setup

Custom Setup

Select the program features you want installed.

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Program

Development Tools
Source Code
Public JRE

Install to:
C:\Program Files\Java\jdk1.6.0_35\

Java(TM) SE Development Kit 6 Update 45 (64-bit) - Progress

Installing

The program features you selected are being installed.

Status: Copying new files

Java Setup - Progress

Installing Java

This may take a few minutes

Status:

3 Billion De

Computers, Printers, Rou
Kindle, Parking Meters, Pul
Credit Cards, Home Secur

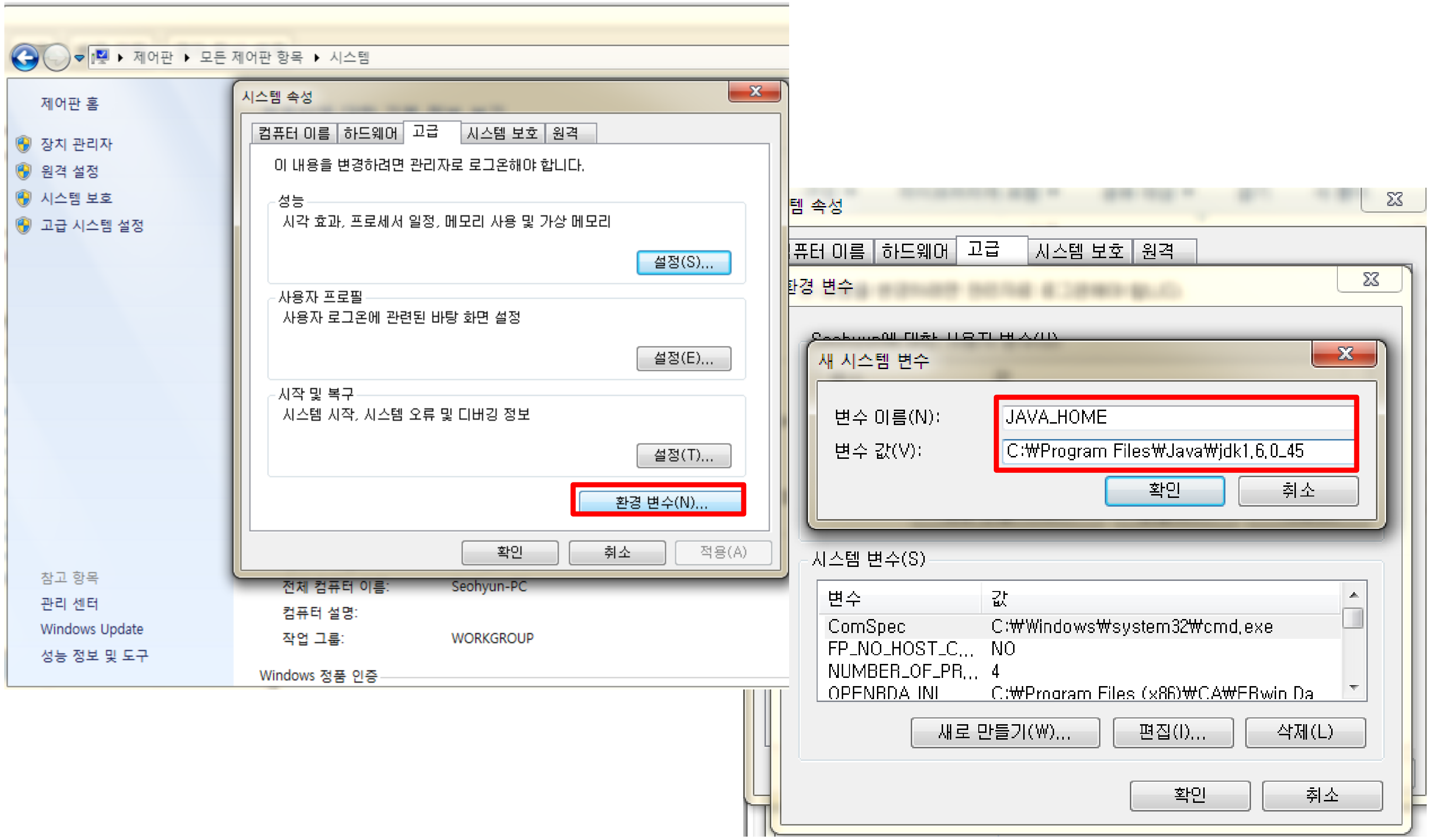
Java(TM) SE Development Kit 6 Update 45 (64-bit) - Complete

Successfully Installed Java(TM) SE Development Kit 6 Update 45 (64-bit)

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

http://www.oracle.com/technetwork/java/index.html

How to install JDK(cont.)



제어판 ▸ 모든 제어판 항목 ▸ 시스템

제어판 홈
장치 관리자
원격 설정
시스템 보호
고급 시스템 설정

참고 항목
관리 센터
Windows Update
성능 정보 및 도구

시스템 속성

컴퓨터 이름 | 하드웨어 | 고급 | 시스템 보호 | 원격

이 내용을 변경하려면 관리자로 로그온해야 합니다.

성능
시각 효과, 프로세서 일정, 메모리 사용 및 가상 메모리
설정(S)...

사용자 프로필
사용자 로그온에 관련된 바탕 화면 설정
설정(E)...

시작 및 복구
시스템 시작, 시스템 오류 및 디버깅 정보
설정(T)...

환경 변수(N)...

확인 | 취소 | 적용(A)

전체 컴퓨터 이름:	Seohyun-PC
컴퓨터 설명:	
작업 그룹:	WORKGROUP

Windows 정품 인증

새 시스템 변수

변수 이름(N):	JAVA_HOME
변수 값(V):	C:₩Program Files₩Java₩jdk1.6.0_45

확인 | 취소

시스템 변수(S)

변수	값
ComSpec	C:₩Windows₩system32₩cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_PR...	4
OPENBDA_INI	C:₩Program Files (x86)₩CA₩ERwin Da

새로 만들기(W)... | 편집(I)... | 삭제(L)

확인 | 취소

- Set up the environmental variable
  - tells the location of the installed JDK

How to install JDK(cont.)

01 JUnit

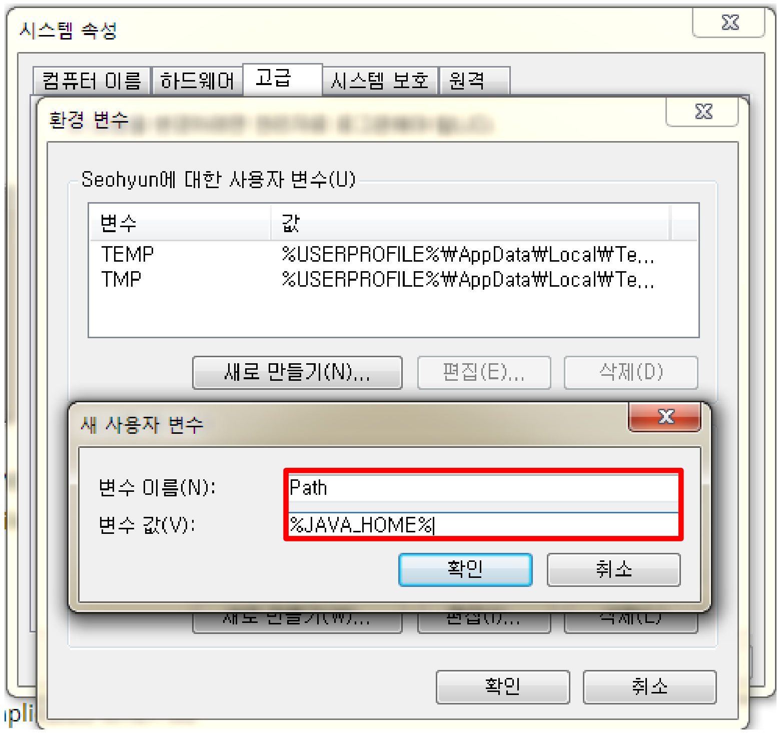
02 Eclipse

03 CheckStyle

04 PMD

05 FindBugs

06 SONAR



시스템 속성

컴퓨터 이름 | 하드웨어 | 고급 | 시스템 보호 | 원격

환경 변수

Seohyun에 대한 사용자 변수(U)

변수	값
TEMP	%USERPROFILE%₩AppData₩Local₩Te...
TMP	%USERPROFILE%₩AppData₩Local₩Te...

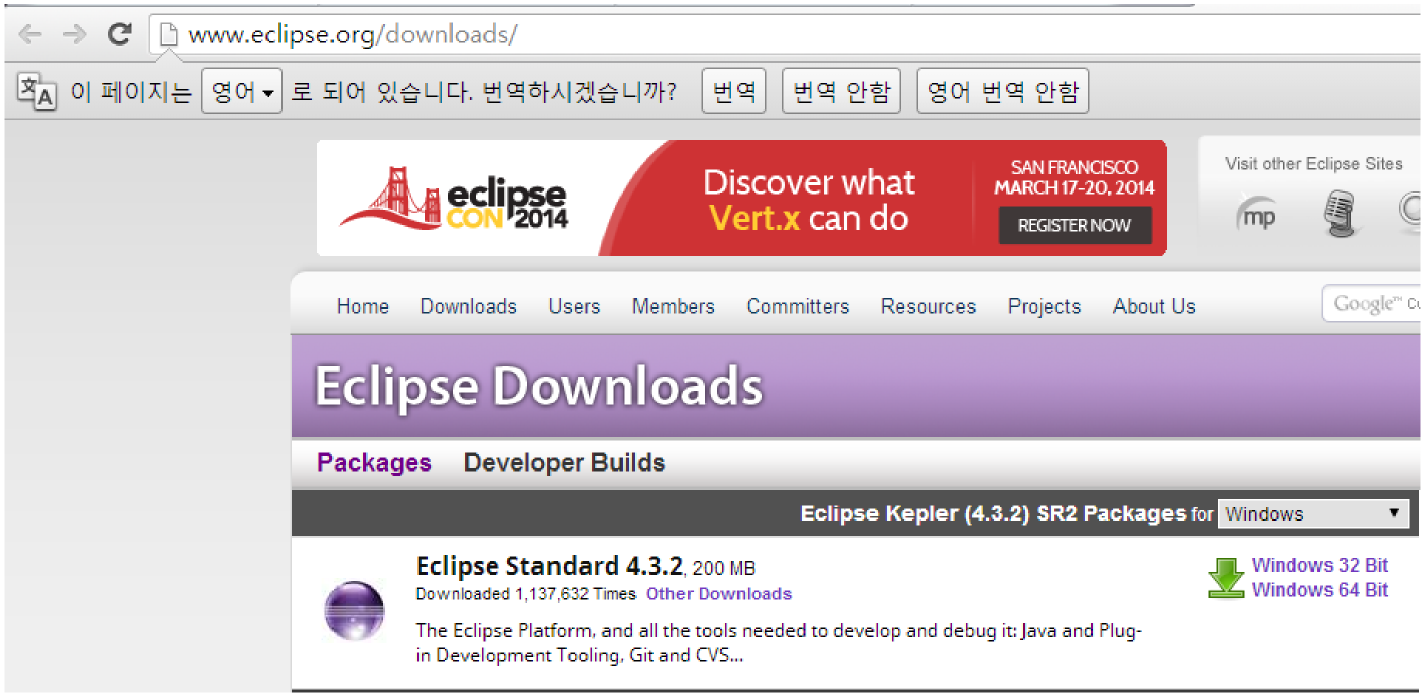
새 사용자 변수

변수 이름(N): Path

변수 값(V): %JAVA_HOME%

- **Set up the path variable**
to help to find the executable file like java.exe for other tools.

How to install Eclipse

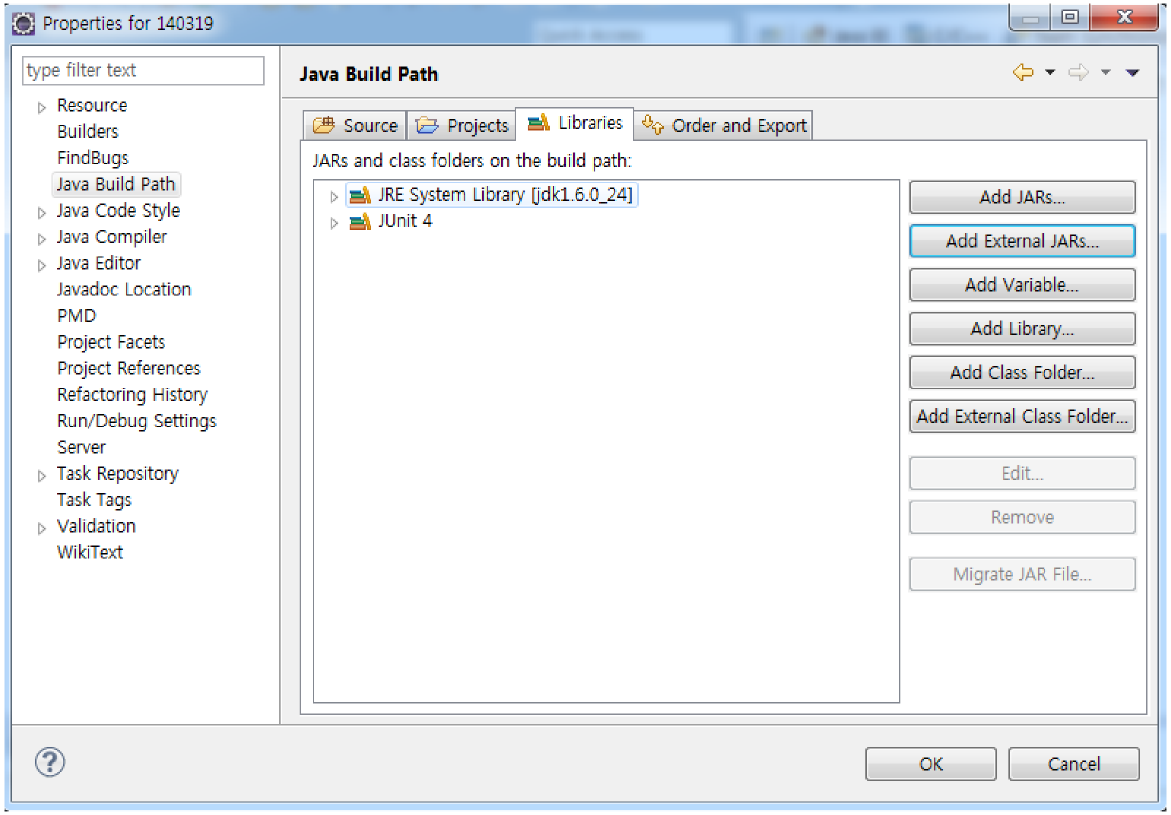


www.eclipse.org/downloads/
이 페이지는 영어 로 되어 있습니다. 번역하시겠습니까?
번역
번역 안함
영어 번역 안함
eclipse CON 2014
Discover what Vert.x can do
SAN FRANCISCO MARCH 17-20, 2014
REGISTER NOW
Visit other Eclipse Sites
Home
Downloads
Users
Members
Committers
Resources
Projects
About Us
Eclipse Downloads
Packages
Developer Builds
Eclipse Kepler (4.3.2) SR2 Packages for Windows
Windows 32 Bit
Windows 64 Bit

Eclipse Standard 4.3.2, 200 MB
Downloaded 1,137,632 Times Other Downloads
The Eclipse Platform, and all the tools needed to develop and debug it: Java and Plug-in Development Tooling, Git and CVS...

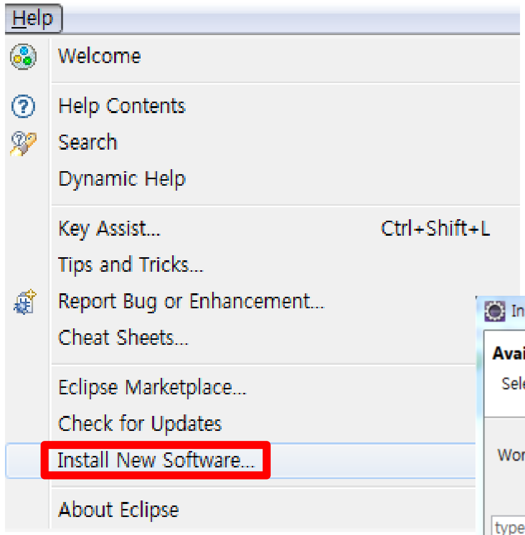
- **http://www.eclipse.org/downloads/**

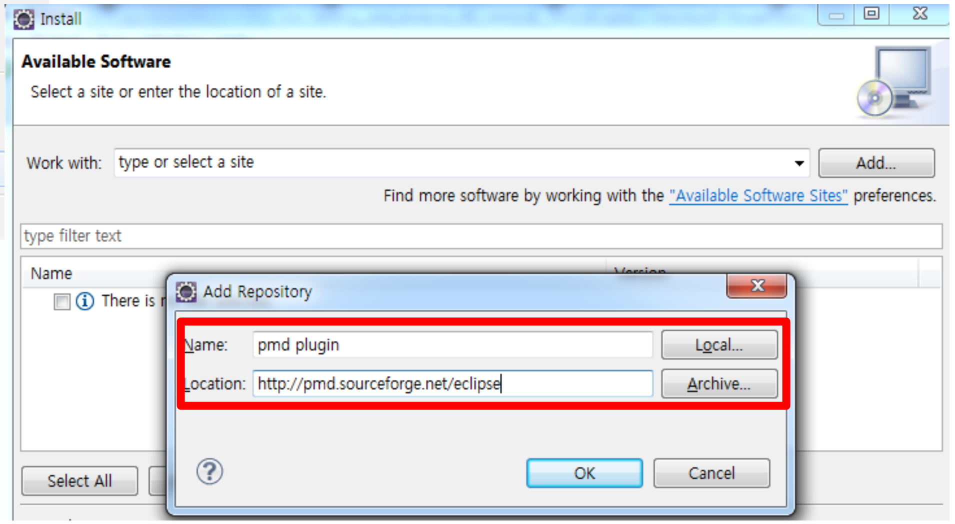
How to add external jar



- right click your project-[Preferences]-[Java Build Path] -[Add External JARs...]

How to install new software




Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the "Available Software Sites" preferences.

Add Repository

Name: pmd plugin

Location: http://pmd.sourceforge.net/eclipse

- [Help] – [Install new software…]

Keyboard shortcuts in Eclipse

Keyboard shortcut	Explaination
Ctrl+Space	Auto complete
Ctrl+Shift+O	Auto import the library
Ctrl+I	Auto indent
Ctrl+Shift+/	Making blocked lines comments

03 CheckStyle

Overview

- Checkstyle is a Open Source development tool written by Oliver Burn, Lars Kühne.

- Checkstyle is a development tool to help programmers write Java code that adheres to a **coding standard**.

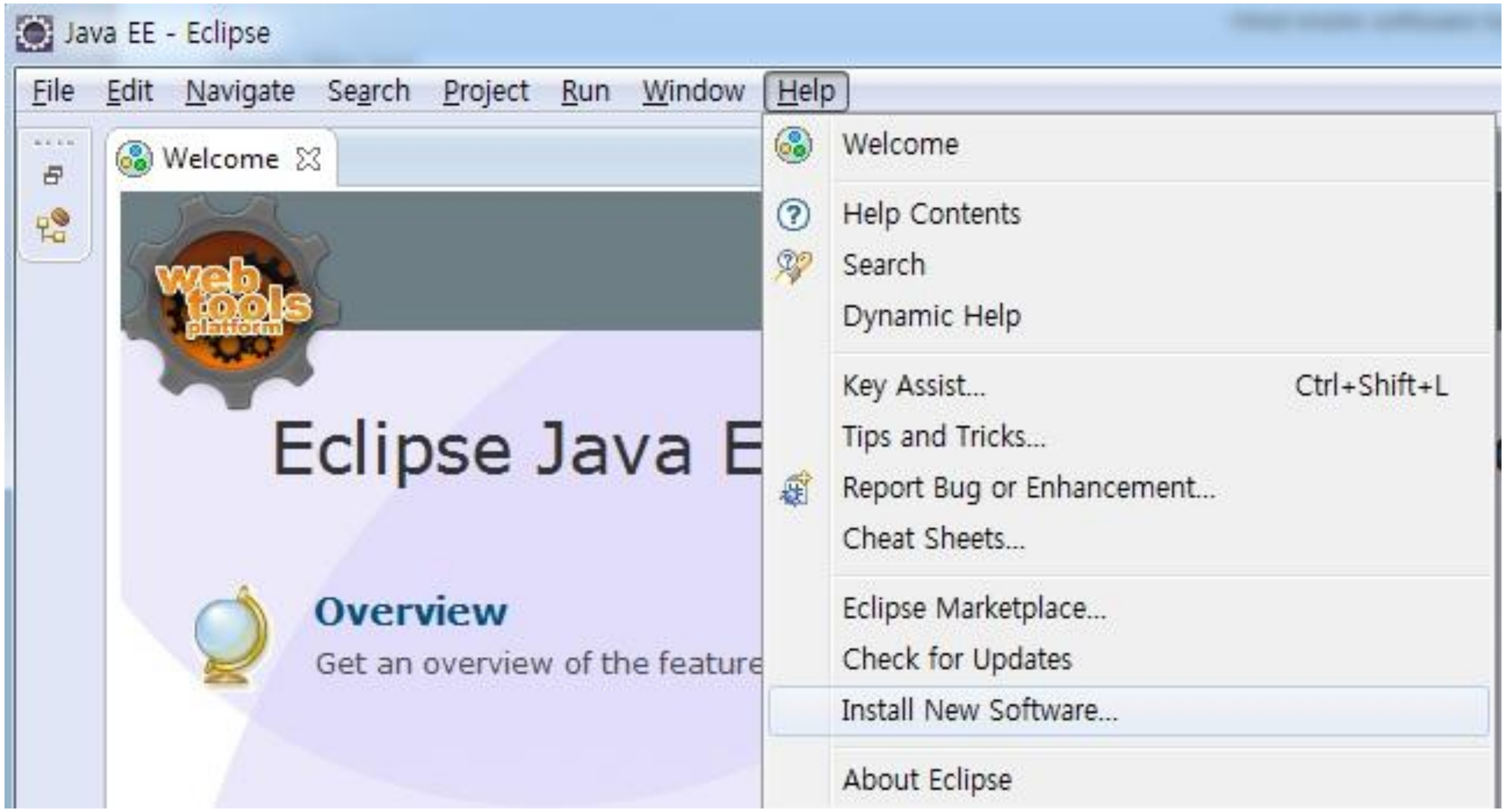
- Checkstyle provides checks that find class design problems, duplicate code, or bug patterns like double checked locking.

Overview(cont.)

IDE / Build tool	Main/Initial Author	Available from
SCM-Manager		SCM-Manager Plugin Page
jGRASP	Larry Barowski	jGRASP Home Page
Sonar	Freddy Mallet (initial author)	Sonar Home Page
Eclipse/RAD/RDz	David Schneider	Eclipse-CS Home Page
Eclipse/RAD/RDz	Roman Ivanov	Project Page
Eclipse/RAD/RDz	Marco van Meegen	Checklipse Home Page
IntelliJ IDEA	Jakub Slawinski	QAPlug
IntelliJ IDEA	James Shiell	Checkstyle-idea Project Page
IntelliJ IDEA	Mark Lussier	JetStyle Project Page
NetBeans	Petr Hejl	Checkstyle Beans
NetBeans	Paul Goulbourn	nbCheckStyle
NetBeans		Software Quality Environment (SQE)
	jCoderZ	fawkeZ
BlueJ	Rick Giles	bluejcheckstyle home page
tIDE		Built in
Emacs JDE	Markus Mohnen	Part of the standard JDEE distribution
jEdit	Todd Papaioannou	JEdit CheckStylePlugin
Vim editor	Xandy Johnson	Plugin Homepage
Maven	Vincent Massol	Checkstyle supported out of the box
QALab	Benoit Xhenseval	QALab Home Page

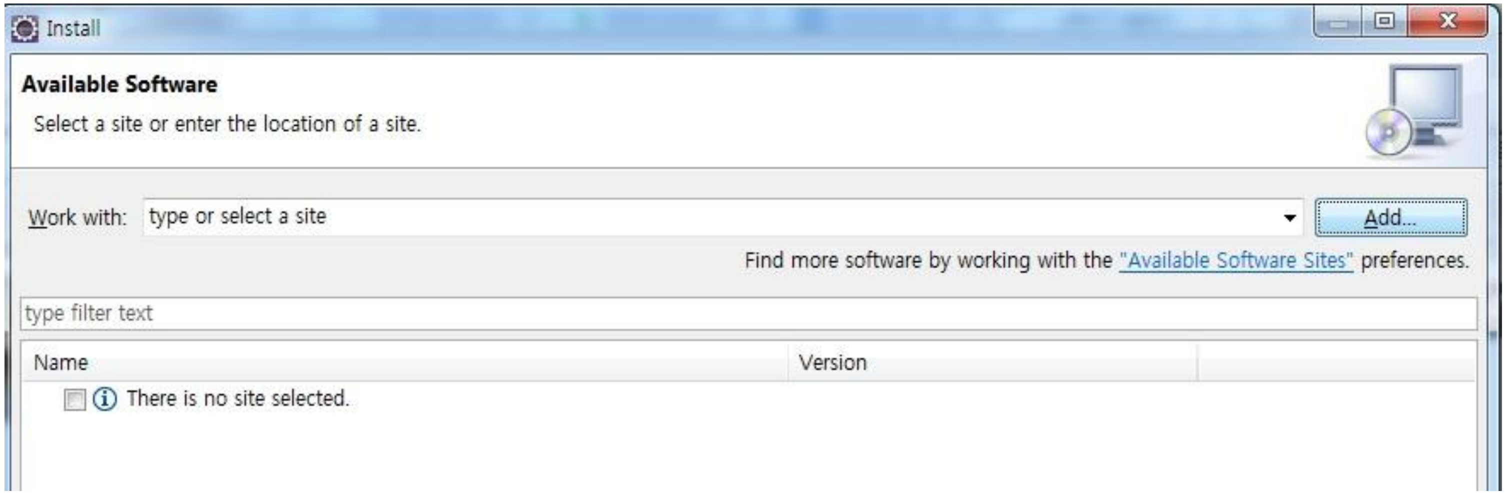
- Checkstyle is most useful if you integrate it in your build process or your developme-nt environment

How to install checkstyle



- [Help => Install New Software] Click!

How to install checkstyle(cont.)



Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site [Add...]

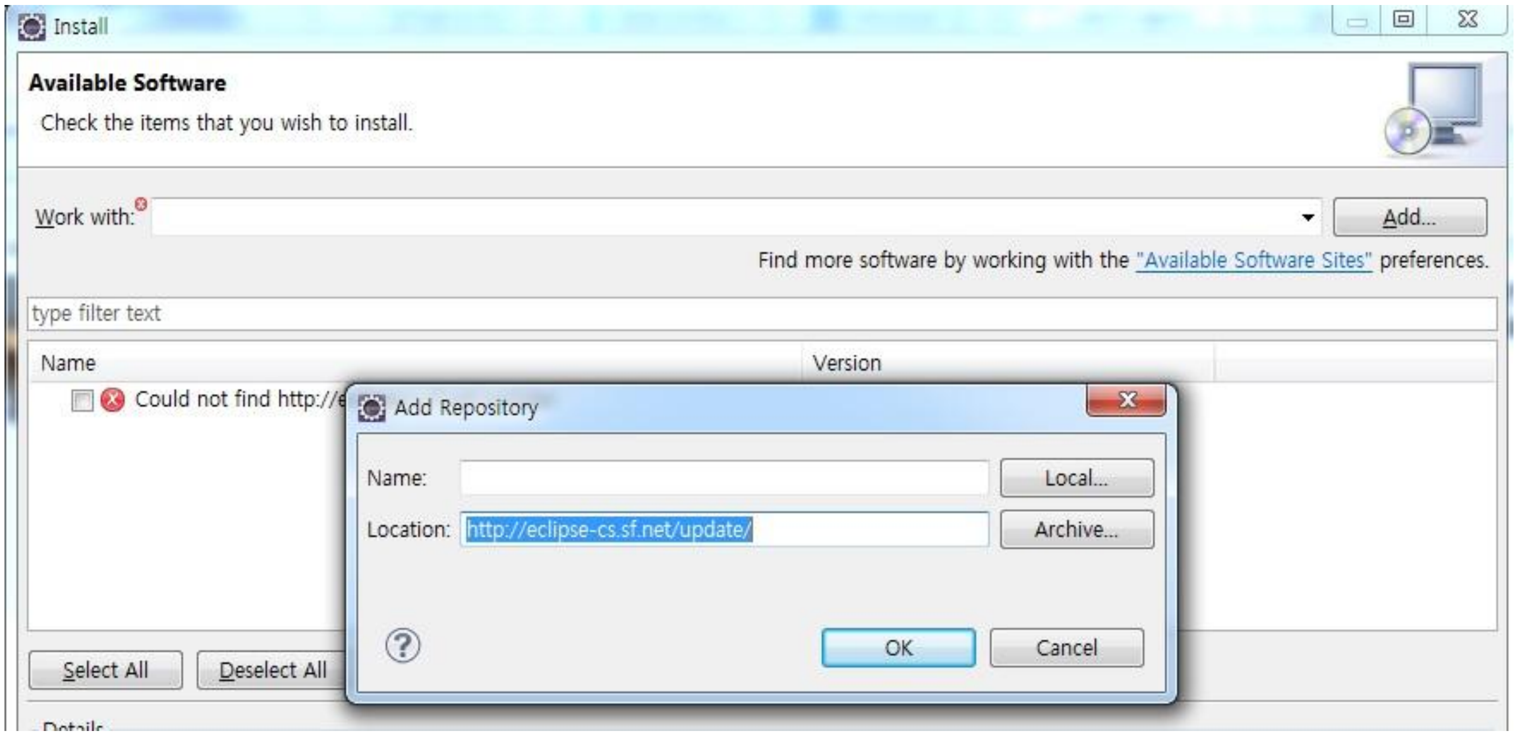
Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
There is no site selected.	

- [Add] Click!

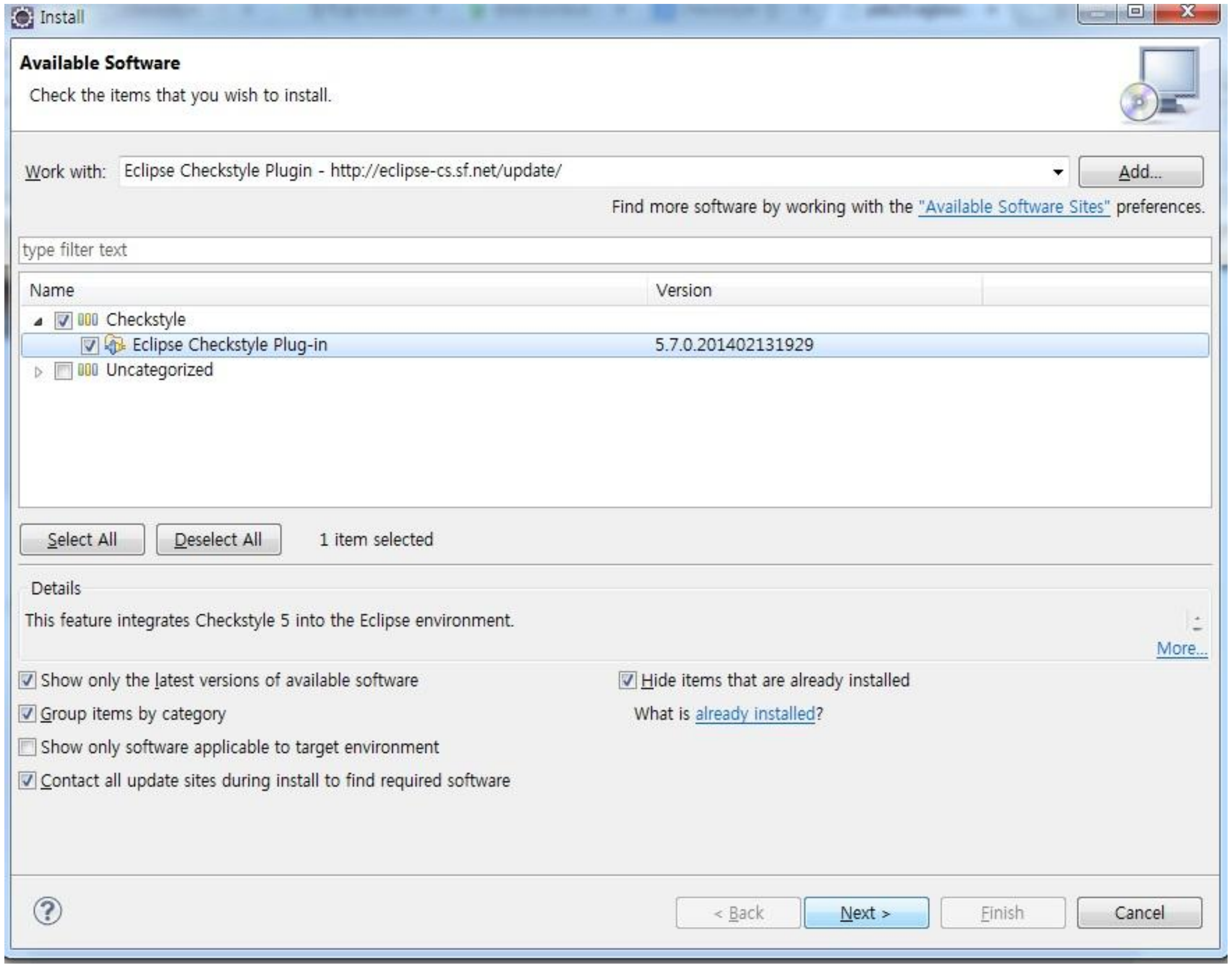
How to install checkstyle(cont.)



Location: http://eclipse-cs.sf.net/update/

- Location :
  [http://eclipse-cs.sf.net/update/]

How to install checkstyle(cont.)


Install

Available Software

Check the items that you wish to install.

Work with: Eclipse Checkstyle Plugin - http://eclipse-cs.sf.net/update/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ Checkstyle	
☑ Eclipse Checkstyle Plug-in	5.7.0.201402131929
☐ Uncategorized	

1 item selected

Details

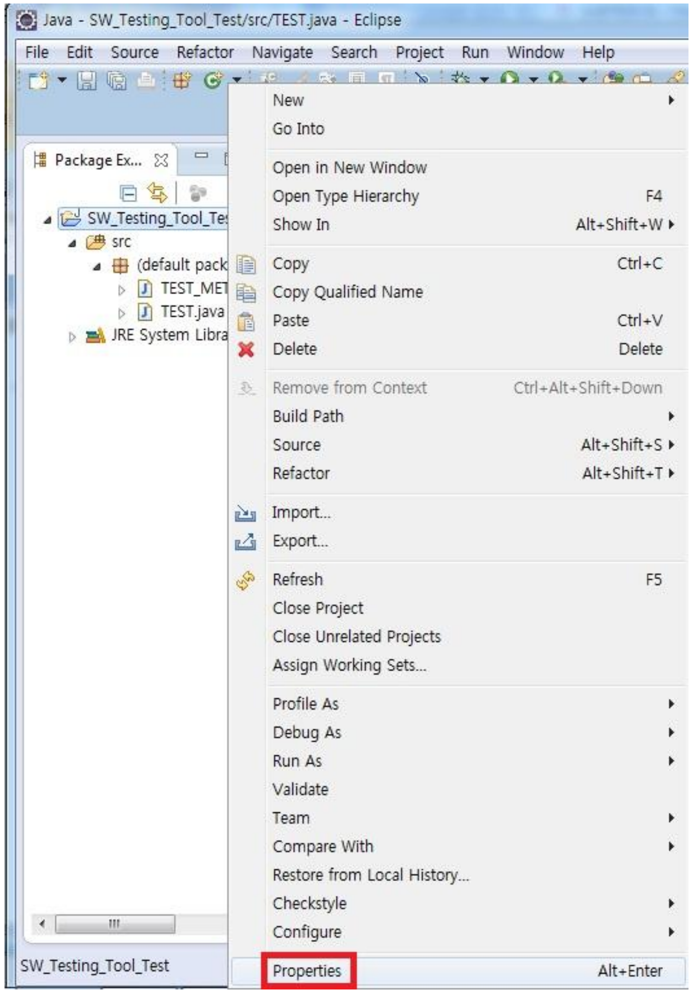
This feature integrates Checkstyle 5 into the Eclipse environment.

- ☑ Show only the latest versions of available software
- ☑ Group items by category
- ☐ Show only software applicable to target environment
- ☑ Contact all update sites during install to find required software
- ☑ Hide items that are already installed

What is already installed?

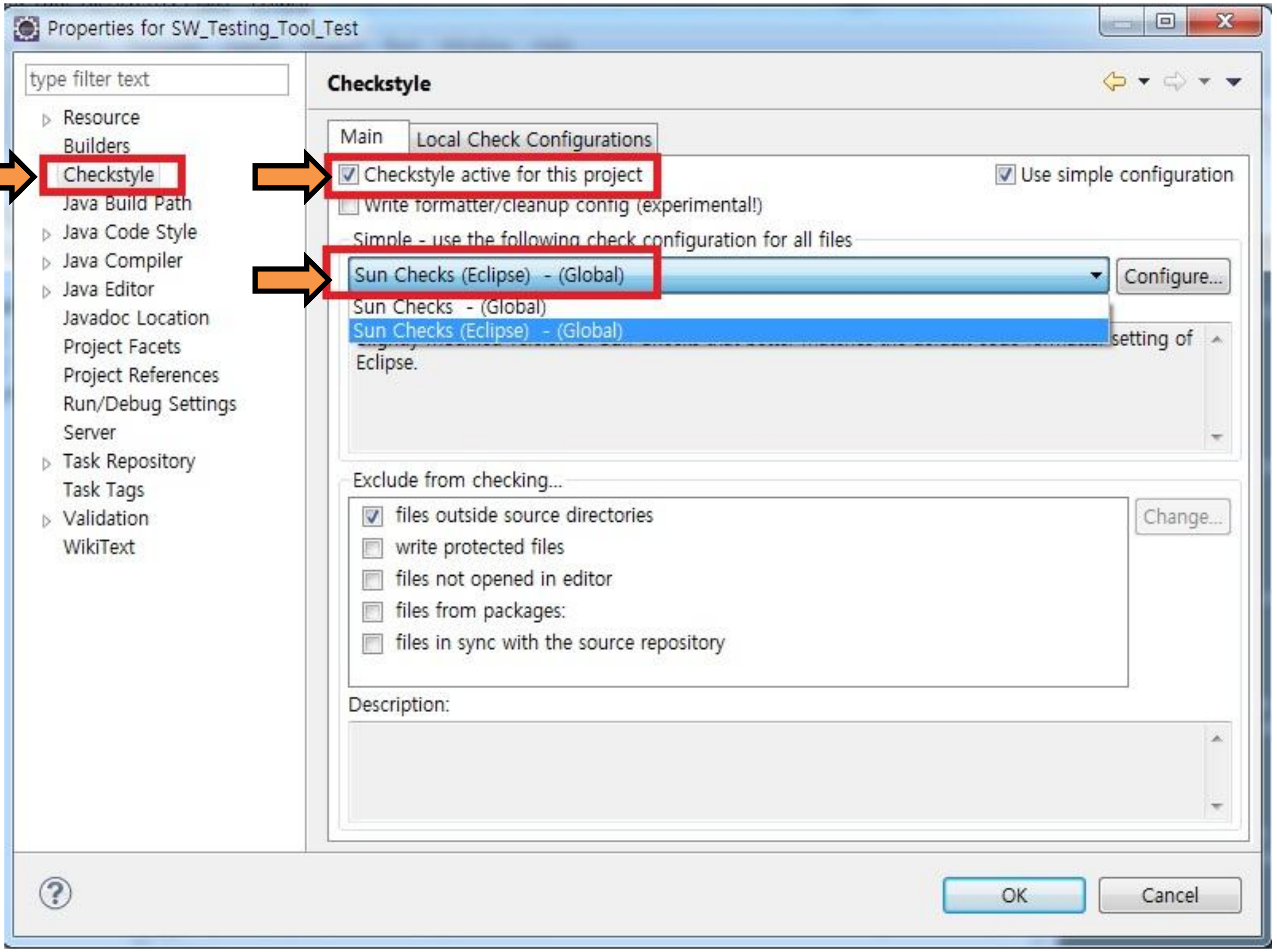
- Check! [Eclipse Checkstyle Plug-in]

How to use checkstyle



- [My project] Right-click! => Properties

How to use checkstyle(cont.)



Properties for SW_Testing_Tool_Test

type filter text

- Resource
- Builders
- Checkstyle
- Java Build Path
- Java Code Style
- Java Compiler
- Java Editor
- Javadoc Location
- Project Facets
- Project References
- Run/Debug Settings
- Server
- Task Repository
- Task Tags
- Validation
- WikiText

Checkstyle

Main | Local Check Configurations

- [✓] Checkstyle active for this project
- [✓] Use simple configuration
- [ ] Write formatter/cleanup config (experimental!)

Simple - use the following check configuration for all files

Sun Checks (Eclipse) - (Global) | Configure...

- Sun Checks - (Global)
- Sun Checks (Eclipse) - (Global)

setting of Eclipse.

Exclude from checking...

- [✓] files outside source directories
- [ ] write protected files
- [ ] files not opened in editor
- [ ] files from packages:
- [ ] files in sync with the source repository

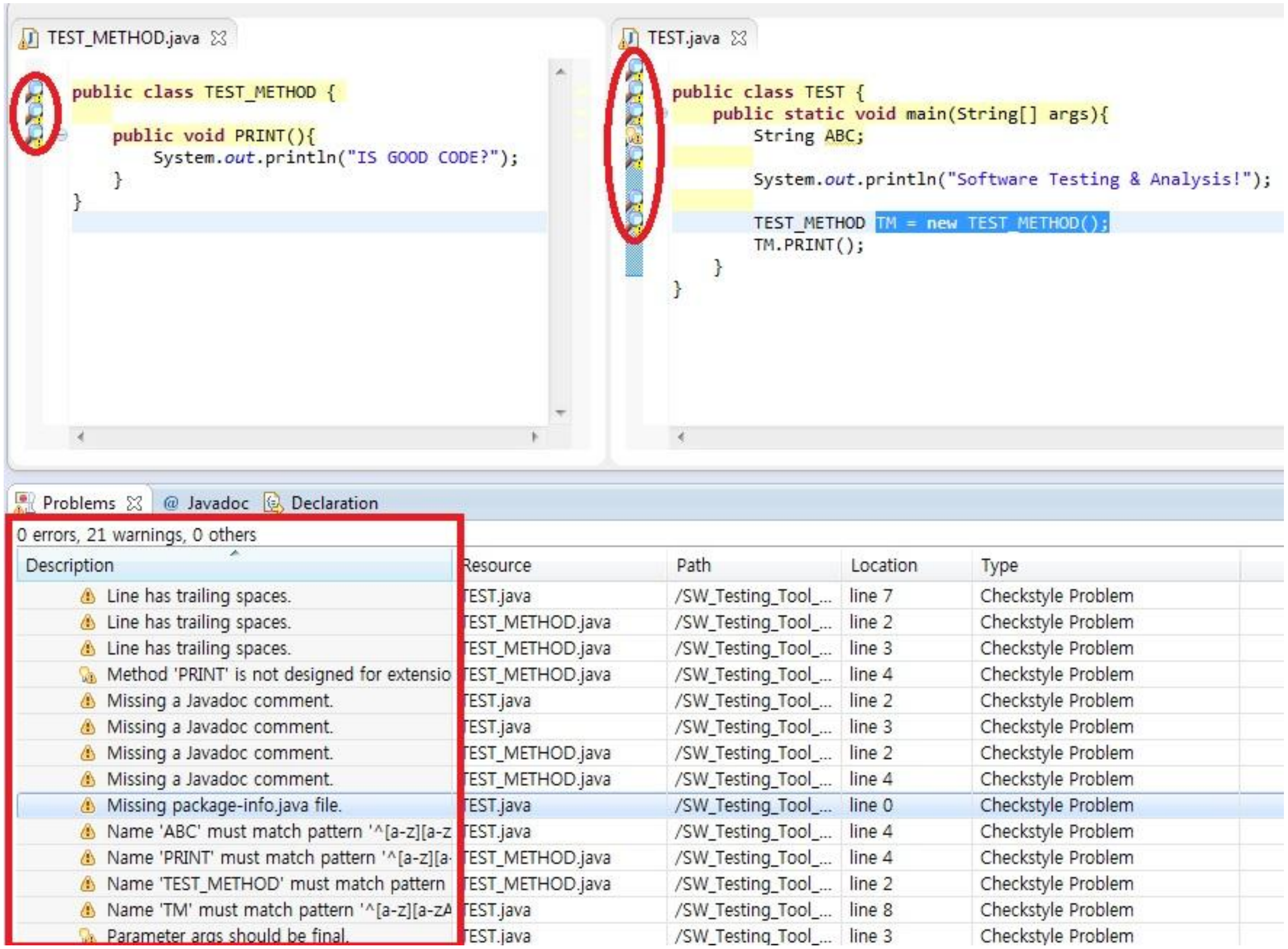
Change...

Description:

OK | Cancel

- User-defined rules also can use.

How to use checkstyle(cont.)



TEST_METHOD.java

```
public class TEST_METHOD {

    public void PRINT(){
        System.out.println("IS GOOD CODE?");
    }
}
```

TEST.java

```
public class TEST {
    public static void main(String[] args){
        String ABC;

        System.out.println("Software Testing & Analysis!");

        TEST_METHOD TM = new TEST_METHOD();
        TM.PRINT();
    }
}
```

Problems | Javadoc | Declaration

0 errors, 21 warnings, 0 others

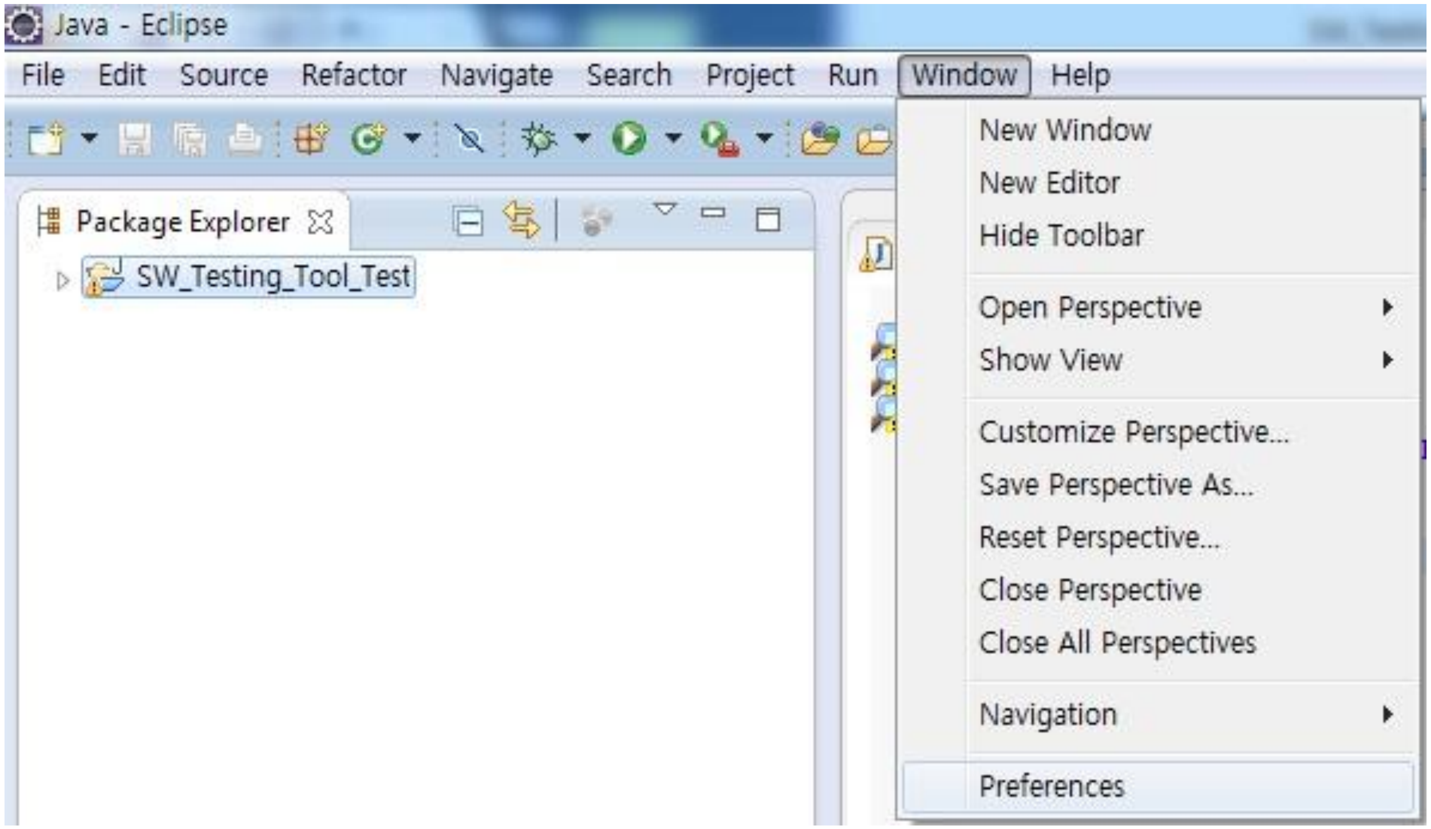
Description	Resource	Path	Location	Type
Line has trailing spaces.	TEST.java	/SW_Testing_Tool_...	line 7	Checkstyle Problem
Line has trailing spaces.	TEST_METHOD.java	/SW_Testing_Tool_...	line 2	Checkstyle Problem
Line has trailing spaces.	TEST_METHOD.java	/SW_Testing_Tool_...	line 3	Checkstyle Problem
Method 'PRINT' is not designed for extensio	TEST_METHOD.java	/SW_Testing_Tool_...	line 4	Checkstyle Problem
Missing a Javadoc comment.	TEST.java	/SW_Testing_Tool_...	line 2	Checkstyle Problem
Missing a Javadoc comment.	TEST.java	/SW_Testing_Tool_...	line 3	Checkstyle Problem
Missing a Javadoc comment.	TEST_METHOD.java	/SW_Testing_Tool_...	line 2	Checkstyle Problem
Missing a Javadoc comment.	TEST_METHOD.java	/SW_Testing_Tool_...	line 4	Checkstyle Problem
Missing package-info.java file.	TEST.java	/SW_Testing_Tool_...	line 0	Checkstyle Problem
Name 'ABC' must match pattern '^[a-z][a-z	TEST.java	/SW_Testing_Tool_...	line 4	Checkstyle Problem
Name 'PRINT' must match pattern '^[a-z][a-	TEST_METHOD.java	/SW_Testing_Tool_...	line 4	Checkstyle Problem
Name 'TEST_METHOD' must match pattern	TEST_METHOD.java	/SW_Testing_Tool_...	line 2	Checkstyle Problem
Name 'TM' must match pattern '^[a-z][a-zA	TEST.java	/SW_Testing_Tool_...	line 8	Checkstyle Problem
Parameter args should be final.	TEST.java	/SW_Testing_Tool_...	line 3	Checkstyle Problem

Checkstyle problem List

항목	원인	회피방법
JavadocPackage	모든 method, class에는 help가 존재해야 한다.	시간상 힘들고, 관리되지 않는 주석은 더욱 큰혼란을 가지고 온다. method의 이름 규칙으로 대신하기로 한다.
NewlineAtEndOfFile	java code의 가장 마지막 줄은 빈공백열로 마쳐져야 한다.	마지막 line에는 항시빈공백을 넣는다.
Translation	Properties file을 이용한 경우, 국가별 번역이 모두 존재해야 한다.	국가별 번역 파일을 따로 만들거나 default 문자열만을 이용한다.
FileLength	java file의 length는2000 line을 넘지 않도록 작성한다.	2000 line이 넘어가는경우, 설계상의 문제가있기 때문에 class를 재정의한다.
FileTabCharacter	java file 내부에 tab 문자열이 있으면 안된다.	tab을 모두 space로 치환해서 사용하도록 한다.
RegexpSingleline	1 line에는 한개의method만이 존재해야 한다.	1 line에 대한 설정을 명확하게 해서 사용하도록 한다.

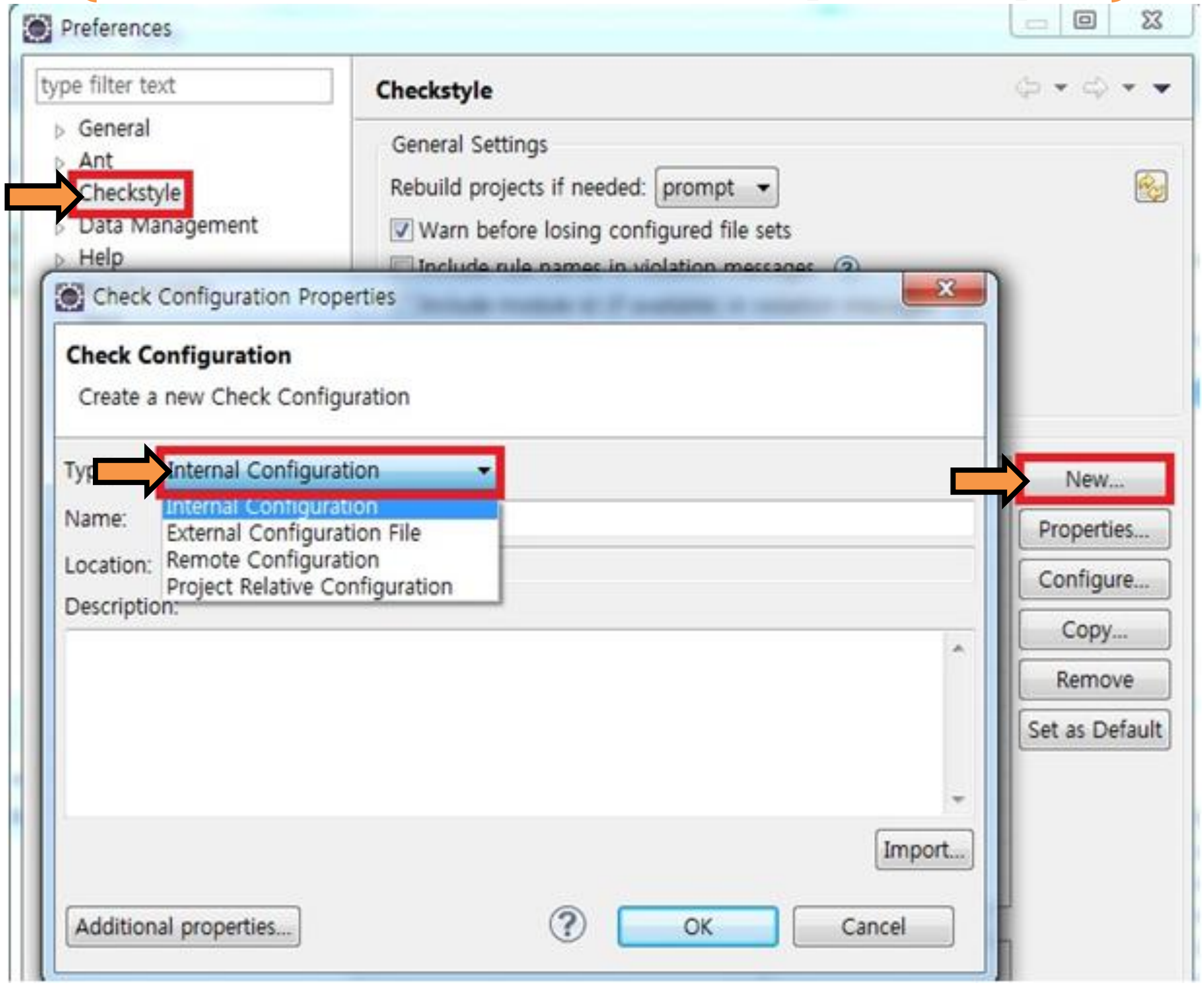
- http://netframework.tistory.com/363

User define rule



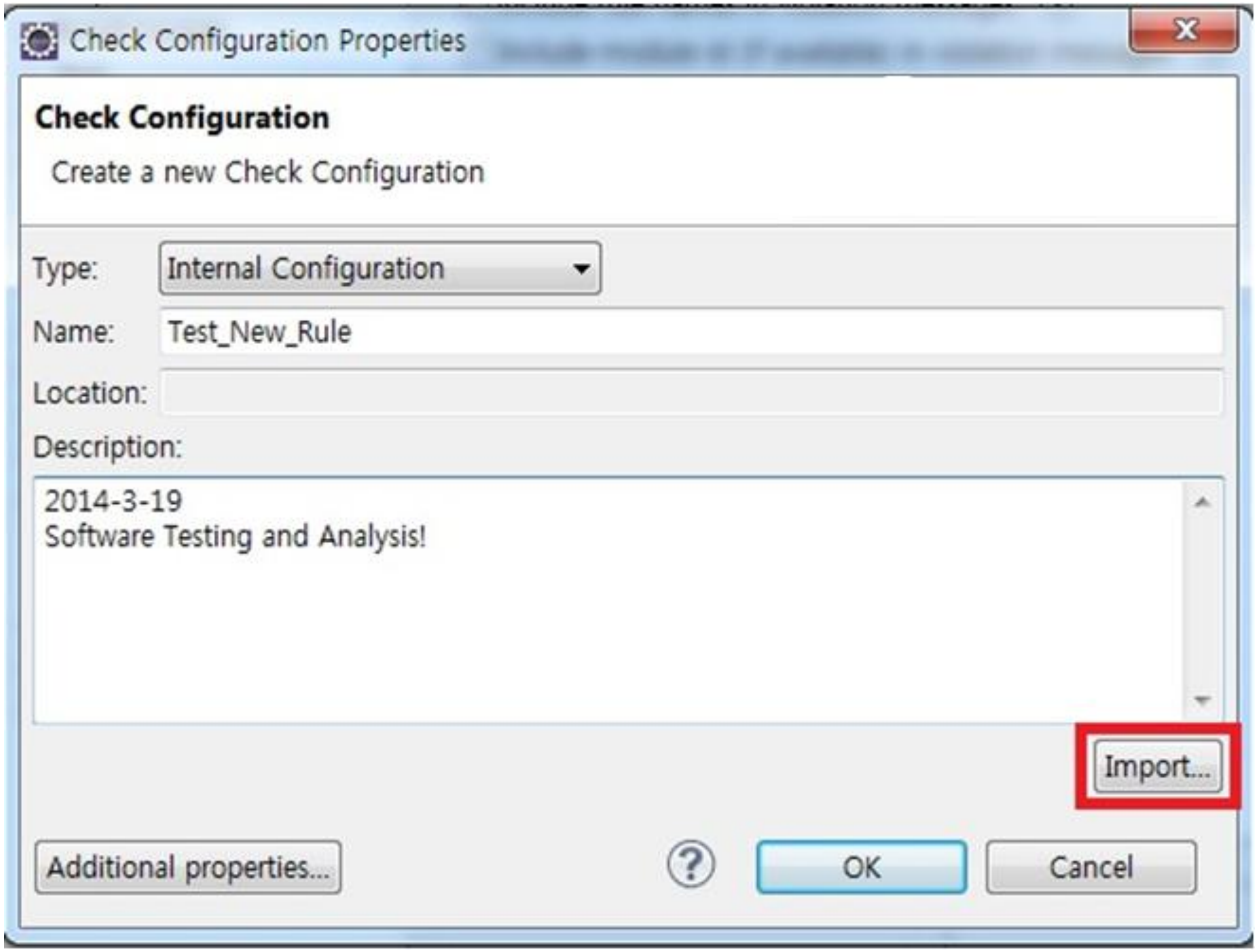
- [Window] – [Preferences] Click!

User define rule(cont.)



- [Checkstyle] – [New] Click!

User define rule(cont.)



Check Configuration

Create a new Check Configuration

Type: Internal Configuration

Name: Test_New_Rule

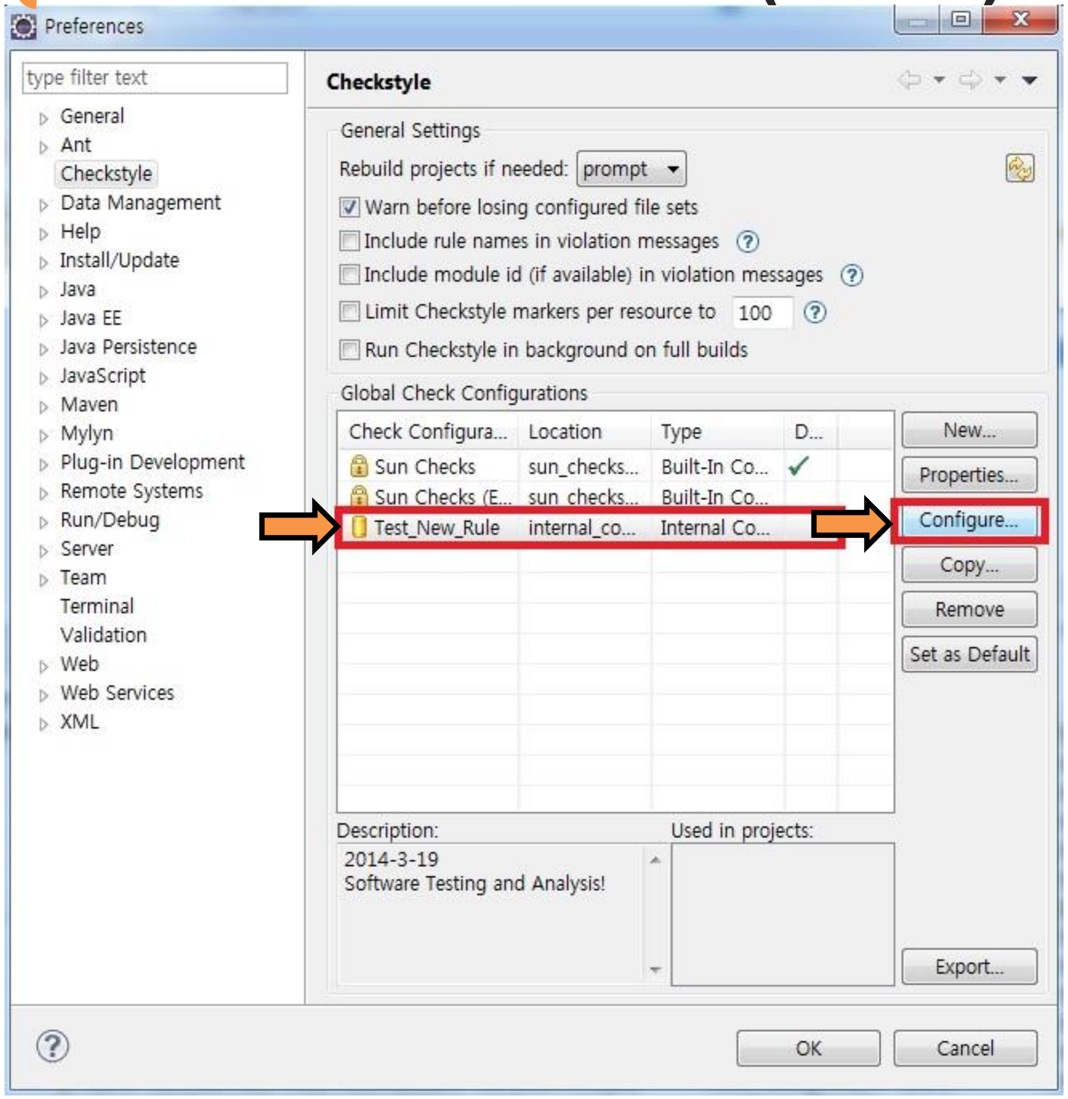
Location:

Description:

2014-3-19
Software Testing and Analysis!

- Can import the other rule. (Extend)

User define rule(cont.)



Checkstyle

General Settings

Rebuild projects if needed: prompt

[✓] Warn before losing configured file sets
[ ] Include rule names in violation messages
[ ] Include module id (if available) in violation messages
[ ] Limit Checkstyle markers per resource to 100
[ ] Run Checkstyle in background on full builds

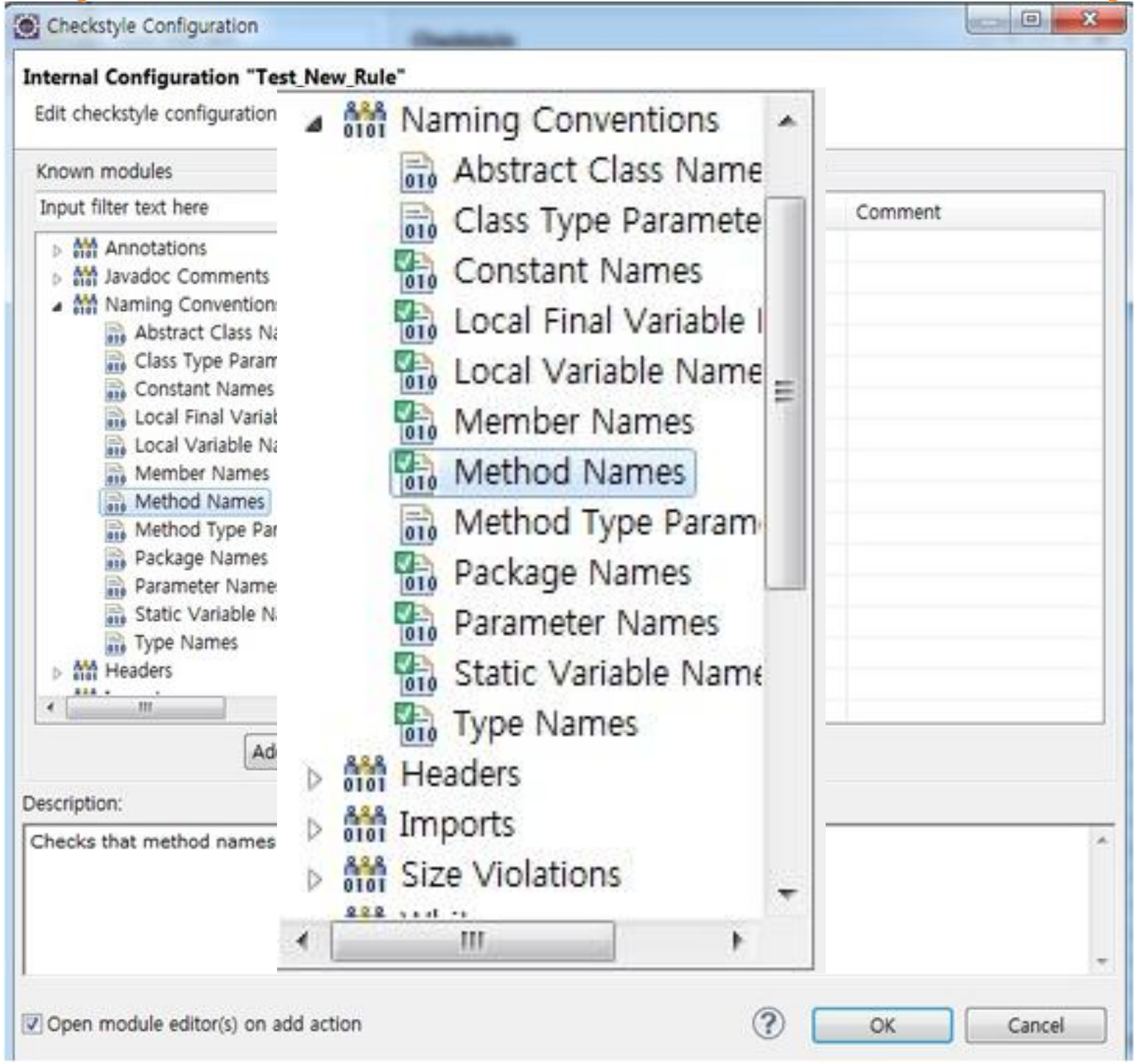
Global Check Configurations

Check Configura...	Location	Type	D...
Sun Checks	sun_checks...	Built-In Co...	✓
Sun Checks (E...	sun_checks...	Built-In Co...	
Test_New_Rule	internal_co...	Internal Co...	

Description:
2014-3-19
Software Testing and Analysis!

Used in projects:

User define rule(cont.)



Checkstyle Configuration

Internal Configuration "Test_New_Rule"

Edit checkstyle configuration

Known modules

Input filter text here

- Annotations
- Javadoc Comments
- Naming Conventions
  - Abstract Class Name
  - Class Type Paramete
  - Constant Names
  - Local Final Variable
  - Local Variable Name
  - Member Names
  - Method Names
  - Method Type Param
  - Package Names
  - Parameter Names
  - Static Variable Name
  - Type Names
- Headers
- Imports
- Size Violations

Comment

Description:

Checks that method names

☑ Open module editor(s) on add action

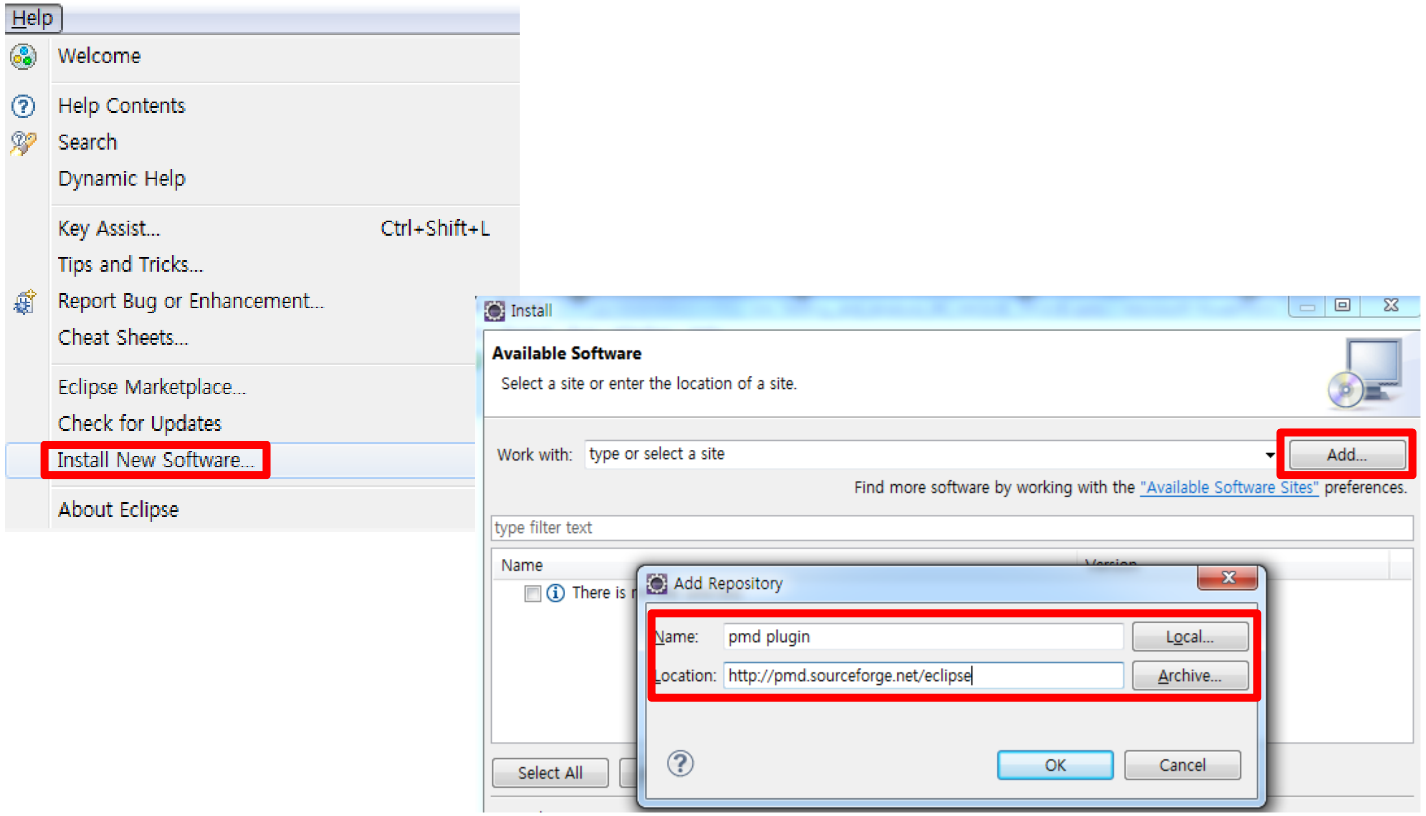
04 PMD

What is PMD?

- PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth.
It supports Java, JavaScript, XML, XSL.

- Additionally it includes CPD, the copy-paste-detector. CPD finds duplicated code in Java, C, C++, C#, PHP, Ruby, Fortran, JavaScript.

How to install PMD in Eclipse?

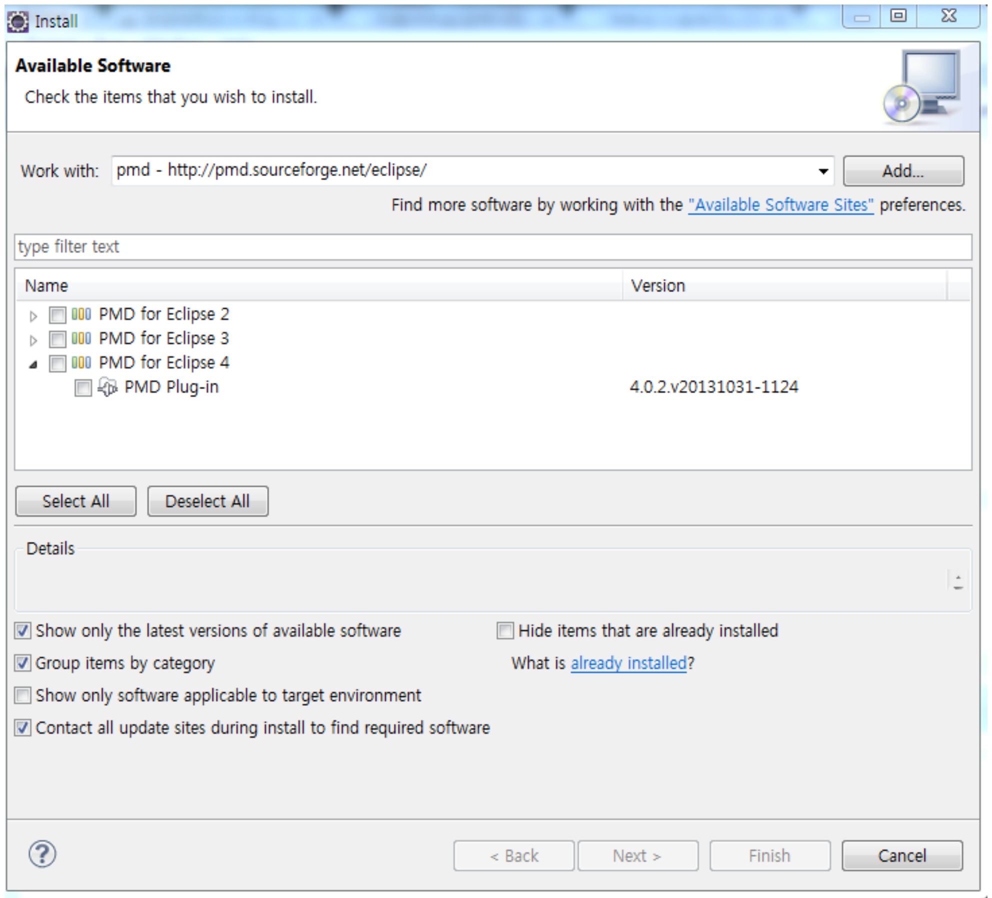


Name: pmd plugin

Location: http://pmd.sourceforge.net/eclipse

- [Help] – [Install new software…]
- http://pmd.sourceforge.net/eclipse

How to install PMD in Eclipse?(cont.)



Install

Available Software

Check the items that you wish to install.

Work with: pmd - http://pmd.sourceforge.net/eclipse/

Find more software by working with the "Available Software Sites" preferences.

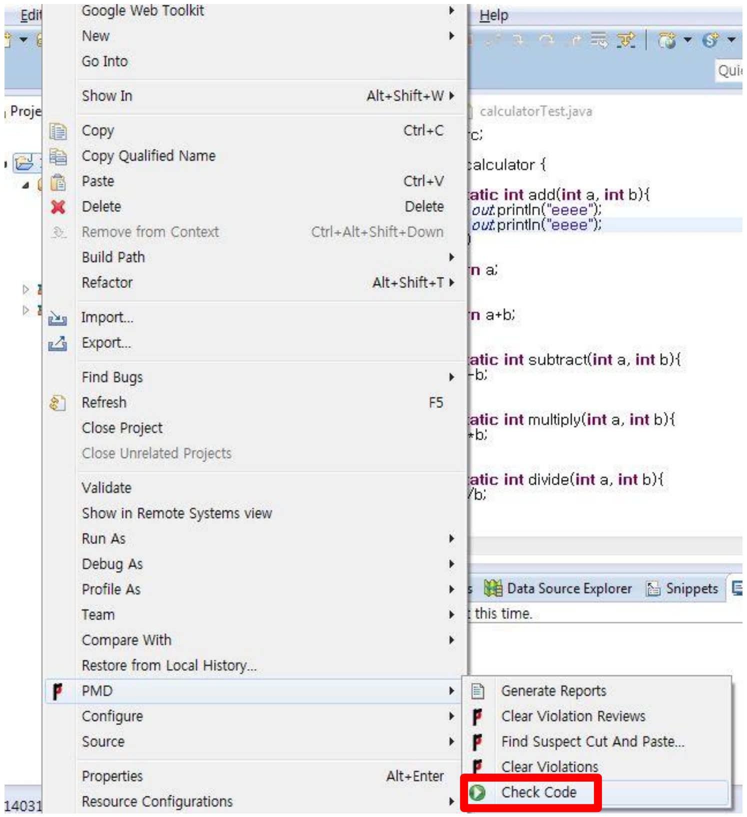
Name	Version
☐ PMD for Eclipse 2	
☐ PMD for Eclipse 3	
☐ PMD for Eclipse 4	
☐ PMD Plug-in	4.0.2.v20131031-1124

Details

☑ Show only the latest versions of available software
☐ Hide items that are already installed
☑ Group items by category
What is already installed?
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software

- select the version.

How to use PMD in Eclipse?



- Right click your project-[PMD]-[Check Code]

01 JUnit

02 Eclipse

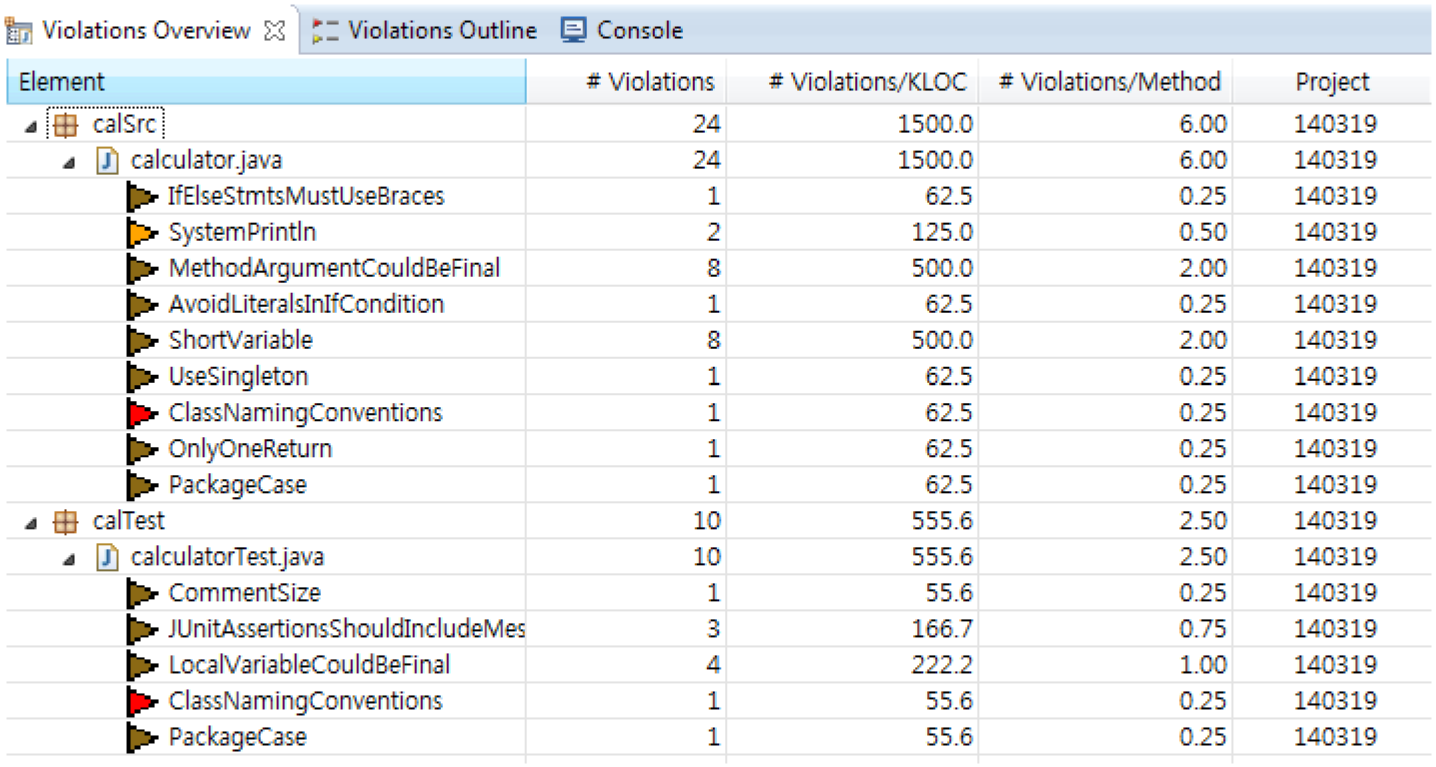
03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

How to use PMD in Eclipse?(cont.)

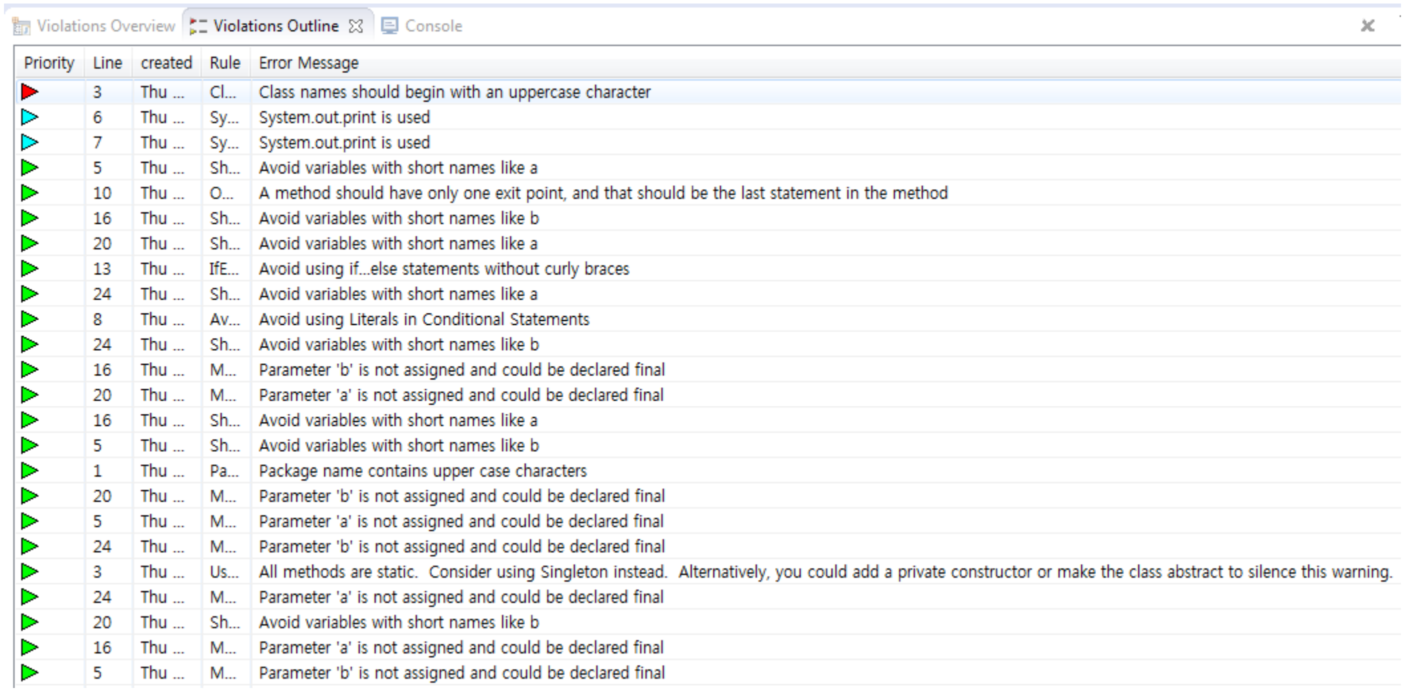


Violations Overview | Violations Outline | Console

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
calSrc	24	1500.0	6.00	140319
calculator.java	24	1500.0	6.00	140319
IfElseStmtsMustUseBraces	1	62.5	0.25	140319
SystemPrintln	2	125.0	0.50	140319
MethodArgumentCouldBeFinal	8	500.0	2.00	140319
AvoidLiteralsInIfCondition	1	62.5	0.25	140319
ShortVariable	8	500.0	2.00	140319
UseSingleton	1	62.5	0.25	140319
ClassNamingConventions	1	62.5	0.25	140319
OnlyOneReturn	1	62.5	0.25	140319
PackageCase	1	62.5	0.25	140319
calTest	10	555.6	2.50	140319
calculatorTest.java	10	555.6	2.50	140319
CommentSize	1	55.6	0.25	140319
JUnitAssertionsShouldIncludeMes	3	166.7	0.75	140319
LocalVariableCouldBeFinal	4	222.2	1.00	140319
ClassNamingConventions	1	55.6	0.25	140319
PackageCase	1	55.6	0.25	140319

- **Violations Overview**

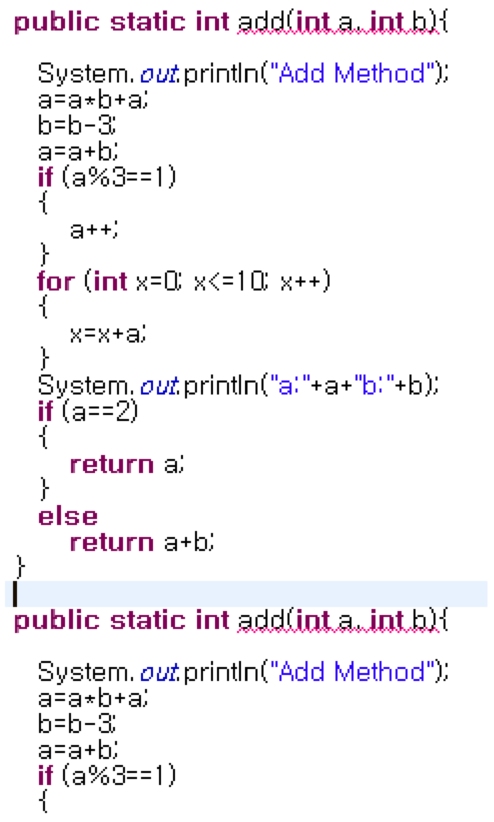
How to use PMD in Eclipse?(cont.)



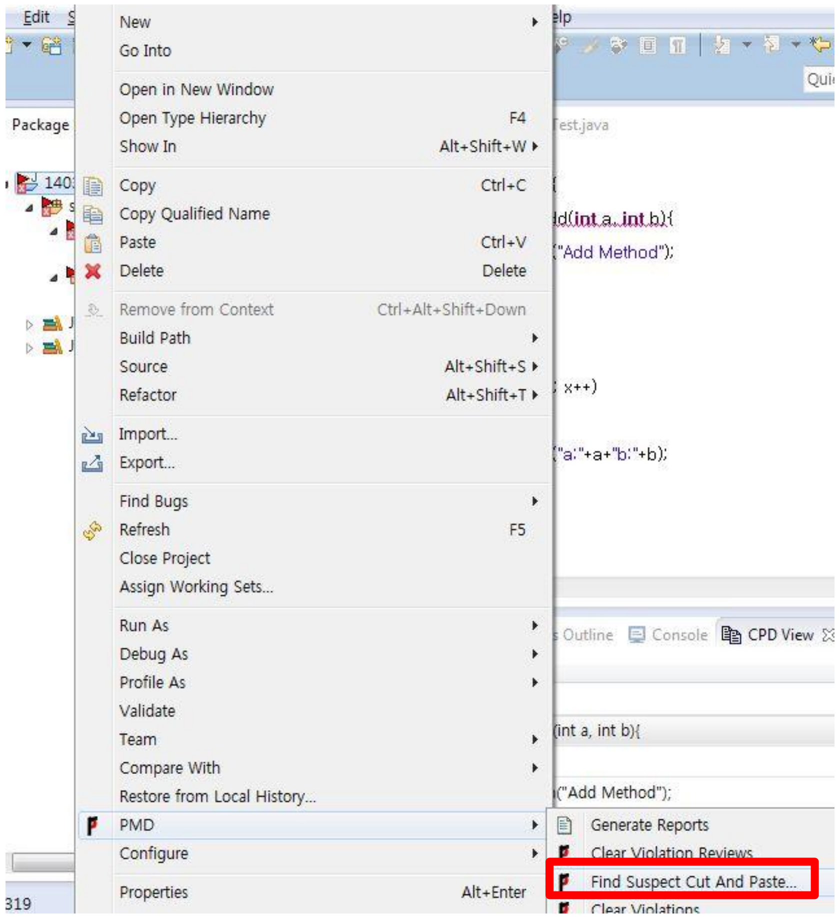
Priority	Line	created	Rule	Error Message
▶	3	Thu ...	Cl...	Class names should begin with an uppercase character
▶	6	Thu ...	Sy...	System.out.print is used
▶	7	Thu ...	Sy...	System.out.print is used
▶	5	Thu ...	Sh...	Avoid variables with short names like a
▶	10	Thu ...	O...	A method should have only one exit point, and that should be the last statement in the method
▶	16	Thu ...	Sh...	Avoid variables with short names like b
▶	20	Thu ...	Sh...	Avoid variables with short names like a
▶	13	Thu ...	IfE...	Avoid using if...else statements without curly braces
▶	24	Thu ...	Sh...	Avoid variables with short names like a
▶	8	Thu ...	Av...	Avoid using Literals in Conditional Statements
▶	24	Thu ...	Sh...	Avoid variables with short names like b
▶	16	Thu ...	M...	Parameter 'b' is not assigned and could be declared final
▶	20	Thu ...	M...	Parameter 'a' is not assigned and could be declared final
▶	16	Thu ...	Sh...	Avoid variables with short names like a
▶	5	Thu ...	Sh...	Avoid variables with short names like b
▶	1	Thu ...	Pa...	Package name contains upper case characters
▶	20	Thu ...	M...	Parameter 'b' is not assigned and could be declared final
▶	5	Thu ...	M...	Parameter 'a' is not assigned and could be declared final
▶	24	Thu ...	M...	Parameter 'b' is not assigned and could be declared final
▶	3	Thu ...	Us...	All methods are static. Consider using Singleton instead. Alternatively, you could add a private constructor or make the class abstract to silence this warning.
▶	24	Thu ...	M...	Parameter 'a' is not assigned and could be declared final
▶	20	Thu ...	Sh...	Avoid variables with short names like b
▶	16	Thu ...	M...	Parameter 'a' is not assigned and could be declared final
▶	5	Thu ...	M...	Parameter 'b' is not assigned and could be declared final

- Violations Outline

How to use PMD in Eclipse?(cont.)

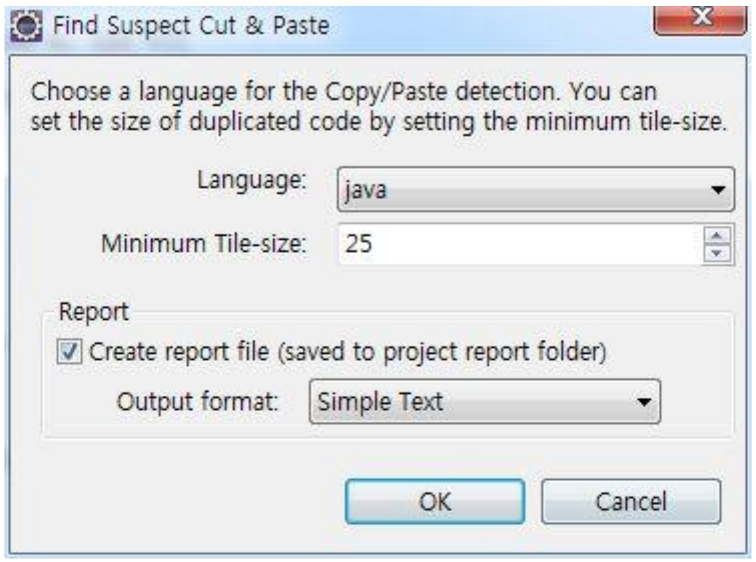


```
public static int add(int a, int b){

    System.out.println("Add Method");
    a=a+b+a;
    b=b-3;
    a=a+b;
    if (a%3==1)
    {
        a++;
    }
    for (int x=0; x<=10; x++)
    {
        x=x+a;
    }
    System.out.println("a:"+a+"b:"+b);
    if (a==2)
    {
        return a;
    }
    else
        return a+b;
}

public static int add(int a, int b){

    System.out.println("Add Method");
    a=a+b+a;
    b=b-3;
    a=a+b;
    if (a%3==1)
    {
```



- right click your project-[PMD]-[Find Suspect Cut And Paste...]

How to use PMD in Eclipse?(cont.)


Find Suspect Cut & Paste

Choose a language for the Copy/Paste detection. You can set the size of duplicated code by setting the minimum tile-size.

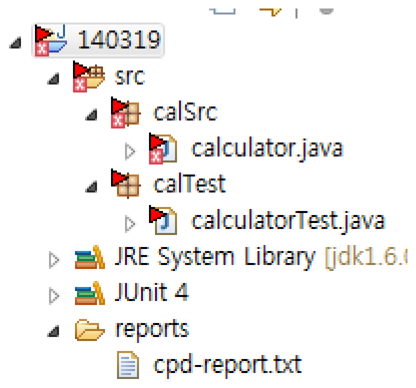
Language: java

Minimum Tile-size: 25

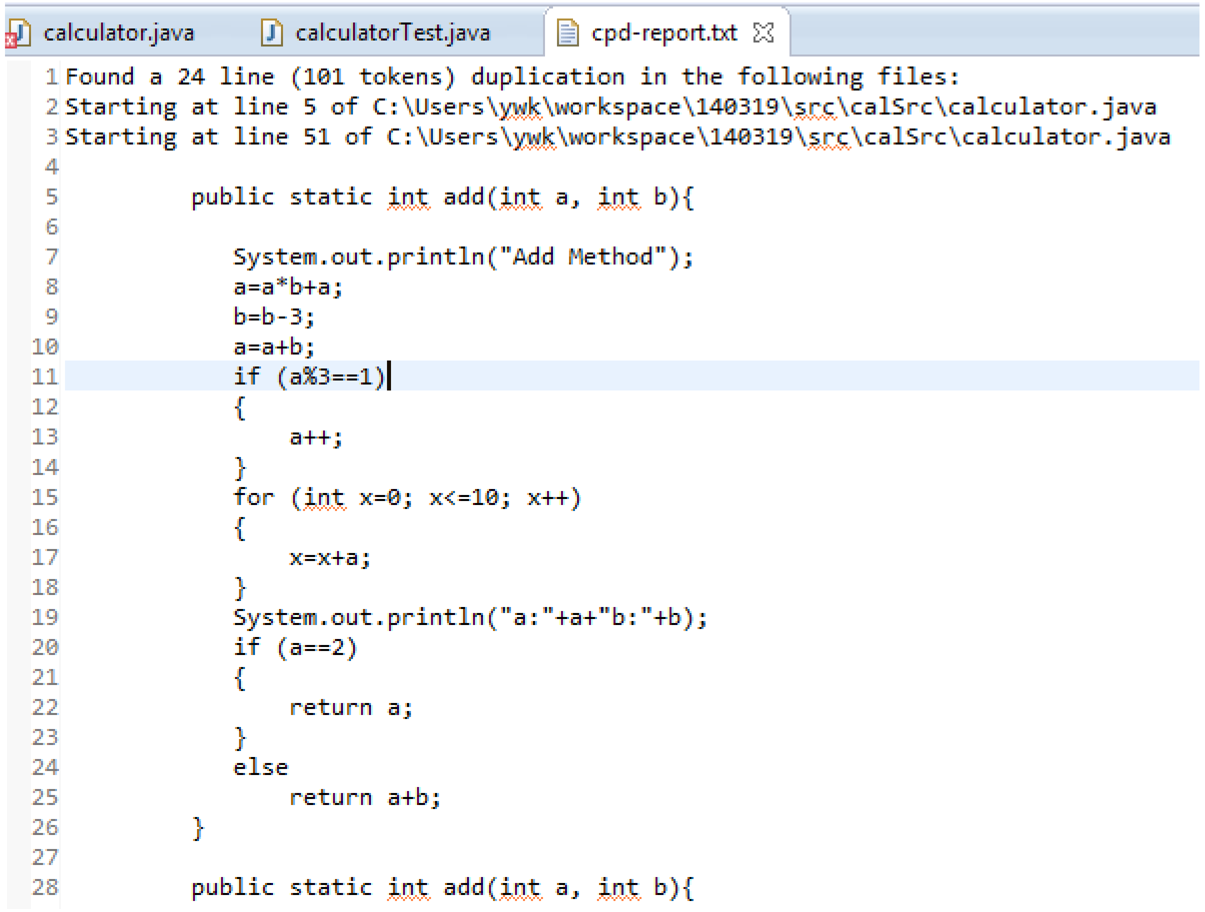
Report

[✓] Create report file (saved to project report folder)

Output format: Simple Text

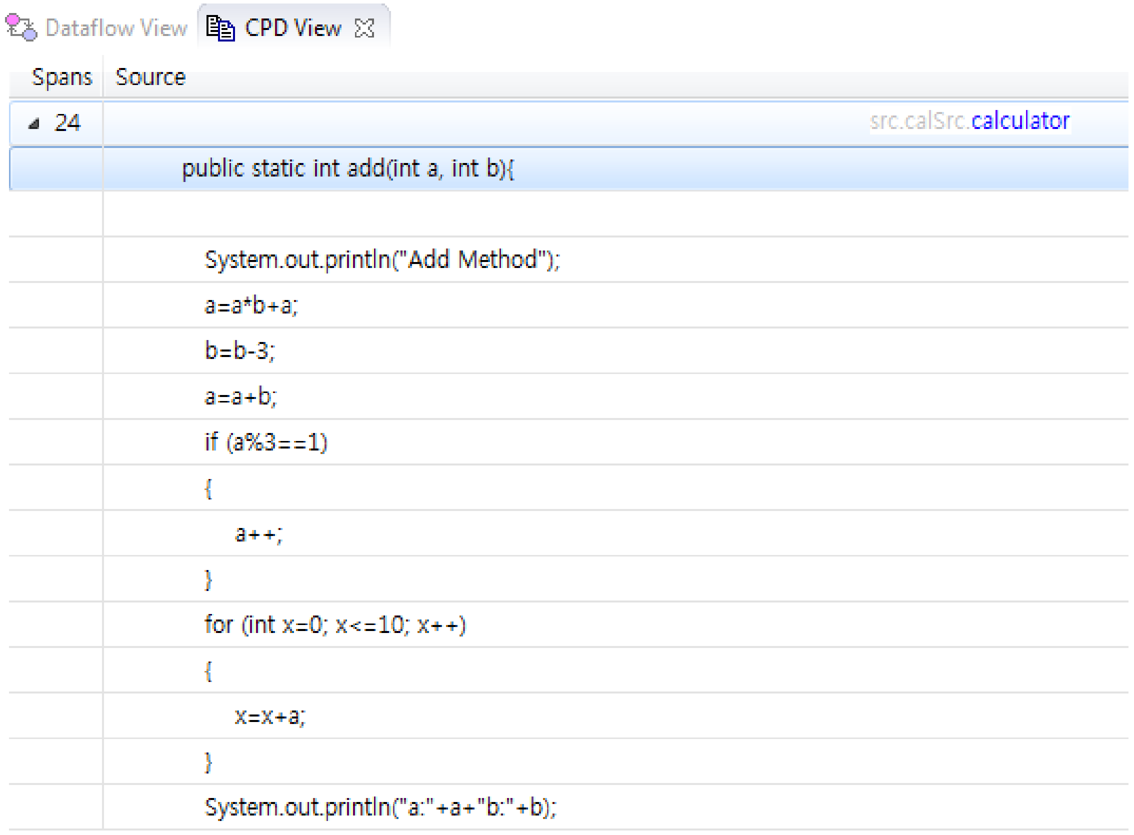

```
140319
  src
    calSrc
      calculator.java
    calTest
      calculatorTest.java
  JRE System Library [jdk1.6.
  JUnit 4
  reports
    cpd-report.txt
```

How to use PMD in Eclipse?(cont.)



```
Found a 24 line (101 tokens) duplication in the following files:
Starting at line 5 of C:\Users\ywk\workspace\140319\src\calSrc\calculator.java
Starting at line 51 of C:\Users\ywk\workspace\140319\src\calSrc\calculator.java

        public static int add(int a, int b){

            System.out.println("Add Method");
            a=a*b+a;
            b=b-3;
            a=a+b;
            if (a%3==1)
            {
                a++;
            }
            for (int x=0; x<=10; x++)
            {
                x=x+a;
            }
            System.out.println("a:"+a+"b:"+b);
            if (a==2)
            {
                return a;
            }
            else
                return a+b;
        }

        public static int add(int a, int b){
```

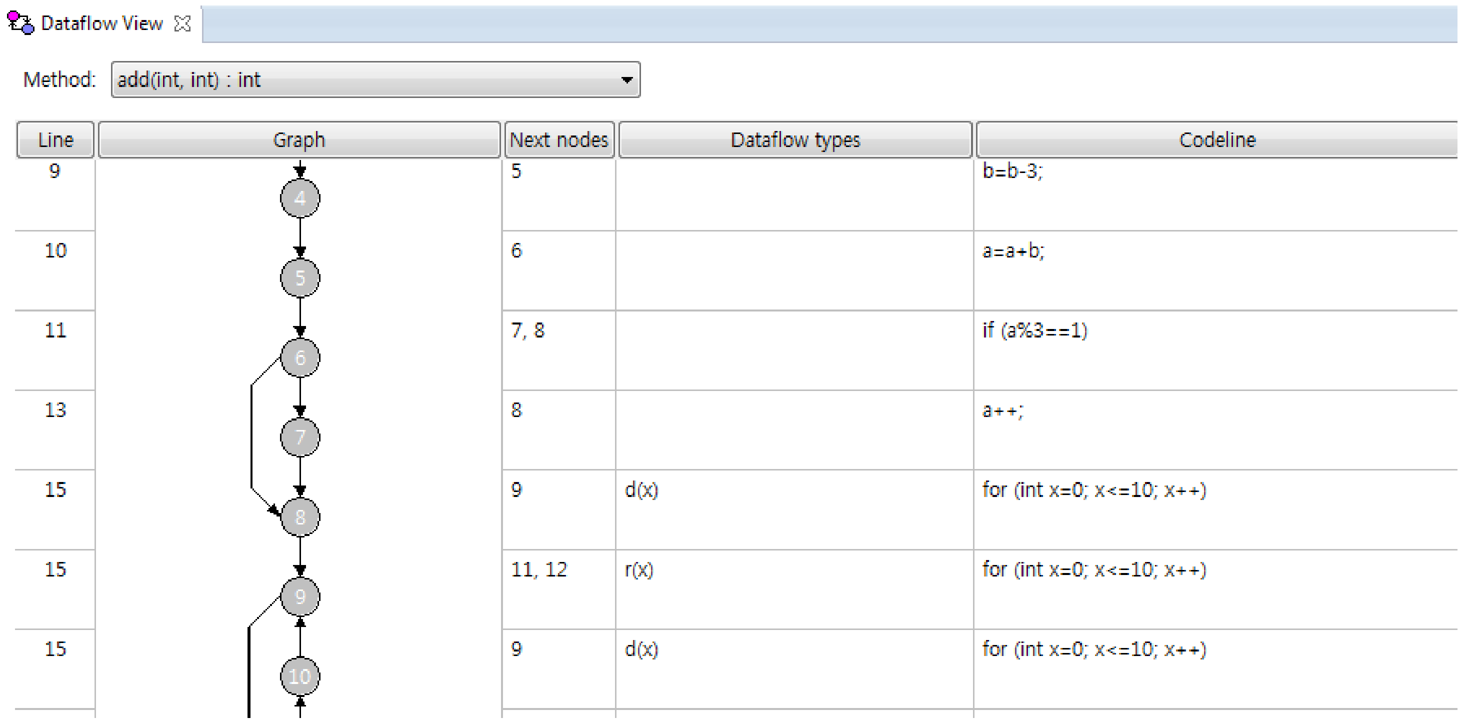
How to use PMD in Eclipse?(cont.)



```
public static int add(int a, int b){

    System.out.println("Add Method");
    a=a*b+a;
    b=b-3;
    a=a+b;
    if (a%3==1)
    {
        a++;
    }
    for (int x=0; x<=10; x++)
    {
        x=x+a;
    }
    System.out.println("a:"+a+"b:"+b);
```

- CPD View

How to use PMD in Eclipse?(cont.)



Dataflow View

Method: add(int, int) : int

Line	Graph	Next nodes	Dataflow types	Codeline
9	4	5		b=b-3;
10	5	6		a=a+b;
11	6	7, 8		if (a%3==1)
13	7	8		a++;
15	8	9	d(x)	for (int x=0; x<=10; x++)
15	9	11, 12	r(x)	for (int x=0; x<=10; x++)
15	10	9	d(x)	for (int x=0; x<=10; x++)

- DataFlow View in each method.

01 JUnit

02 Eclipse

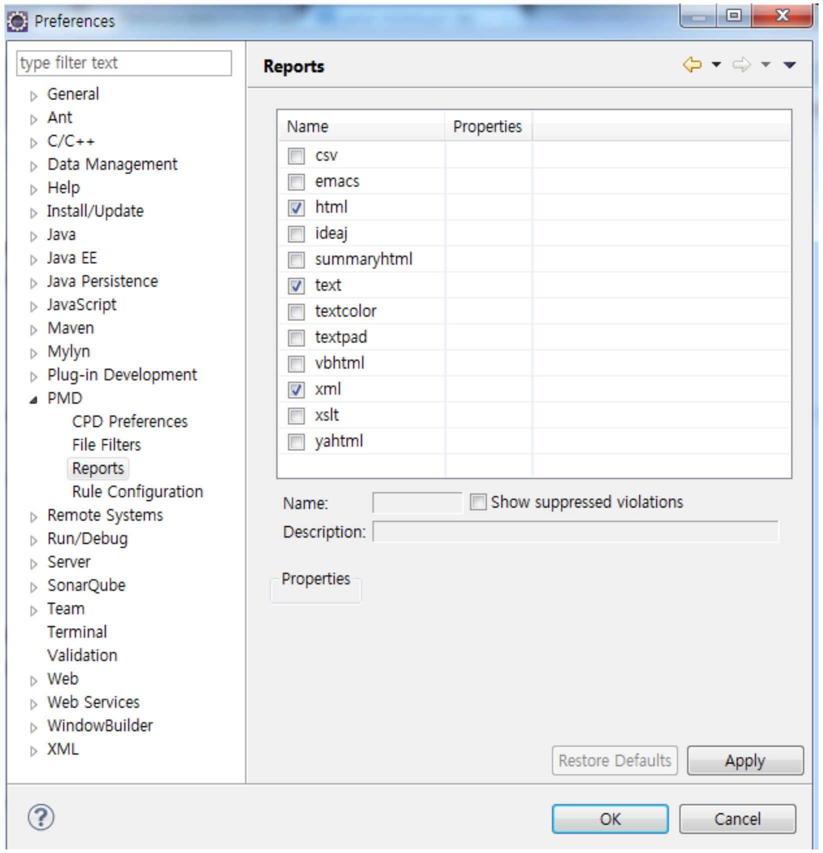
03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

How to report in Eclipse?



Reports

Name	Properties
☐ csv	
☐ emacs	
☑ html	
☐ ideaj	
☐ summaryhtml	
☑ text	
☐ textcolor	
☐ textpad	
☐ vbhtml	
☑ xml	
☐ xslt	
☐ yahtml	

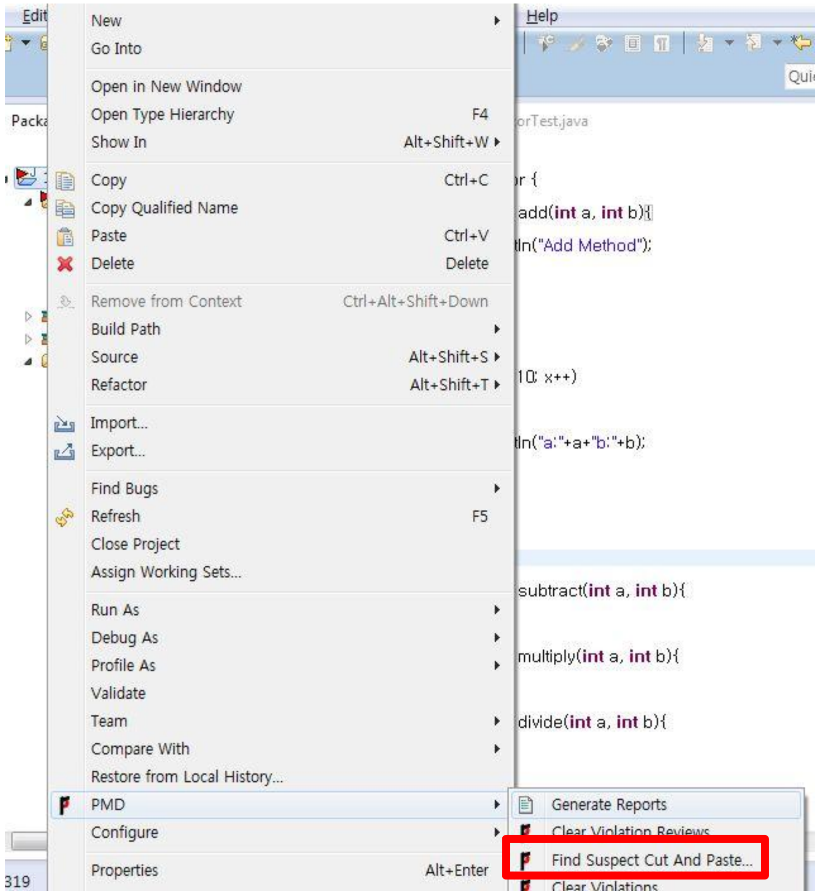
Name: ☐ Show suppressed violations

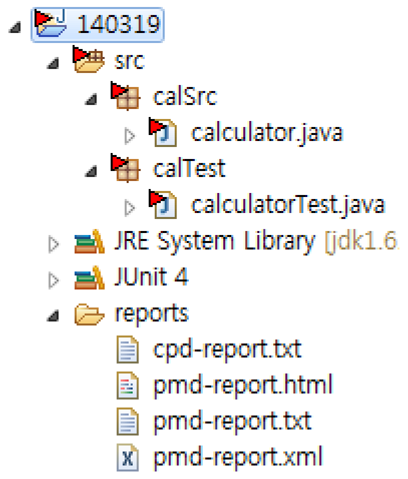
Description:

Properties

- [Windows]-[Preferences]-[PMD]-[Reports]

How to report in Eclipse?(cont.)





- right click to your project-[PMD]-[Generate Reports]

How to report in Eclipse?(cont.)

01 JUnit

02 Eclipse

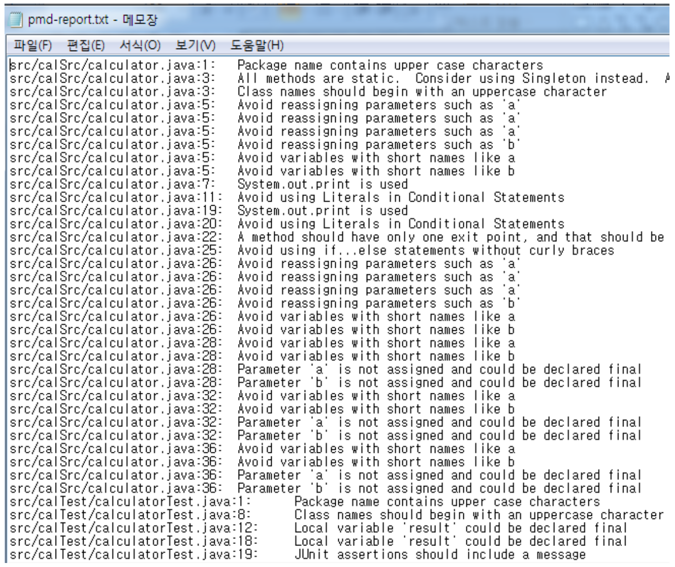
03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

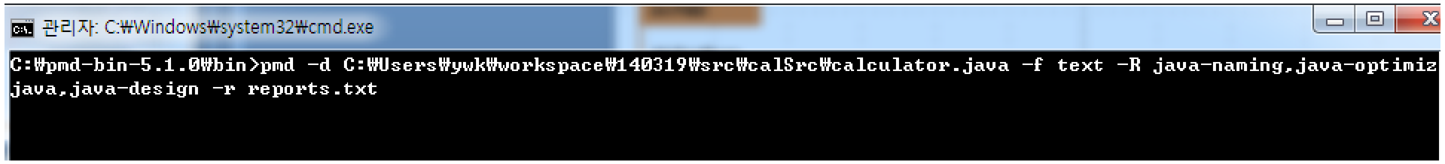
```
<?xml version="1.0" encoding="UTF-8" ?>
<pmd version="5.0.5" timestamp="2014-03-20T15:19:40.966">
- <file name="src/calSrc/calculator.java">
    <violation beginline="1" endline="1" begincolumn="0" endcolumn="0" rule="PackageCase" ruleset="Naming" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/naming.html#PackageCase" priority="3">Package name contains upper case characters</violation>
    <violation beginline="3" endline="86" begincolumn="0" endcolumn="0" rule="UseSingleton" ruleset="Design" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/design.html#UseSingleton" priority="3">All methods are static. Consider using Singleton instead.
      Alternatively, you could add a private constructor or make the class abstract to silence this warning.</violation>
    <violation beginline="3" endline="86" begincolumn="0" endcolumn="0" rule="ClassNamingConventions" ruleset="Naming" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/naming.html#ClassNamingConventions" priority="1">Class names should begin with an uppercase
      character</violation>
    <violation beginline="5" endline="5" begincolumn="0" endcolumn="0" rule="AvoidReassigningParameters" ruleset="Design" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/design.html#AvoidReassigningParameters" priority="2">Avoid reassigning parameters such as
      'a'</violation>
    <violation beginline="5" endline="5" begincolumn="0" endcolumn="0" rule="AvoidReassigningParameters" ruleset="Design" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/design.html#AvoidReassigningParameters" priority="2">Avoid reassigning parameters such as
      'a'</violation>
    <violation beginline="5" endline="5" begincolumn="0" endcolumn="0" rule="AvoidReassigningParameters" ruleset="Design" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/design.html#AvoidReassigningParameters" priority="2">Avoid reassigning parameters such as
      'a'</violation>
    <violation beginline="5" endline="5" begincolumn="0" endcolumn="0" rule="AvoidReassigningParameters" ruleset="Design" package="calSrc" class="calculator"
      externalInfoUrl="http://pmd.sourceforge.net/pmd-5.0.5/rules/java/design.html#AvoidReassigningParameters" priority="2">Avoid reassigning parameters such as
      'b'</violation>
```



```
src/calSrc/calculator.java:1:   Package name contains upper case characters
src/calSrc/calculator.java:3:   All methods are static.  Consider using Singleton instead.
src/calSrc/calculator.java:3:   Class names should begin with an uppercase character
src/calSrc/calculator.java:5:   Avoid reassigning parameters such as 'a'
src/calSrc/calculator.java:5:   Avoid reassigning parameters such as 'a'
src/calSrc/calculator.java:5:   Avoid reassigning parameters such as 'a'
src/calSrc/calculator.java:5:   Avoid reassigning parameters such as 'b'
src/calSrc/calculator.java:5:   Avoid variables with short names like a
src/calSrc/calculator.java:5:   Avoid variables with short names like b
src/calSrc/calculator.java:7:   System.out.print is used
src/calSrc/calculator.java:11:  Avoid using Literals in Conditional Statements
src/calSrc/calculator.java:19:  System.out.print is used
src/calSrc/calculator.java:20:  Avoid using Literals in Conditional Statements
src/calSrc/calculator.java:22:  A method should have only one exit point, and that should be
src/calSrc/calculator.java:25:  Avoid using if...else statements without curly braces
src/calSrc/calculator.java:26:  Avoid reassigning parameters such as 'a'
src/calSrc/calculator.java:26:  Avoid reassigning parameters such as 'a'
src/calSrc/calculator.java:26:  Avoid reassigning parameters such as 'a'
src/calSrc/calculator.java:26:  Avoid reassigning parameters such as 'b'
src/calSrc/calculator.java:26:  Avoid variables with short names like a
src/calSrc/calculator.java:26:  Avoid variables with short names like b
src/calSrc/calculator.java:28:  Avoid variables with short names like a
src/calSrc/calculator.java:28:  Avoid variables with short names like b
src/calSrc/calculator.java:28:  Parameter 'a' is not assigned and could be declared final
src/calSrc/calculator.java:28:  Parameter 'b' is not assigned and could be declared final
src/calSrc/calculator.java:32:  Avoid variables with short names like a
src/calSrc/calculator.java:32:  Avoid variables with short names like b
src/calSrc/calculator.java:32:  Parameter 'a' is not assigned and could be declared final
src/calSrc/calculator.java:32:  Parameter 'b' is not assigned and could be declared final
src/calSrc/calculator.java:36:  Avoid variables with short names like a
src/calSrc/calculator.java:36:  Avoid variables with short names like b
src/calSrc/calculator.java:36:  Parameter 'a' is not assigned and could be declared final
src/calSrc/calculator.java:36:  Parameter 'b' is not assigned and could be declared final
src/calTest/calculatorTest.java:1:      Package name contains upper case characters
src/calTest/calculatorTest.java:8:      Class names should begin with an uppercase character
src/calTest/calculatorTest.java:12:     Local variable 'result' could be declared final
src/calTest/calculatorTest.java:18:     Local variable 'result' could be declared final
src/calTest/calculatorTest.java:19:     JUnit assertions should include a message
```

Can I use PMD without Eclipse?

- download latest version(5.1) http://pmd.sourceforge.net
- unzip the file



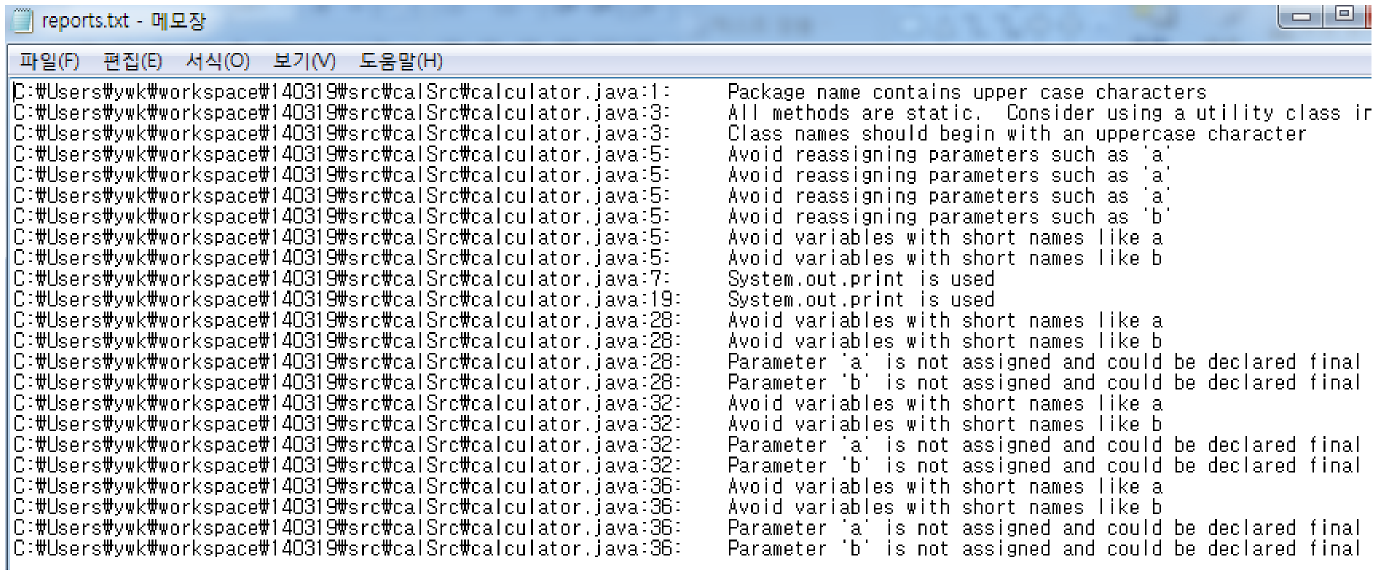
```
관리자: C:₩Windows₩system32₩cmd.exe
C:₩pmd-bin-5.1.0₩bin>pmd -d C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java -f text -R java-naming,java-optimiz
java,java-design -r reports.txt
```

- move to C:\...\pmd-bin-5.1.0\bin

pmd –d [source dir] –f [format type] –R [rulesets] -r [report dir]

- format type : text, html, xml, nicehtml
- default ruleset directory : pmd-bin-5.1.0\docs\rules

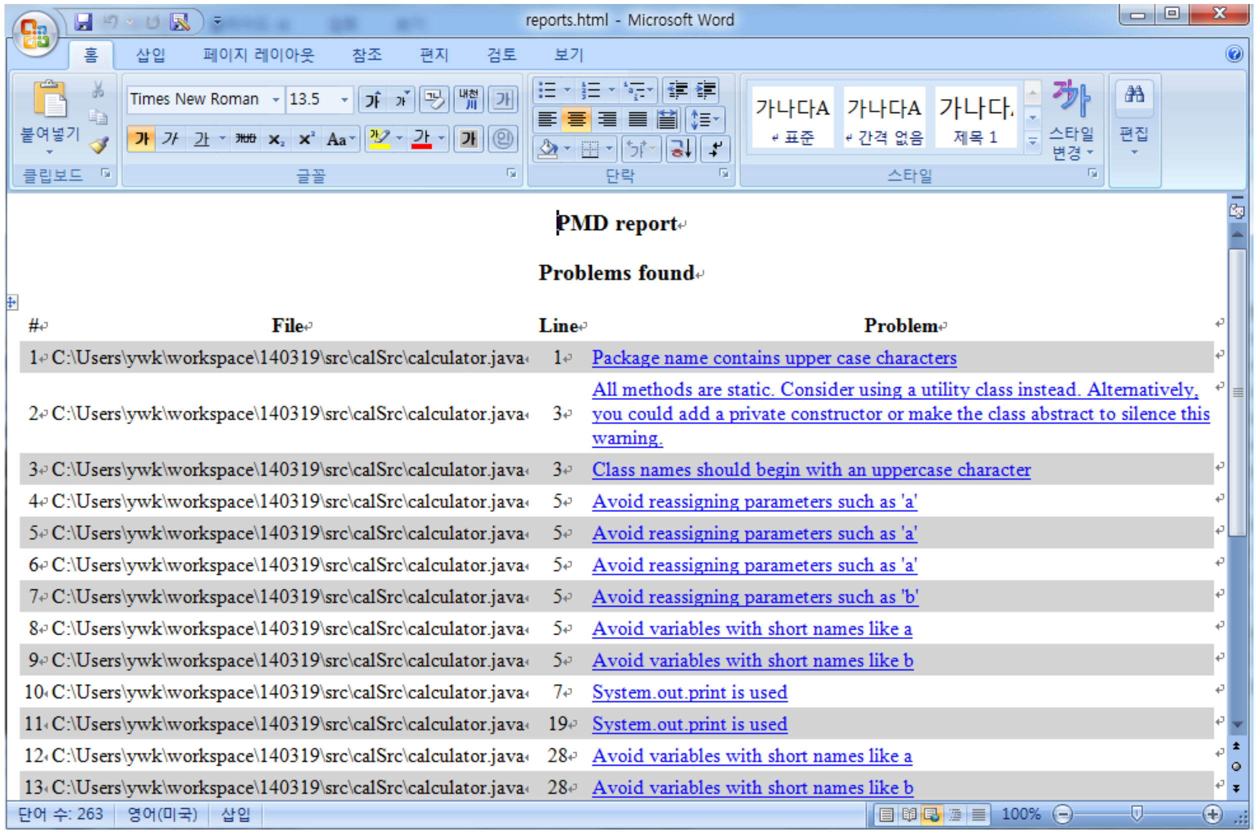
How to use PMD without Eclipse?(cont.)



```
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:1:	Package name contains upper case characters
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:3:	All methods are static.  Consider using a utility class ir
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:3:	Class names should begin with an uppercase character
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:5:	Avoid reassigning parameters such as 'a'
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:5:	Avoid reassigning parameters such as 'a'
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:5:	Avoid reassigning parameters such as 'a'
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:5:	Avoid reassigning parameters such as 'b'
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:5:	Avoid variables with short names like a
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:5:	Avoid variables with short names like b
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:7:	System.out.print is used
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:19:	System.out.print is used
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:28:	Avoid variables with short names like a
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:28:	Avoid variables with short names like b
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:28:	Parameter 'a' is not assigned and could be declared final
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:28:	Parameter 'b' is not assigned and could be declared final
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:32:	Avoid variables with short names like a
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:32:	Avoid variables with short names like b
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:32:	Parameter 'a' is not assigned and could be declared final
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:32:	Parameter 'b' is not assigned and could be declared final
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:36:	Avoid variables with short names like a
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:36:	Avoid variables with short names like b
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:36:	Parameter 'a' is not assigned and could be declared final
C:₩Users₩ywk₩workspace₩140319₩src₩calSrc₩calculator.java:36:	Parameter 'b' is not assigned and could be declared final
```

- 01 JUnit
- 02 Eclipse
- 03 CheckStyle
- **04 PMD**
- 05 FindBugs
- 06 SONAR

How to use PMD without Eclipse?(cont.)



PMD report

Problems found

#	File	Line	Problem
1	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	1	Package name contains upper case characters
2	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	3	All methods are static. Consider using a utility class instead. Alternatively, you could add a private constructor or make the class abstract to silence this warning.
3	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	3	Class names should begin with an uppercase character
4	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	5	Avoid reassigning parameters such as 'a'
5	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	5	Avoid reassigning parameters such as 'a'
6	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	5	Avoid reassigning parameters such as 'a'
7	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	5	Avoid reassigning parameters such as 'b'
8	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	5	Avoid variables with short names like a
9	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	5	Avoid variables with short names like b
10	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	7	System.out.print is used
11	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	19	System.out.print is used
12	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	28	Avoid variables with short names like a
13	C:\Users\ywk\workspace\140319\src\calSrc\calculator.java	28	Avoid variables with short names like b

01 JUnit

02 Eclipse

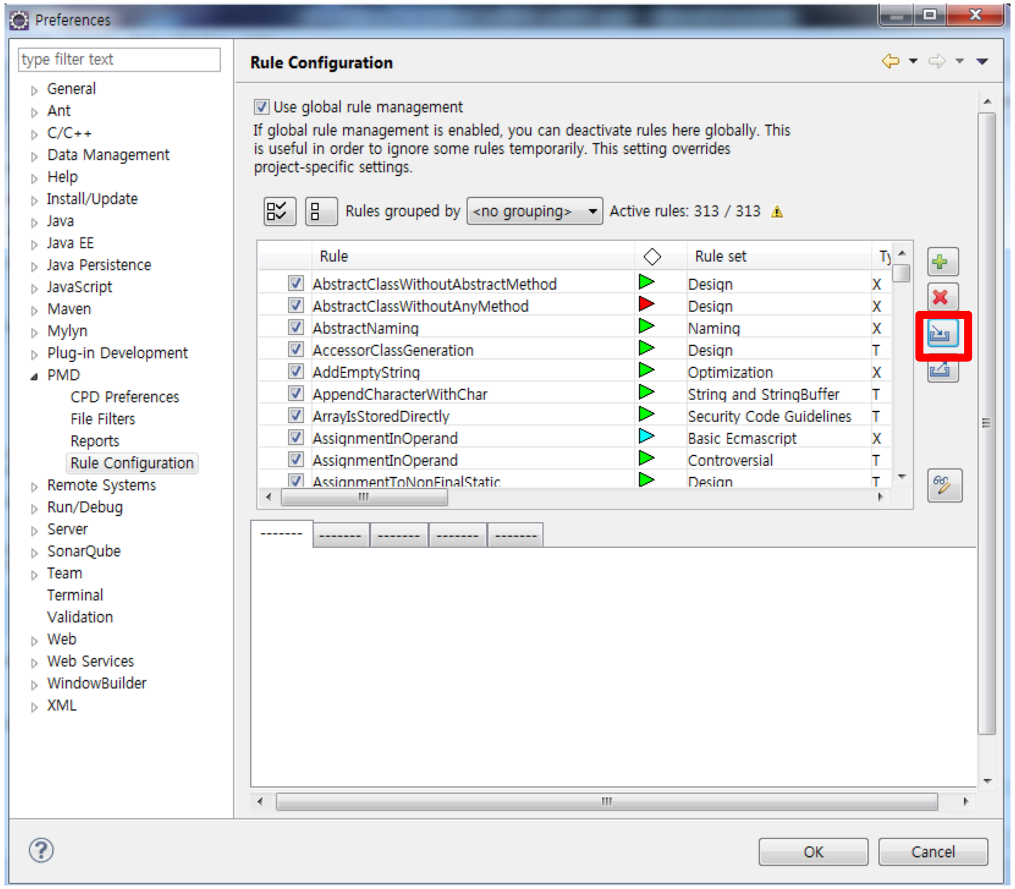
03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

About Ruleset...



Preferences

type filter text

- General
- Ant
- C/C++
- Data Management
- Help
- Install/Update
- Java
- Java EE
- Java Persistence
- JavaScript
- Maven
- Mylyn
- Plug-in Development
- PMD
  - CPD Preferences
  - File Filters
  - Reports
  - Rule Configuration
- Remote Systems
- Run/Debug
- Server
- SonarQube
- Team
- Terminal
- Validation
- Web
- Web Services
- WindowBuilder
- XML

Rule Configuration

☑ Use global rule management

If global rule management is enabled, you can deactivate rules here globally. This is useful in order to ignore some rules temporarily. This setting overrides project-specific settings.

Rules grouped by <no grouping> Active rules: 313 / 313

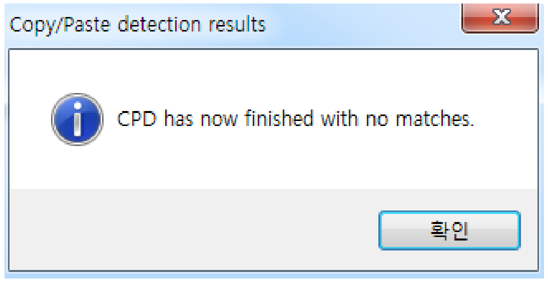
	Rule	◇	Rule set	Ty
☑	AbstractClassWithoutAbstractMethod	▶	Design	X
☑	AbstractClassWithoutAnyMethod	▶	Design	X
☑	AbstractNaming	▶	Naming	X
☑	AccessorClassGeneration	▶	Design	T
☑	AddEmptyString	▶	Optimization	X
☑	AppendCharacterWithChar	▶	String and StringBuffer	T
☑	ArrayIsStoredDirectly	▶	Security Code Guidelines	T
☑	AssignmentInOperand	▶	Basic Ecmascript	X
☑	AssignmentInOperand	▶	Controversial	T
☑	AssignmentToNonFinalStatic	▶	Design	T

OK Cancel

- [Windows]-[Preferences]-[PMD]-[Rule Configurations]-[Import Ruleset...]

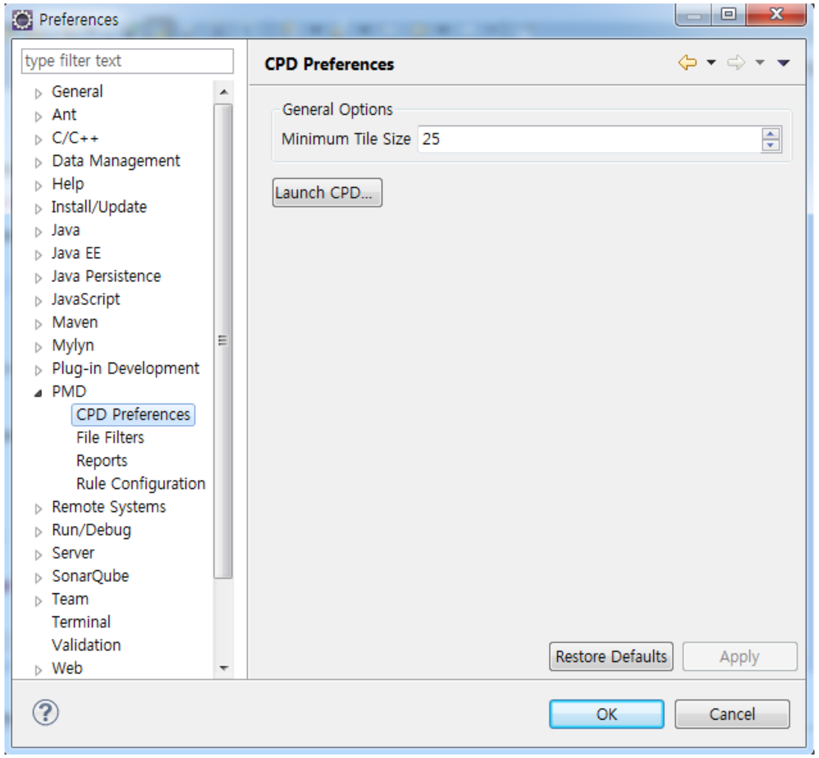
Solutions for error…

- When CPD doesn't works…



Copy/Paste detection results

CPD has now finished with no matches.



Preferences

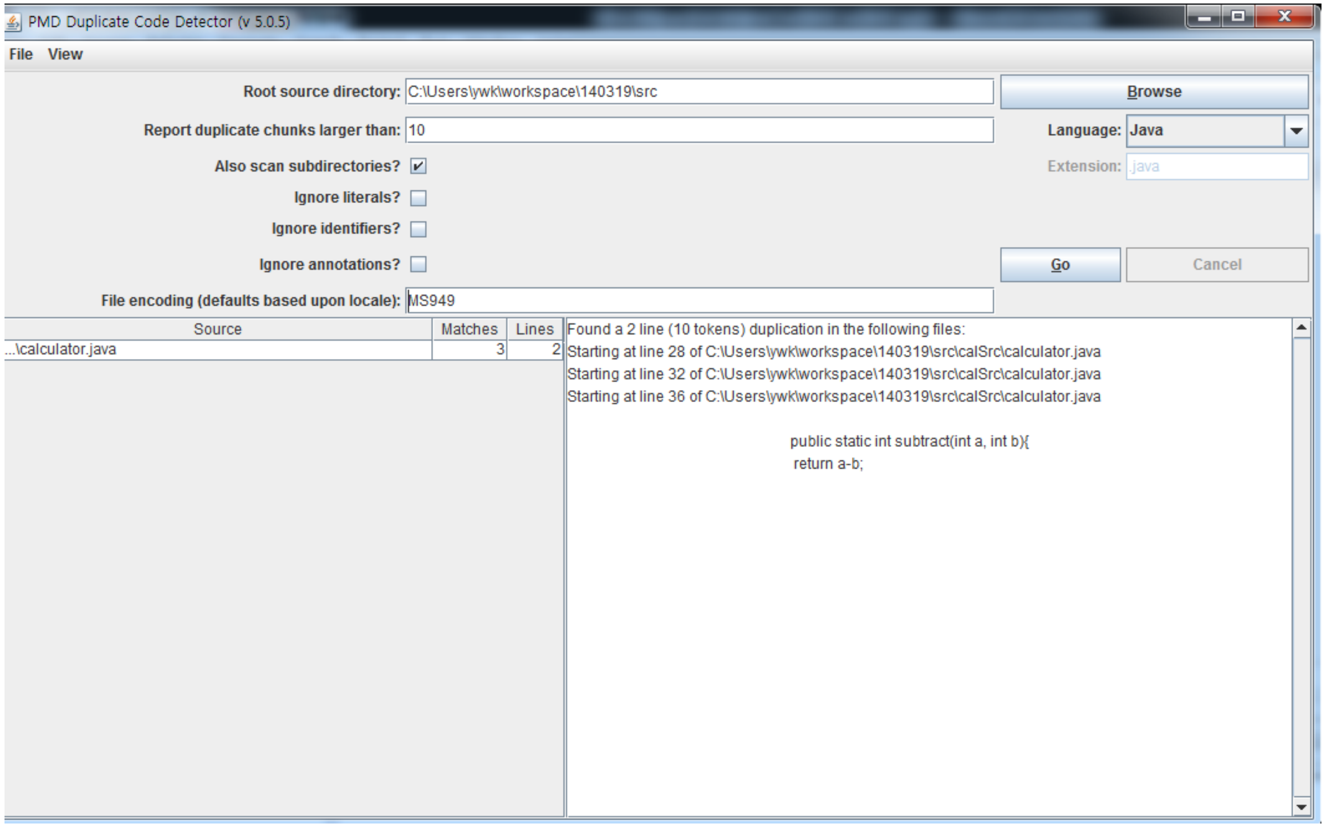
CPD Preferences

General Options

Minimum Tile Size 25

- [Windows]-[Preferences]-[PMD]-[CPD Preferences]-[Launch CPD]

Solutions for error...(cont.)



Root source directory: C:\Users\ywk\workspace\140319\src

Report duplicate chunks larger than: 10

Language: Java

Also scan subdirectories? ✓

Extension: .java

Ignore literals? ☐

Ignore identifiers? ☐

Ignore annotations? ☐

File encoding (defaults based upon locale): MS949

Source	Matches	Lines
...\calculator.java	3	2

```
Found a 2 line (10 tokens) duplication in the following files:
Starting at line 28 of C:\Users\ywk\workspace\140319\src\calSrc\calculator.java
Starting at line 32 of C:\Users\ywk\workspace\140319\src\calSrc\calculator.java
Starting at line 36 of C:\Users\ywk\workspace\140319\src\calSrc\calculator.java

                public static int subtract(int a, int b){
                 return a-b;
```

After using PMD…

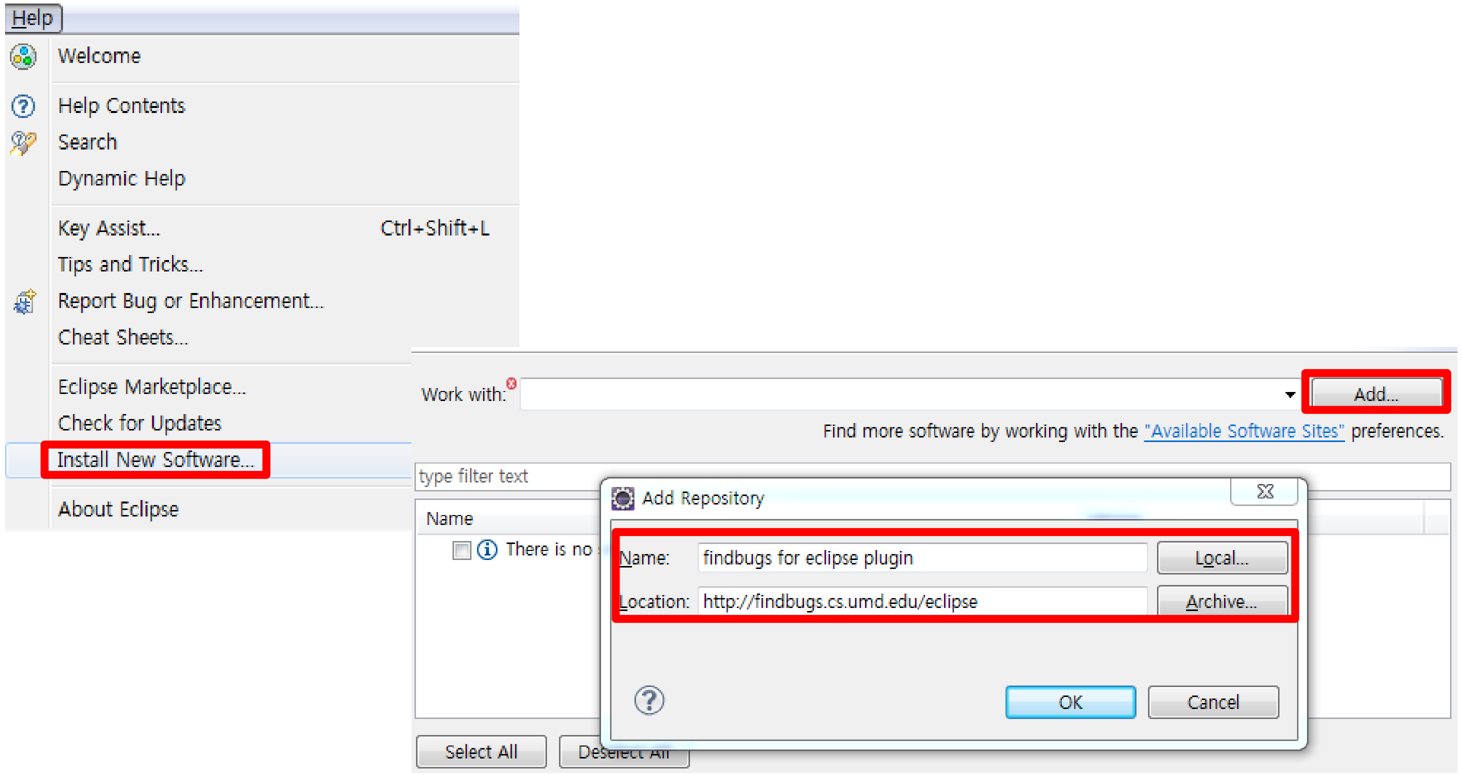
Side	points
Good	You can use PMD without Eclipse!
	There're several report formats like html,xml,test…
	Very detailed!
	You can check CPD function!
Bad	PMD doesn't support UI like findbugs.
	Tooooooo detailed!

05 FindBugs

What is Findbugs?

- FindBugs is an open source program which looks for bugs in Java code.
- It uses static analysis to identify hundreds of different potential types of errors in Java programs.
- FindBugs operates on Java bytecode,rather than source code.
- The software is distributed as a stand-alone GUI application, and also plug-ins available for Eclipse, NetBeans,IntelliJ IDEA, Gradle, Hudson and Jenkins.

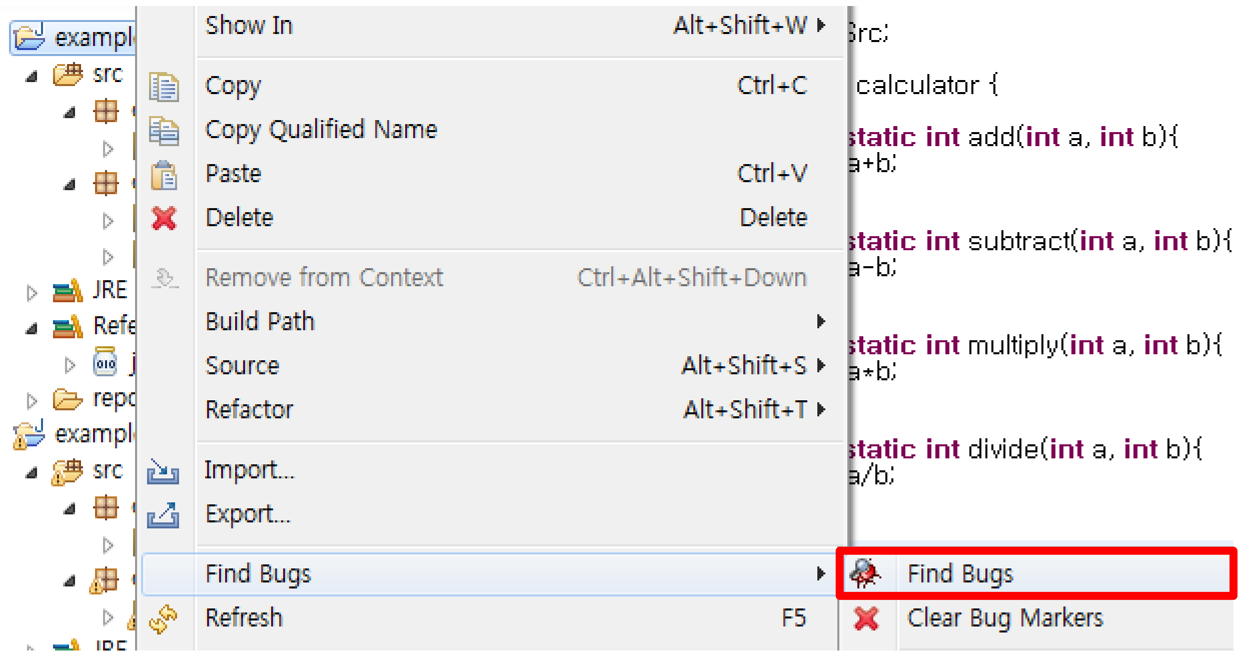
How to install Findbugs in Eclipse



Name: findbugs for eclipse plugin
Location: http://findbugs.cs.umd.edu/eclipse

- [Help] – [Install new software…]
- http://findbugs.cs.umd.edu/eclipse

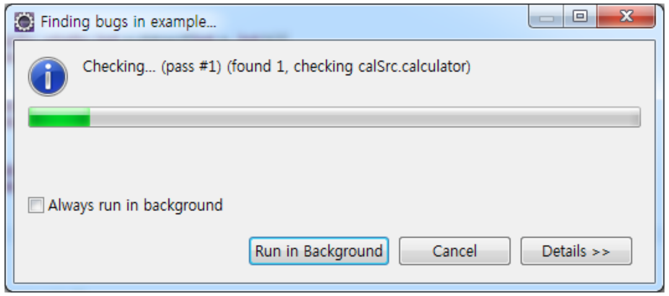
How to use Findbugs in Eclipse



```
rc;
calculator {
static int add(int a, int b){
a+b;
static int subtract(int a, int b){
a-b;
static int multiply(int a, int b){
a*b;
static int divide(int a, int b){
a/b;
```

- right click your project and [Find bugs]-[Find Bugs]

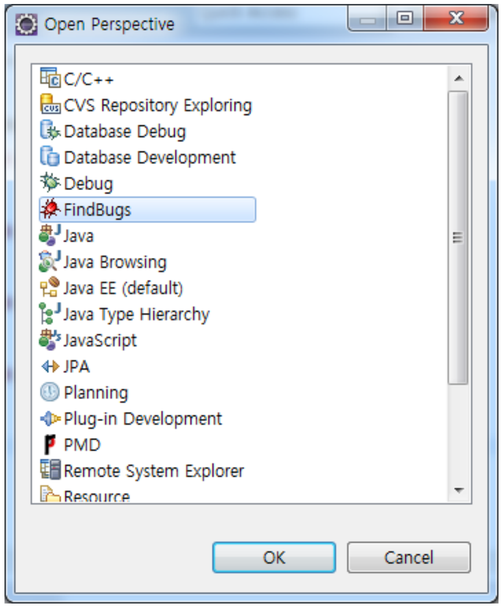
How to use Findbugs in Eclipse(cont.)



Finding bugs in example...

Checking... (pass #1) (found 1, checking calSrc.calculator)

Always run in background

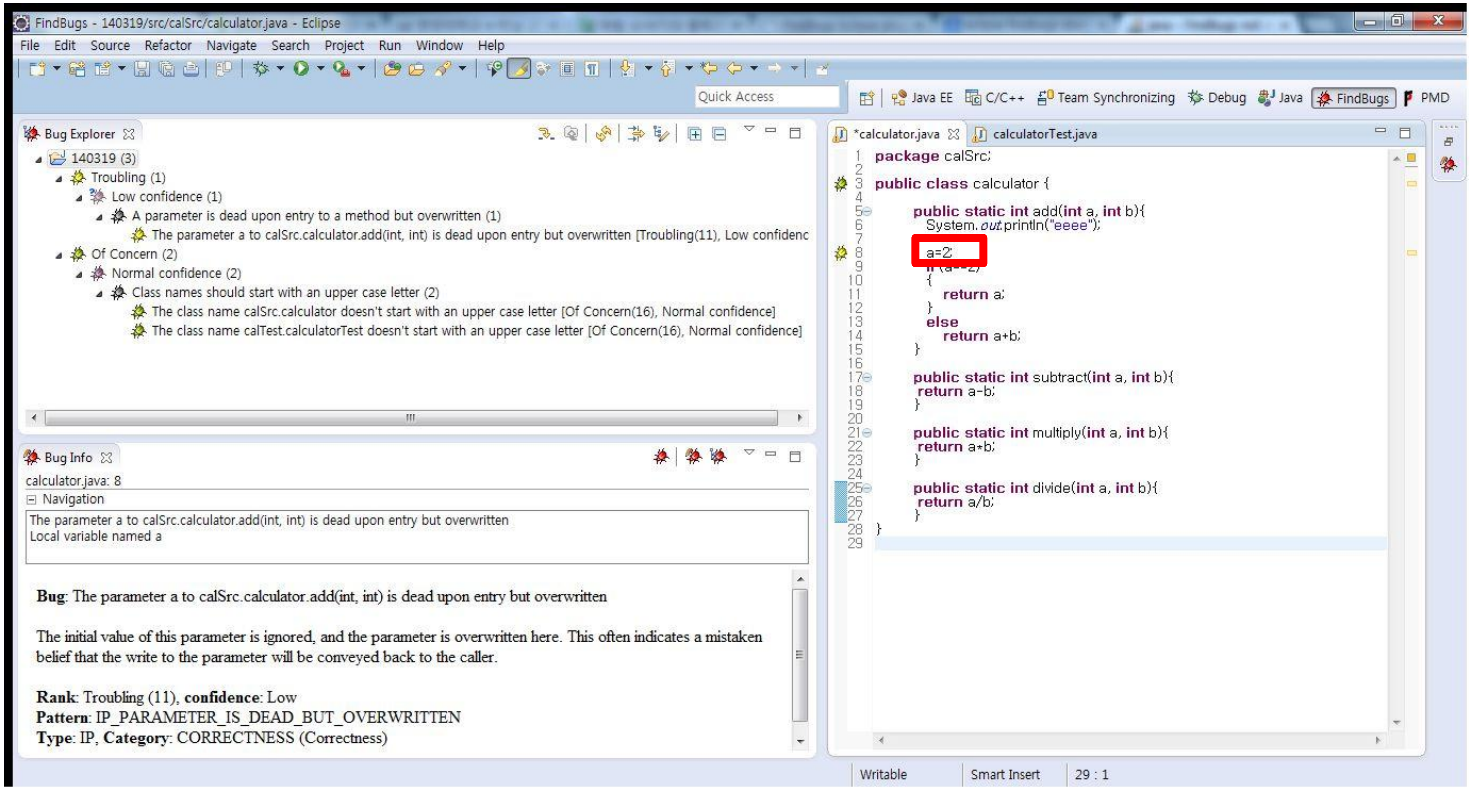


Open Perspective

C/C++
CVS Repository Exploring
Database Debug
Database Development
Debug
FindBugs
Java
Java Browsing
Java EE (default)
Java Type Hierarchy
JavaScript
JPA
Planning
Plug-in Development
PMD
Remote System Explorer
Resource

- [Window]-[open perspective]-[FindBugs]

How to use Findbugs in Eclipse(cont.)



```
140319 (3)
  Troubling (1)
    Low confidence (1)
      A parameter is dead upon entry to a method but overwritten (1)
        The parameter a to calSrc.calculator.add(int, int) is dead upon entry but overwritten [Troubling(11), Low confidenc
  Of Concern (2)
    Normal confidence (2)
      Class names should start with an upper case letter (2)
        The class name calSrc.calculator doesn't start with an upper case letter [Of Concern(16), Normal confidence]
        The class name calTest.calculatorTest doesn't start with an upper case letter [Of Concern(16), Normal confidence]
```

```java
package calSrc;

public class calculator {

        public static int add(int a, int b){
          System.out.println("eeee");

          a=2;
          if (a==2)
          {
             return a;
          }
          else
             return a+b;
        }

        public static int subtract(int a, int b){
         return a-b;
        }

        public static int multiply(int a, int b){
         return a*b;
        }

        public static int divide(int a, int b){
         return a/b;
        }
}
```

calculator.java: 8

Navigation

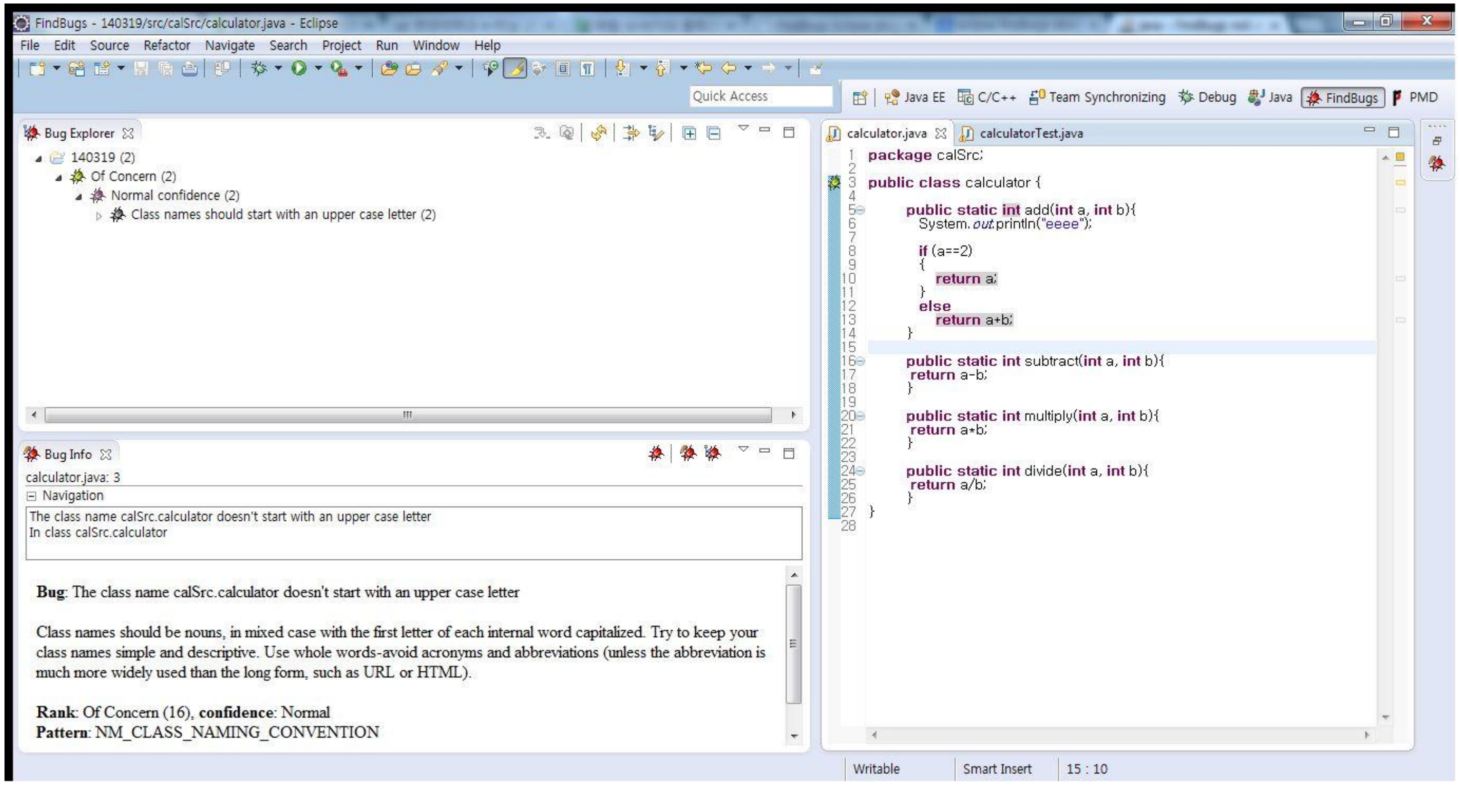
The parameter a to calSrc.calculator.add(int, int) is dead upon entry but overwritten
Local variable named a

Bug: The parameter a to calSrc.calculator.add(int, int) is dead upon entry but overwritten

The initial value of this parameter is ignored, and the parameter is overwritten here. This often indicates a mistaken belief that the write to the parameter will be conveyed back to the caller.

Rank: Troubling (11), **confidence**: Low
Pattern: IP_PARAMETER_IS_DEAD_BUT_OVERWRITTEN
Type: IP, **Category**: CORRECTNESS (Correctness)

How to use Findbugs in Eclipse(cont.)



The class name calSrc.calculator doesn't start with an upper case letter
In class calSrc.calculator

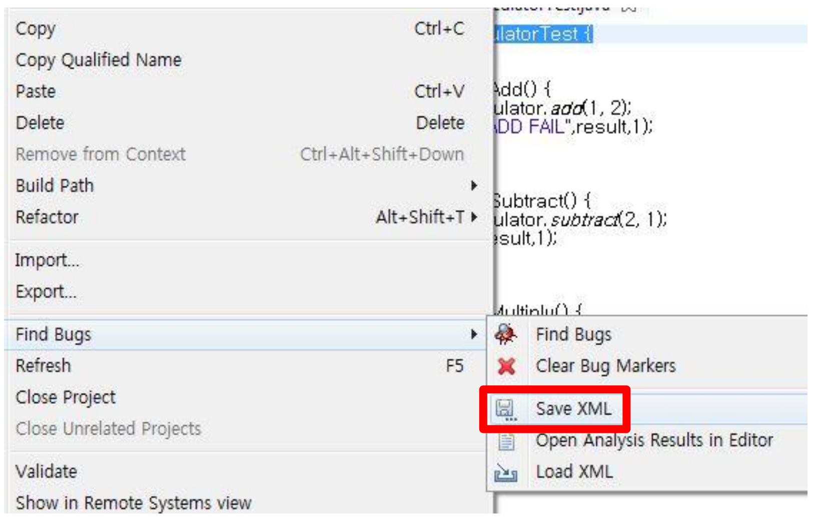
Bug: The class name calSrc.calculator doesn't start with an upper case letter

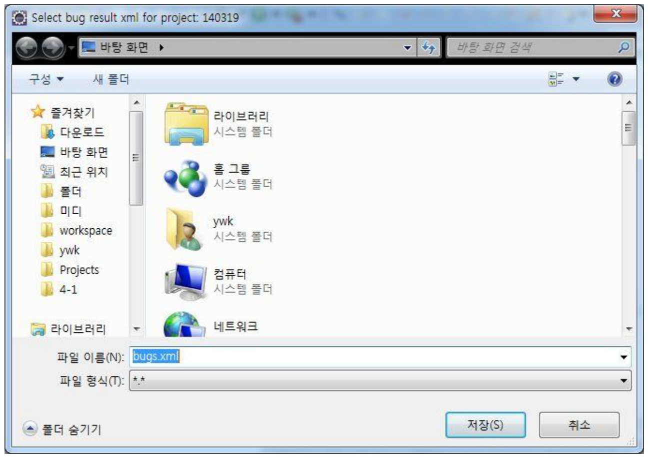
Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).

Rank: Of Concern (16), **confidence**: Normal
Pattern: NM_CLASS_NAMING_CONVENTION

```
package calSrc;

public class calculator {

    public static int add(int a, int b){
      System.out.println("eeee");

      if (a==2)
      {
         return a;
      }
      else
         return a+b;
    }

    public static int subtract(int a, int b){
     return a-b;
    }

    public static int multiply(int a, int b){
     return a*b;
    }

    public static int divide(int a, int b){
     return a/b;
    }
}
```

How to make bug report





- right click to project-[Find Bugs]-[Save XML]

01 JUnit

02 Eclipse

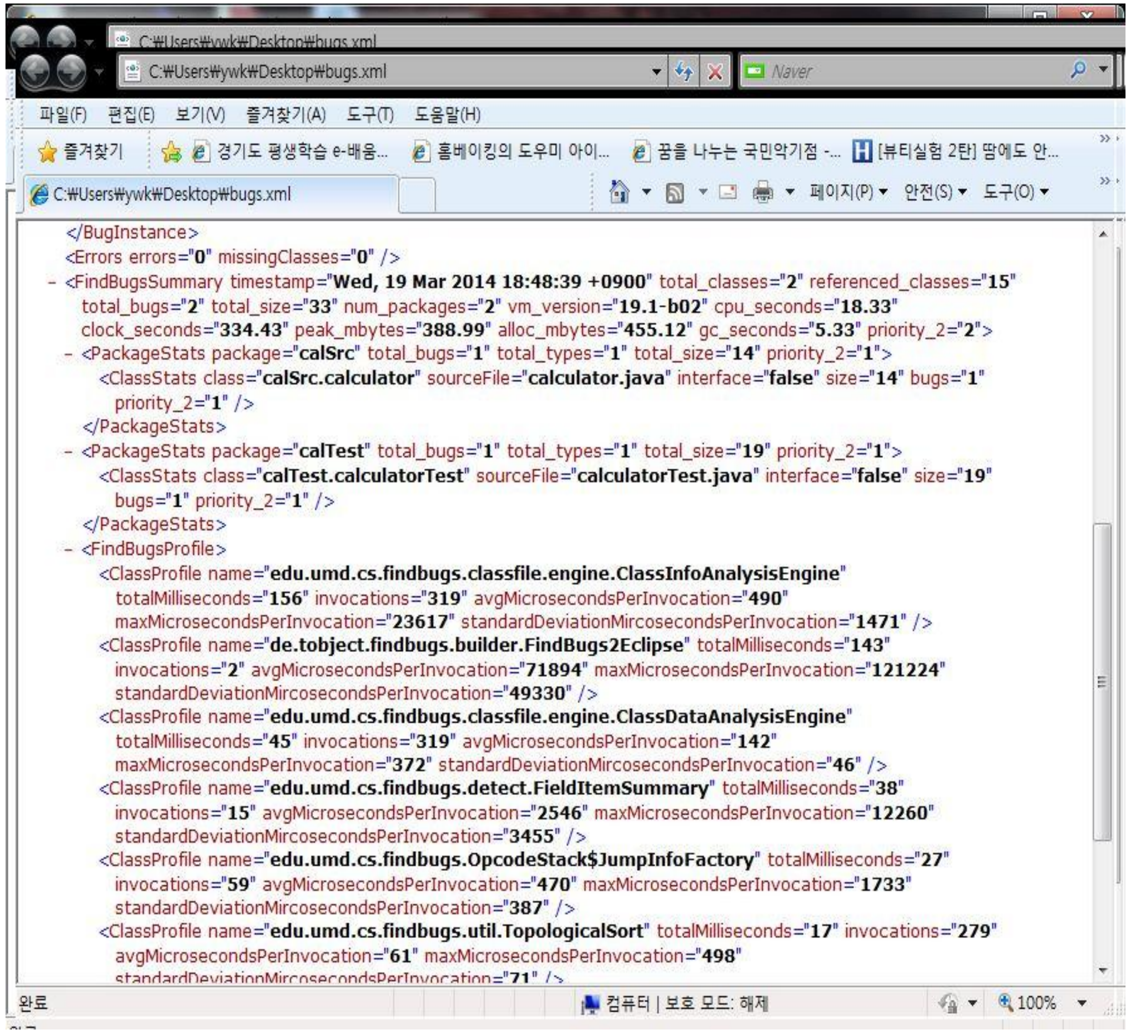
03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

How to make bug report(cont.)

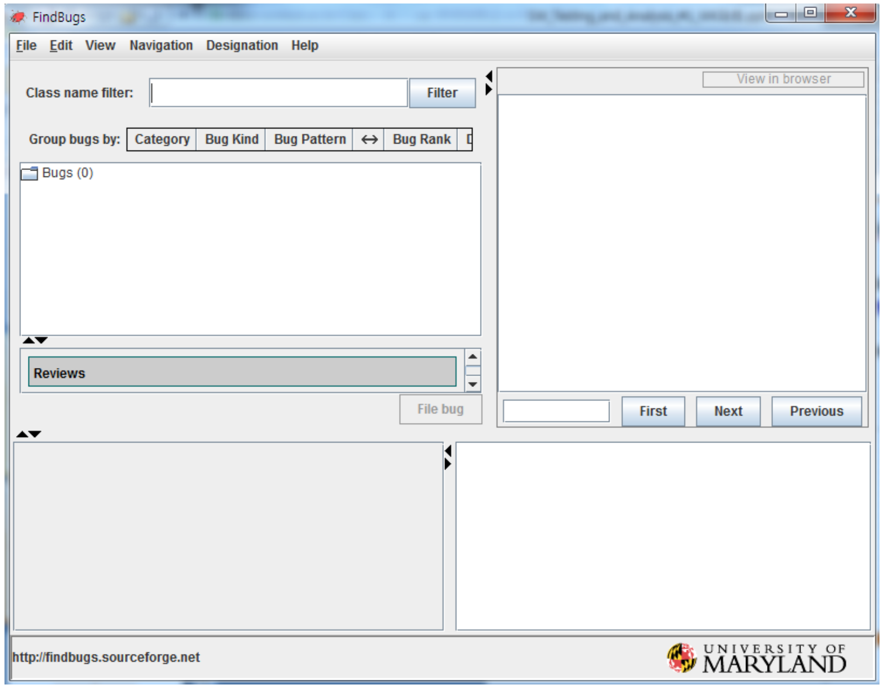

C:₩Users₩ywk₩Desktop₩bugs.xml
Naver
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
경기도 평생학습 e-배움...
홈베이킹의 도우미 아이...
꿈을 나누는 국민악기점 -...
[뷰티실험 2탄] 땀에도 안...
C:₩Users₩ywk₩Desktop₩bugs.xml
페이지(P) 안전(S) 도구(O)
완료
컴퓨터 | 보호 모드: 해제
100%

```
  </BugInstance>
  <Errors errors="0" missingClasses="0" />
- <FindBugsSummary timestamp="Wed, 19 Mar 2014 18:48:39 +0900" total_classes="2" referenced_classes="15"
    total_bugs="2" total_size="33" num_packages="2" vm_version="19.1-b02" cpu_seconds="18.33"
    clock_seconds="334.43" peak_mbytes="388.99" alloc_mbytes="455.12" gc_seconds="5.33" priority_2="2">
  - <PackageStats package="calSrc" total_bugs="1" total_types="1" total_size="14" priority_2="1">
      <ClassStats class="calSrc.calculator" sourceFile="calculator.java" interface="false" size="14" bugs="1"
        priority_2="1" />
    </PackageStats>
  - <PackageStats package="calTest" total_bugs="1" total_types="1" total_size="19" priority_2="1">
      <ClassStats class="calTest.calculatorTest" sourceFile="calculatorTest.java" interface="false" size="19"
        bugs="1" priority_2="1" />
    </PackageStats>
  - <FindBugsProfile>
      <ClassProfile name="edu.umd.cs.findbugs.classfile.engine.ClassInfoAnalysisEngine"
        totalMilliseconds="156" invocations="319" avgMicrosecondsPerInvocation="490"
        maxMicrosecondsPerInvocation="23617" standardDeviationMircosecondsPerInvocation="1471" />
      <ClassProfile name="de.tobject.findbugs.builder.FindBugs2Eclipse" totalMilliseconds="143"
        invocations="2" avgMicrosecondsPerInvocation="71894" maxMicrosecondsPerInvocation="121224"
        standardDeviationMircosecondsPerInvocation="49330" />
      <ClassProfile name="edu.umd.cs.findbugs.classfile.engine.ClassDataAnalysisEngine"
        totalMilliseconds="45" invocations="319" avgMicrosecondsPerInvocation="142"
        maxMicrosecondsPerInvocation="372" standardDeviationMircosecondsPerInvocation="46" />
      <ClassProfile name="edu.umd.cs.findbugs.detect.FieldItemSummary" totalMilliseconds="38"
        invocations="15" avgMicrosecondsPerInvocation="2546" maxMicrosecondsPerInvocation="12260"
        standardDeviationMircosecondsPerInvocation="3455" />
      <ClassProfile name="edu.umd.cs.findbugs.OpcodeStack$JumpInfoFactory" totalMilliseconds="27"
        invocations="59" avgMicrosecondsPerInvocation="470" maxMicrosecondsPerInvocation="1733"
        standardDeviationMircosecondsPerInvocation="387" />
      <ClassProfile name="edu.umd.cs.findbugs.util.TopologicalSort" totalMilliseconds="17" invocations="279"
        avgMicrosecondsPerInvocation="61" maxMicrosecondsPerInvocation="498"
        standardDeviationMircosecondsPerInvocation="71" />
```

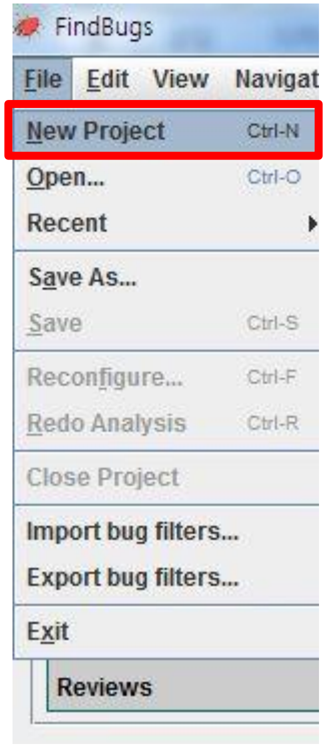
How to use FindBugs UI

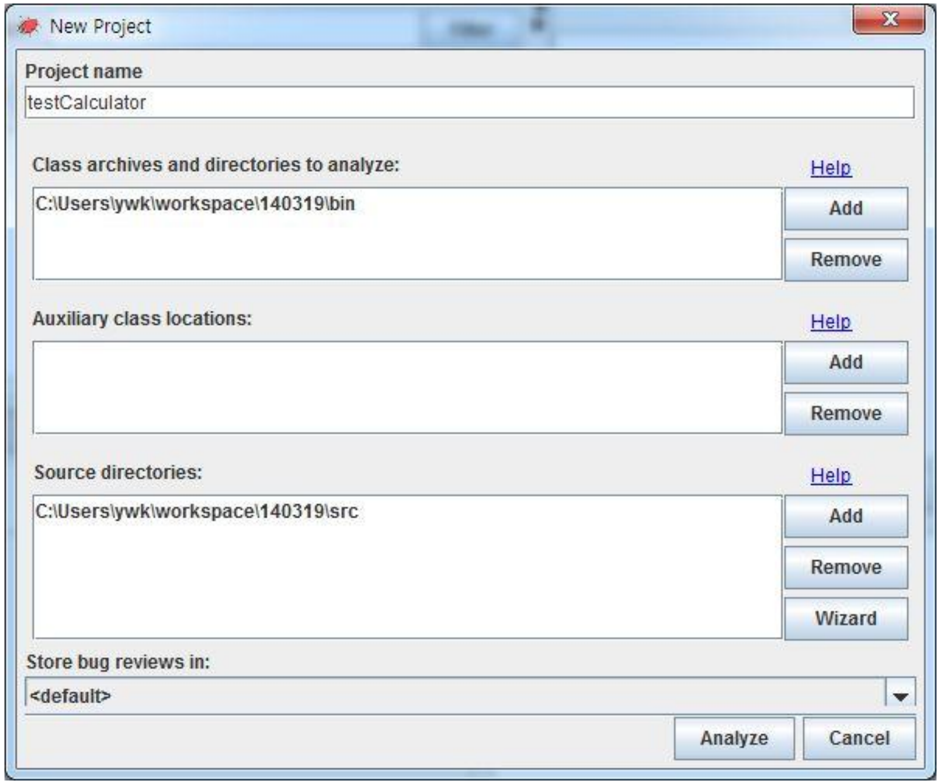
- download findbugs-version.zip
http://findbugs.sourceforge.net/downloads.html

- unzip and execute bin\findbugs.bat


FindBugs
File Edit View Navigation Designation Help
Class name filter:
Filter
View in browser
Group bugs by: Category Bug Kind Bug Pattern Bug Rank
Bugs (0)
Reviews
File bug
First Next Previous
http://findbugs.sourceforge.net
UNIVERSITY OF MARYLAND

How to use FindBugs UI(cont.)




New Project

Project name: testCalculator

Class archives and directories to analyze: C:\Users\ywk\workspace\140319\bin

Auxiliary class locations:

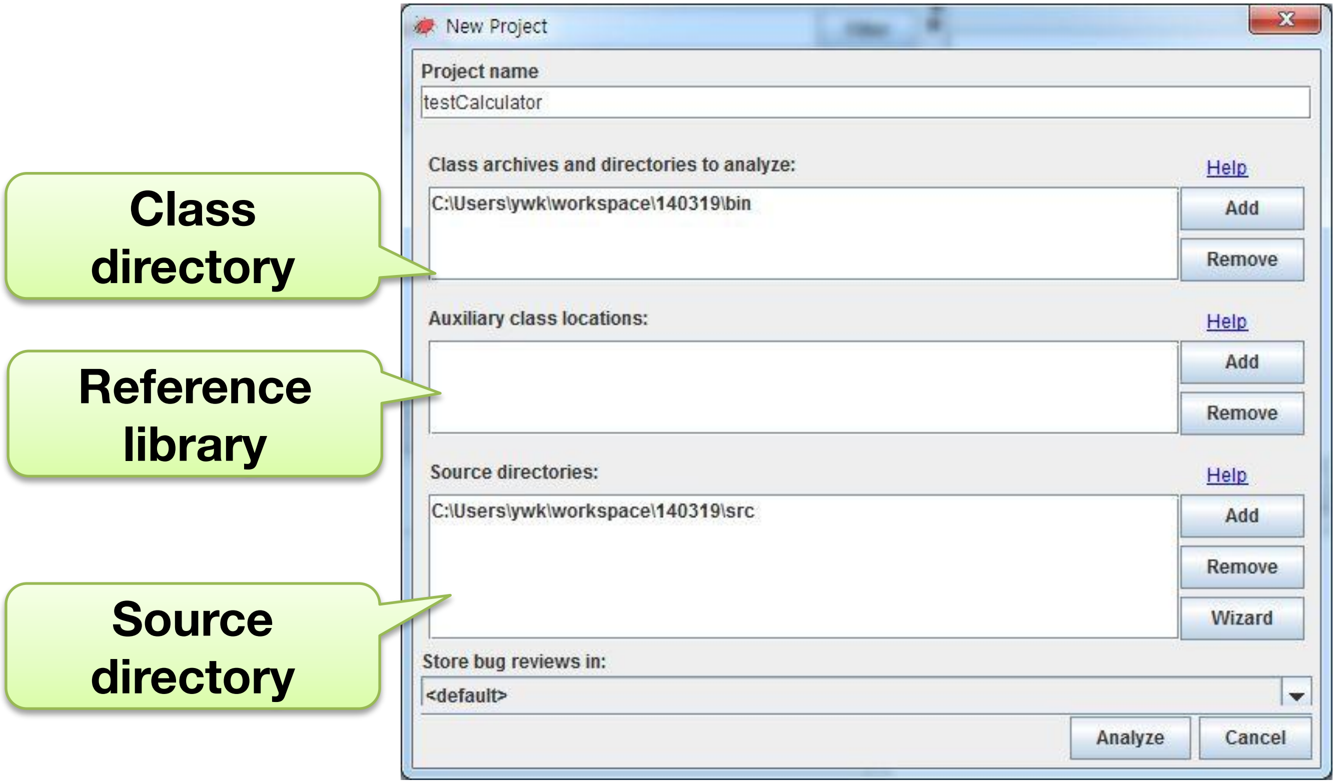
Source directories: C:\Users\ywk\workspace\140319\src

Store bug reviews in: <default>

- [File]-[New Project]
- Fill the blanks

TM

How to use FindBugs UI(cont.)



Project name: testCalculator

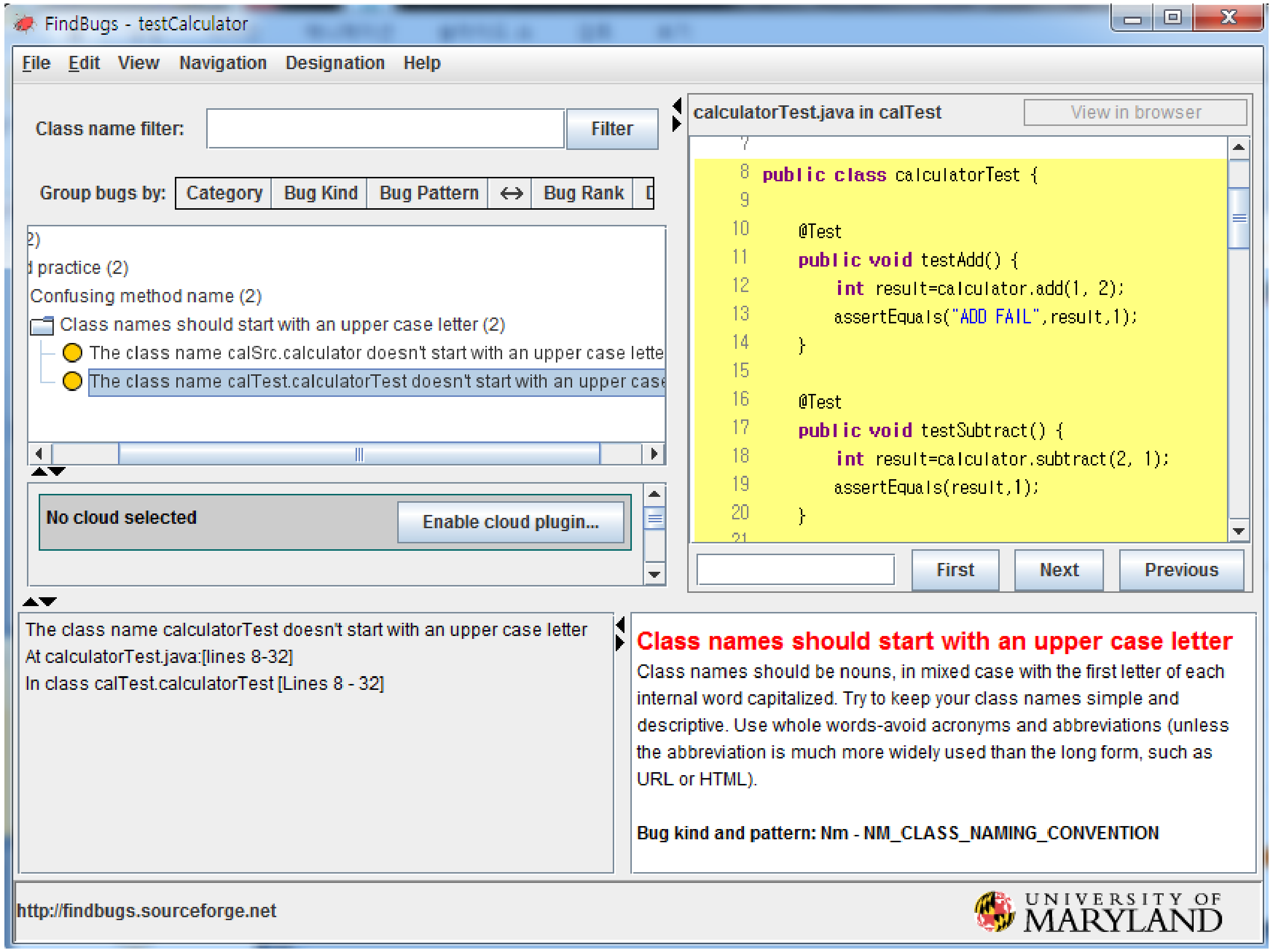
Class archives and directories to analyze: C:\Users\ywk\workspace\140319\bin

Auxiliary class locations:

Source directories: C:\Users\ywk\workspace\140319\src

Store bug reviews in: <default>

How to use FindBugs UI(cont.)


FindBugs - testCalculator
File Edit View Navigation Designation Help
Class name filter:
Filter
Group bugs by: Category Bug Kind Bug Pattern ↔ Bug Rank
calculatorTest.java in calTest
View in browser
First Next Previous
No cloud selected
Enable cloud plugin...
http://findbugs.sourceforge.net
UNIVERSITY OF MARYLAND

2)
d practice (2)
Confusing method name (2)
Class names should start with an upper case letter (2)
The class name calSrc.calculator doesn't start with an upper case lette
The class name calTest.calculatorTest doesn't start with an upper case

```
public class calculatorTest {

    @Test
    public void testAdd() {
        int result=calculator.add(1, 2);
        assertEquals("ADD FAIL",result,1);
    }

    @Test
    public void testSubtract() {
        int result=calculator.subtract(2, 1);
        assertEquals(result,1);
    }
```

The class name calculatorTest doesn't start with an upper case letter
At calculatorTest.java:[lines 8-32]
In class calTest.calculatorTest [Lines 8 - 32]

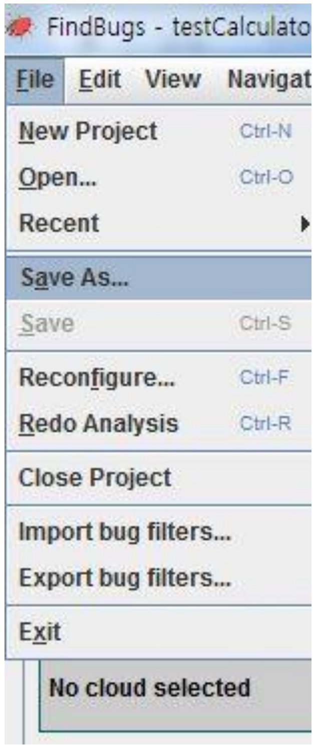
Class names should start with an upper case letter

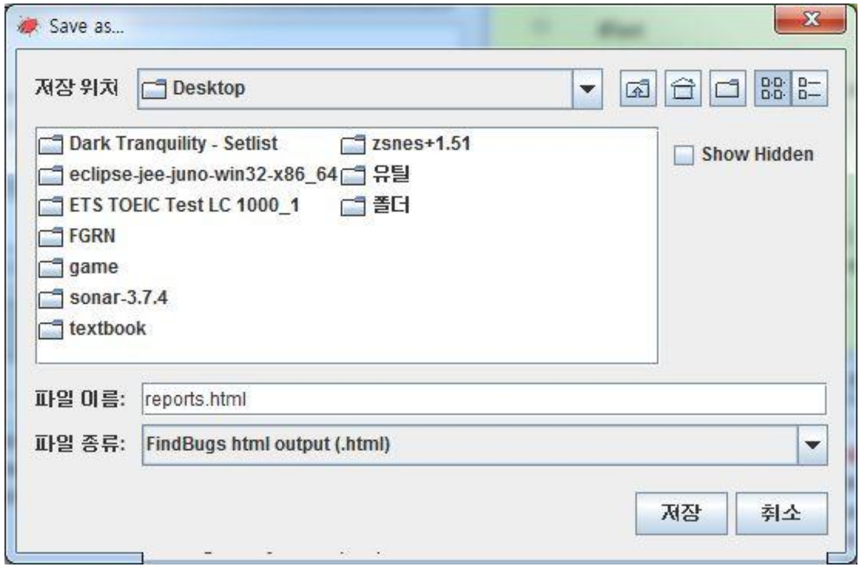
Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).

Bug kind and pattern: Nm - NM_CLASS_NAMING_CONVENTION

TM

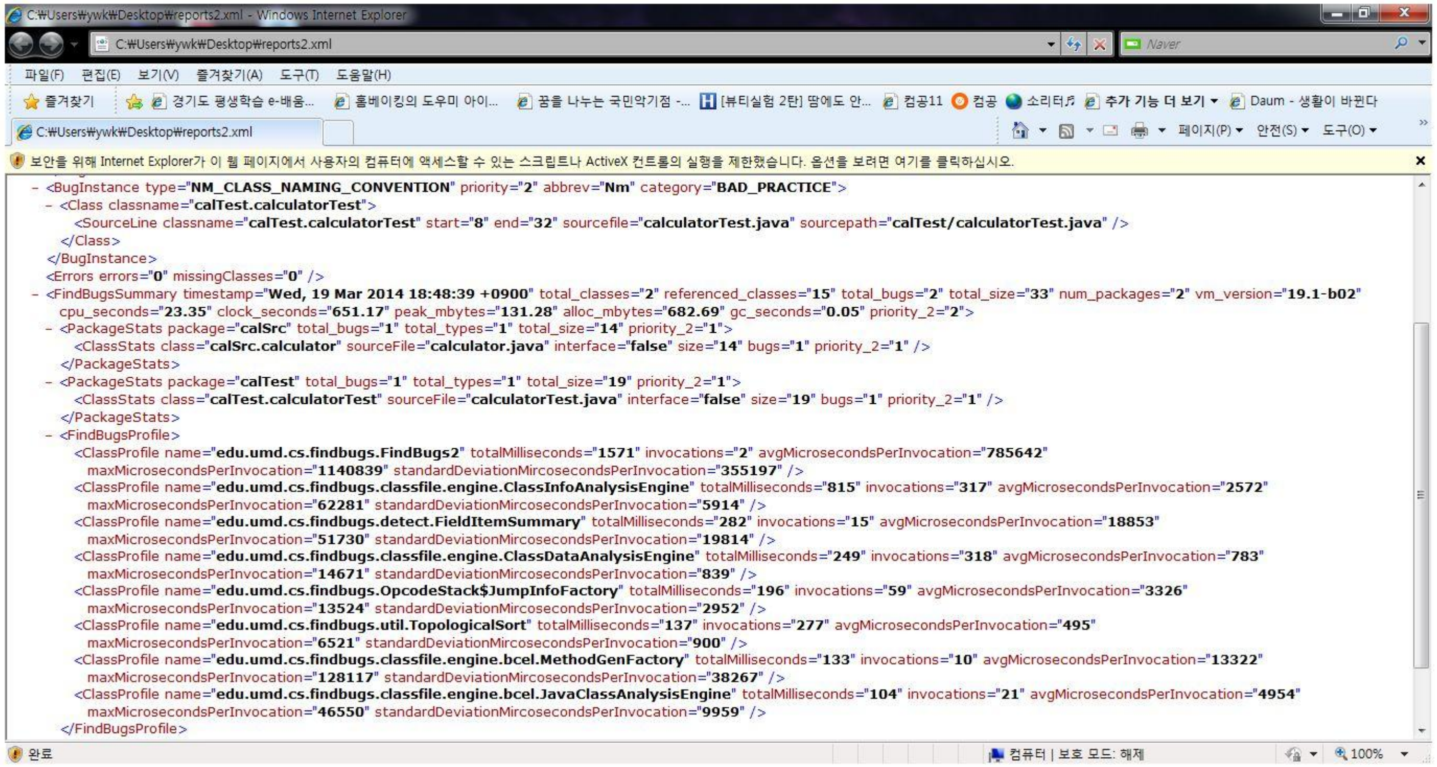
How to make bug reports





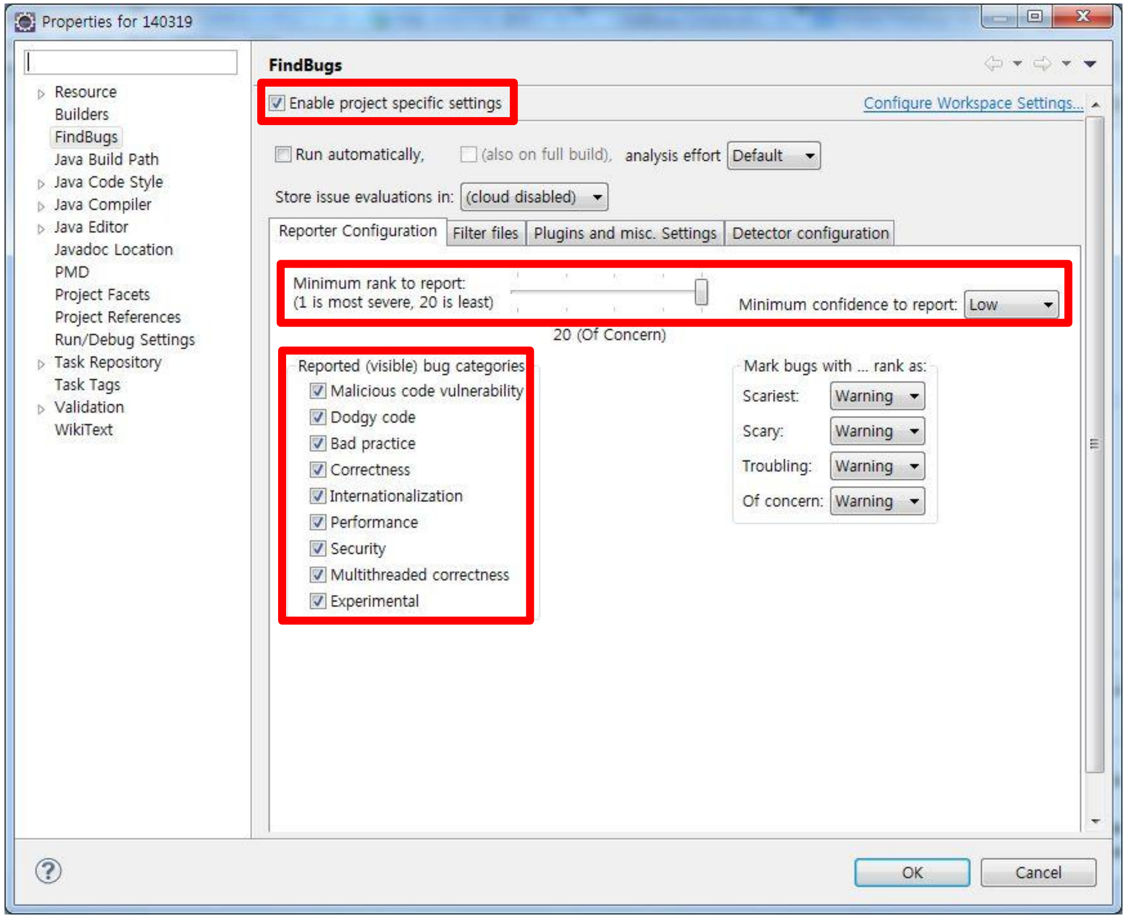
- [File]-[Save As…]

How to make bug reports(cont.)


C:₩Users₩ywk₩Desktop₩reports2.xml - Windows Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
페이지(P) 안전(S) 도구(O)
보안을 위해 Internet Explorer가 이 웹 페이지에서 사용자의 컴퓨터에 액세스할 수 있는 스크립트나 ActiveX 컨트롤의 실행을 제한했습니다. 옵션을 보려면 여기를 클릭하십시오.
완료
컴퓨터 | 보호 모드: 해제
100%

```
- <BugInstance type="NM_CLASS_NAMING_CONVENTION" priority="2" abbrev="Nm" category="BAD_PRACTICE">
  - <Class classname="calTest.calculatorTest">
      <SourceLine classname="calTest.calculatorTest" start="8" end="32" sourcefile="calculatorTest.java" sourcepath="calTest/calculatorTest.java" />
    </Class>
  </BugInstance>
  <Errors errors="0" missingClasses="0" />
- <FindBugsSummary timestamp="Wed, 19 Mar 2014 18:48:39 +0900" total_classes="2" referenced_classes="15" total_bugs="2" total_size="33" num_packages="2" vm_version="19.1-b02"
    cpu_seconds="23.35" clock_seconds="651.17" peak_mbytes="131.28" alloc_mbytes="682.69" gc_seconds="0.05" priority_2="2">
  - <PackageStats package="calSrc" total_bugs="1" total_types="1" total_size="14" priority_2="1">
      <ClassStats class="calSrc.calculator" sourceFile="calculator.java" interface="false" size="14" bugs="1" priority_2="1" />
    </PackageStats>
  - <PackageStats package="calTest" total_bugs="1" total_types="1" total_size="19" priority_2="1">
      <ClassStats class="calTest.calculatorTest" sourceFile="calculatorTest.java" interface="false" size="19" bugs="1" priority_2="1" />
    </PackageStats>
  - <FindBugsProfile>
      <ClassProfile name="edu.umd.cs.findbugs.FindBugs2" totalMilliseconds="1571" invocations="2" avgMicrosecondsPerInvocation="785642"
        maxMicrosecondsPerInvocation="1140839" standardDeviationMircosecondsPerInvocation="355197" />
      <ClassProfile name="edu.umd.cs.findbugs.classfile.engine.ClassInfoAnalysisEngine" totalMilliseconds="815" invocations="317" avgMicrosecondsPerInvocation="2572"
        maxMicrosecondsPerInvocation="62281" standardDeviationMircosecondsPerInvocation="5914" />
      <ClassProfile name="edu.umd.cs.findbugs.detect.FieldItemSummary" totalMilliseconds="282" invocations="15" avgMicrosecondsPerInvocation="18853"
        maxMicrosecondsPerInvocation="51730" standardDeviationMircosecondsPerInvocation="19814" />
      <ClassProfile name="edu.umd.cs.findbugs.classfile.engine.ClassDataAnalysisEngine" totalMilliseconds="249" invocations="318" avgMicrosecondsPerInvocation="783"
        maxMicrosecondsPerInvocation="14671" standardDeviationMircosecondsPerInvocation="839" />
      <ClassProfile name="edu.umd.cs.findbugs.OpcodeStack$JumpInfoFactory" totalMilliseconds="196" invocations="59" avgMicrosecondsPerInvocation="3326"
        maxMicrosecondsPerInvocation="13524" standardDeviationMircosecondsPerInvocation="2952" />
      <ClassProfile name="edu.umd.cs.findbugs.util.TopologicalSort" totalMilliseconds="137" invocations="277" avgMicrosecondsPerInvocation="495"
        maxMicrosecondsPerInvocation="6521" standardDeviationMircosecondsPerInvocation="900" />
      <ClassProfile name="edu.umd.cs.findbugs.classfile.engine.bcel.MethodGenFactory" totalMilliseconds="133" invocations="10" avgMicrosecondsPerInvocation="13322"
        maxMicrosecondsPerInvocation="128117" standardDeviationMircosecondsPerInvocation="38267" />
      <ClassProfile name="edu.umd.cs.findbugs.classfile.engine.bcel.JavaClassAnalysisEngine" totalMilliseconds="104" invocations="21" avgMicrosecondsPerInvocation="4954"
        maxMicrosecondsPerInvocation="46550" standardDeviationMircosecondsPerInvocation="9959" />
    </FindBugsProfile>
```

Solutions for errors


Properties for 140319

FindBugs

- [✓] Enable project specific settings
- Configure Workspace Settings...
- [ ] Run automatically, [ ] (also on full build), analysis effort: Default
- Store issue evaluations in: (cloud disabled)

Reporter Configuration | Filter files | Plugins and misc. Settings | Detector configuration

Minimum rank to report: (1 is most severe, 20 is least) 20 (Of Concern)
Minimum confidence to report: Low

Reported (visible) bug categories
- [✓] Malicious code vulnerability
- [✓] Dodgy code
- [✓] Bad practice
- [✓] Correctness
- [✓] Internationalization
- [✓] Performance
- [✓] Security
- [✓] Multithreaded correctness
- [✓] Experimental

Mark bugs with ... rank as:

Scariest:	Warning
Scary:	Warning
Troubling:	Warning
Of concern:	Warning

- If findbugs doesn't work and show any reports… check your project [properties]-[FindBugs]

01 JUnit

02 Eclipse

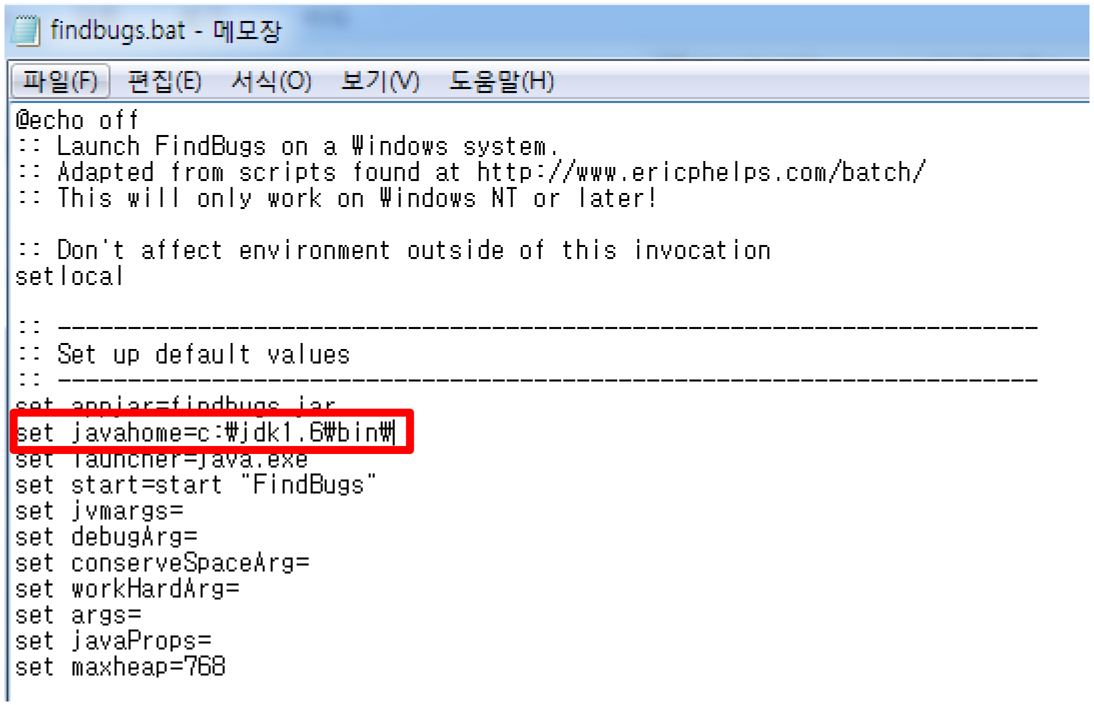
03 CheckStyle

04 PMD

05 FindBugs

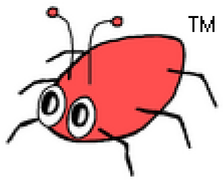
06 SONAR

Solutions for errors(cont.)

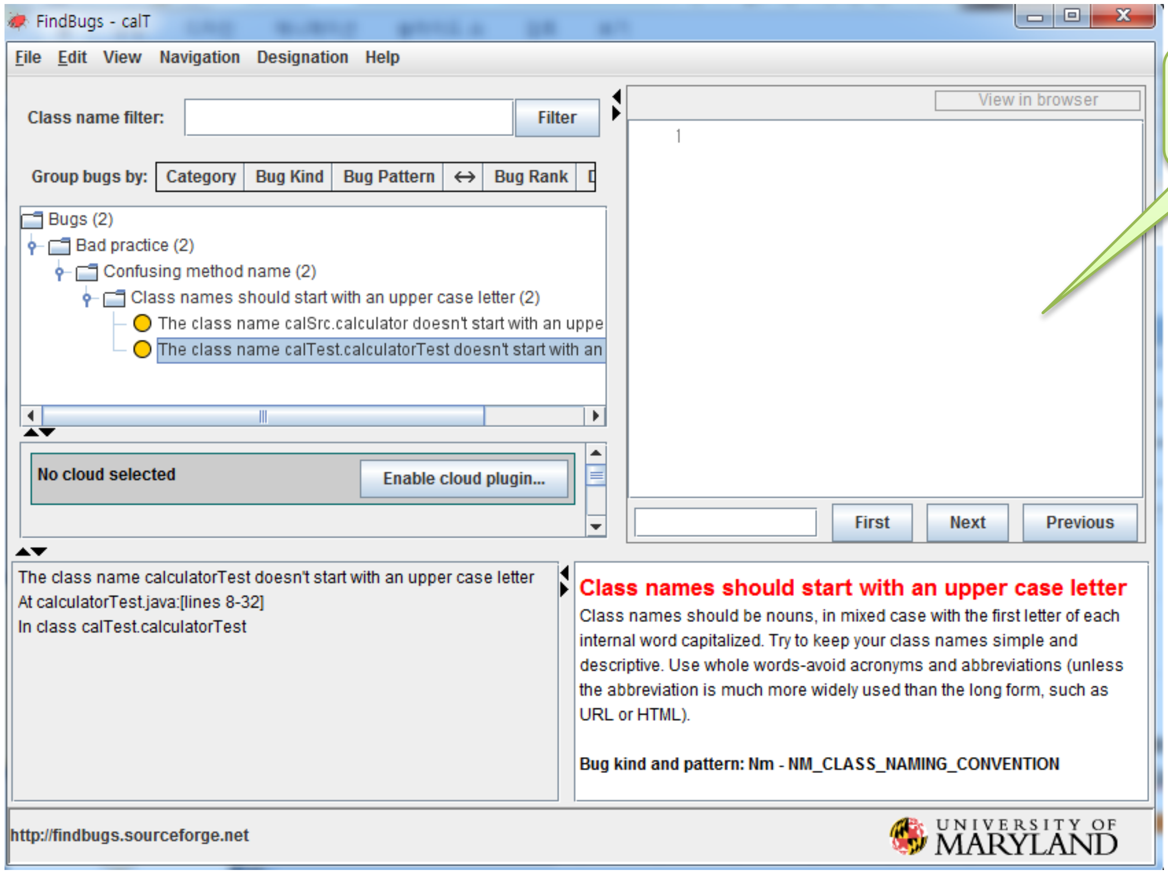


```
@echo off
:: Launch FindBugs on a Windows system.
:: Adapted from scripts found at http://www.ericphelps.com/batch/
:: This will only work on Windows NT or later!

:: Don't affect environment outside of this invocation
setlocal

:: ----------------------------------------------------------------------
:: Set up default values
:: ----------------------------------------------------------------------
set appjar=findbugs.jar
set javahome=c:₩jdk1.6₩bin₩
set launcher=java.exe
set start=start "FindBugs"
set jvmargs=
set debugArg=
set conserveSpaceArg=
set workHardArg=
set args=
set javaProps=
set maxheap=768
```

- **If findbugs batch file disappears…**
open the batch file and edit set javahome=JDK bin dir.



Solutions for errors(cont.)



```
Bugs (2)
  Bad practice (2)
    Confusing method name (2)
      Class names should start with an upper case letter (2)
        The class name calSrc.calculator doesn't start with an uppe
        The class name calTest.calculatorTest doesn't start with an
```

The class name calculatorTest doesn't start with an upper case letter
At calculatorTest.java:[lines 8-32]
In class calTest.calculatorTest

Class names should start with an upper case letter
Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).

Bug kind and pattern: Nm - NM_CLASS_NAMING_CONVENTION

?

- If source code doesn't appear in FindBugs UI…
See 55 page and redefine your project directory.

01 JUnit

02 Eclipse

03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

After using FindBugs...

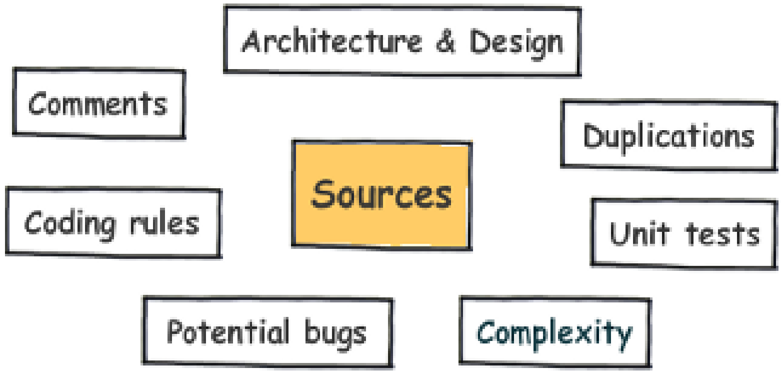
Side	points
Good	Simple! but Detailed explaination!
	Not too heavy!
	Well designed UI without Eclipse!
	Can analyze jar file!
	Can analyze without source if you only have class file!
Bad	Findbugs takes longer time than PMD...
	Reports only XML format...

06 Sonar

sonarqube

What is Sonar?

- SonarQube's numerous dashboards offer quick insight.
- SonarQube is an open platform to manage code quality. As such, it covers the 7 axes of code quality:

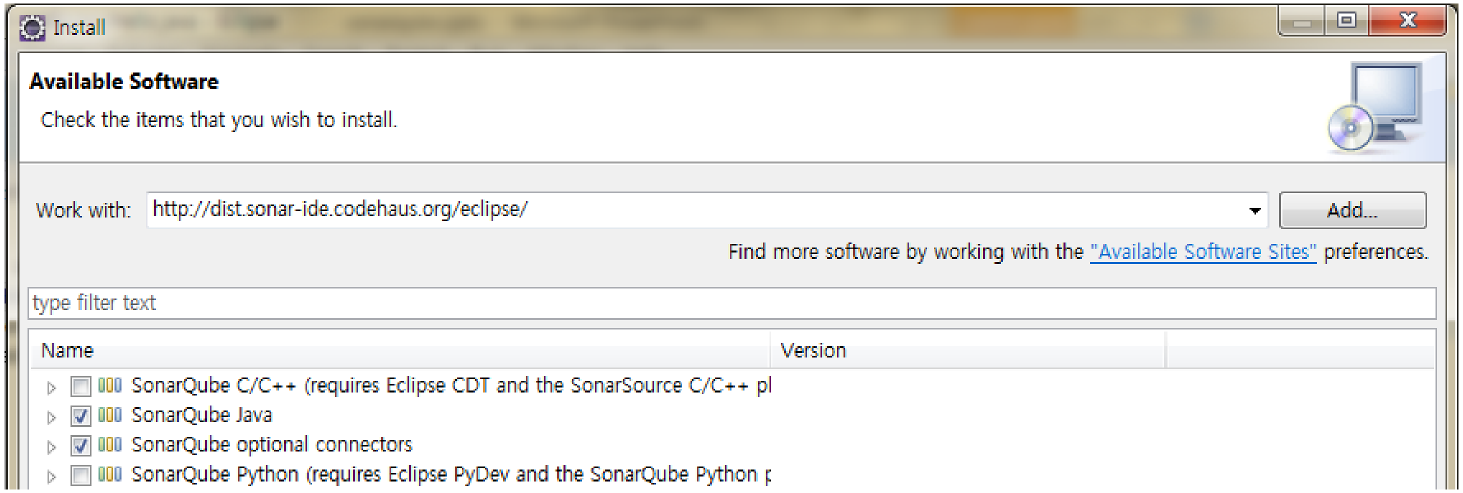


- SonarQube's numerous dashboards offer quick insight.

- Several methods are available to replay the past, showing how your metrics evolved.

sonarqube

How to install Sonar?



Available Software

Check the items that you wish to install.

Work with: http://dist.sonar-ide.codehaus.org/eclipse/ Add...

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ SonarQube C/C++ (requires Eclipse CDT and the SonarSource C/C++ pl	
☑ SonarQube Java	
☑ SonarQube optional connectors	
☐ SonarQube Python (requires Eclipse PyDev and the SonarQube Python p	

- Install Sonar plugin in Eclipse
  [Help]-[Install new software]
  → http://dist.sonar-ide.codehaus.org/eclipse/

How to install Sonar?

- Download SonarQube

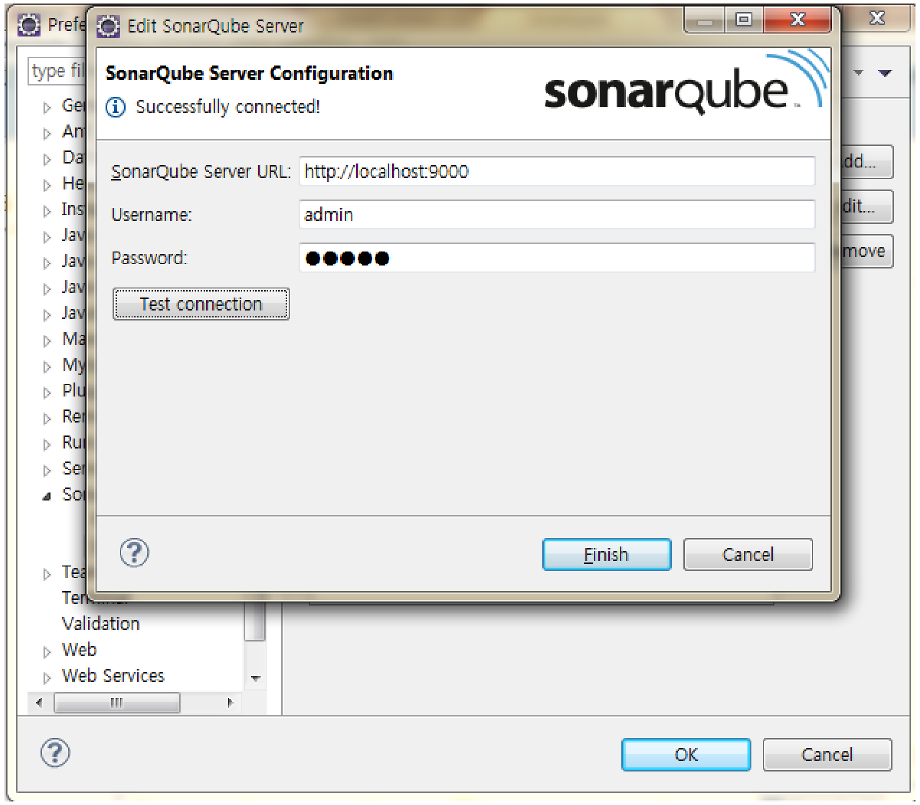
  http://www.sonarqube.org/ → Download → latest version(3.7.4) →unzip the file

- Download SonarQube Runner

  http://www.sonarqube.org/ → Download → latest version(2.3) → unzip the file

How to install Sonar?

- 01 JUnit
- 02 Eclipse
- 03 CheckStyle
- 04 PMD
- 05 FindBugs
- **06 SONAR**



Edit SonarQube Server

SonarQube Server Configuration

Successfully connected!

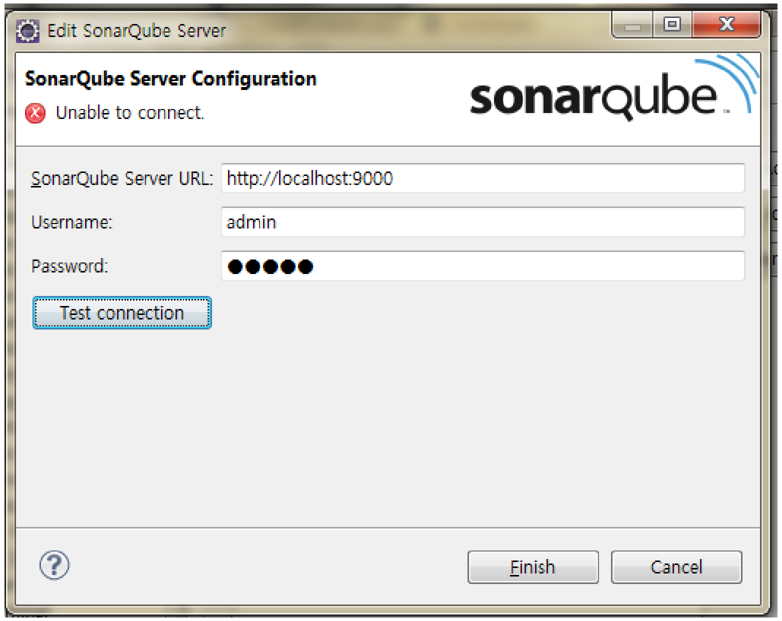
SonarQube Server URL: http://localhost:9000

Username: admin

Password: ●●●●●

- Connect to SonarQube server : [Window]-[preferences]-[SonarQube]-[servers] –[test connection]

How to install Sonar?


Edit SonarQube Server

SonarQube Server Configuration

Unable to connect.

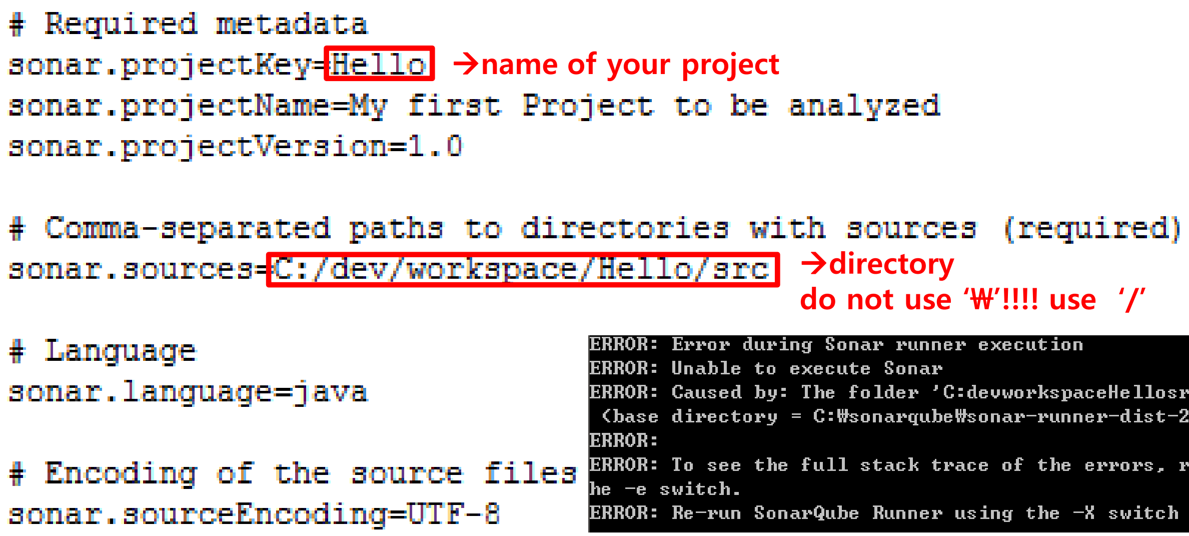
SonarQube Server URL: http://localhost:9000

Username: admin

Password: ●●●●●

- When you connect to SonarQube server…
  - -Open SonarQube server before connecting!
  - - C:\...\sonar-3.7.4\sonar-3.7.4\bin\windows-x86-xx \StartSonar.bat
- If you didn't open the Server…

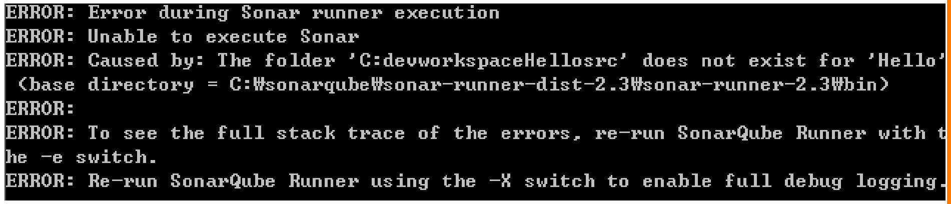
How to install Sonar?



```
# Required metadata
sonar.projectKey=Hello
sonar.projectName=My first Project to be analyzed
sonar.projectVersion=1.0

# Comma-separated paths to directories with sources (required)
sonar.sources=C:/dev/workspace/Hello/src

# Language
sonar.language=java

# Encoding of the source files
sonar.sourceEncoding=UTF-8
```

→name of your project

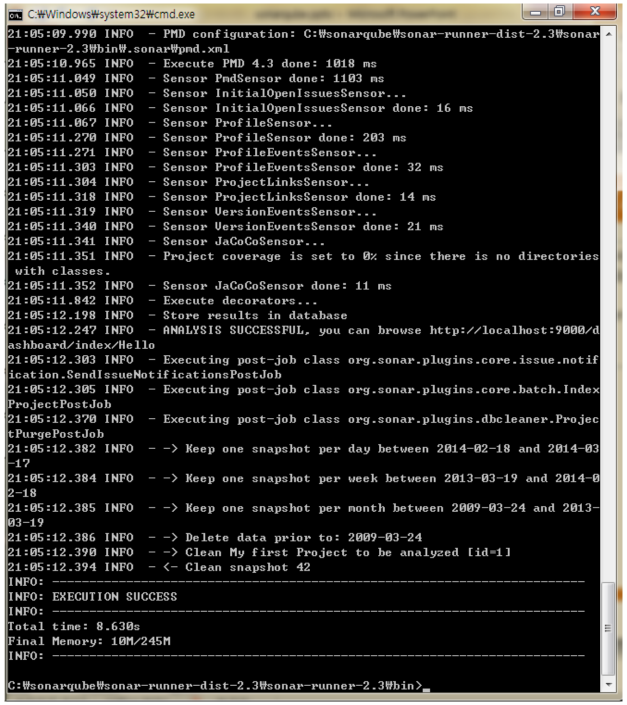
→directory
do not use '₩'!!!! use '/'



```
ERROR: Error during Sonar runner execution
ERROR: Unable to execute Sonar
ERROR: Caused by: The folder 'C:devworkspaceHellosrc' does not exist for 'Hello'
 (base directory = C:₩sonarqube₩sonar-runner-dist-2.3₩sonar-runner-2.3₩bin)
ERROR:
ERROR: To see the full stack trace of the errors, re-run SonarQube Runner with t
he -e switch.
ERROR: Re-run SonarQube Runner using the -X switch to enable full debug logging.
```

- C:\sonarqube\sonar-runner-dist-2.3\sonar-runner-2.3\conf\sonar-runner.properties →Edit the file!!!

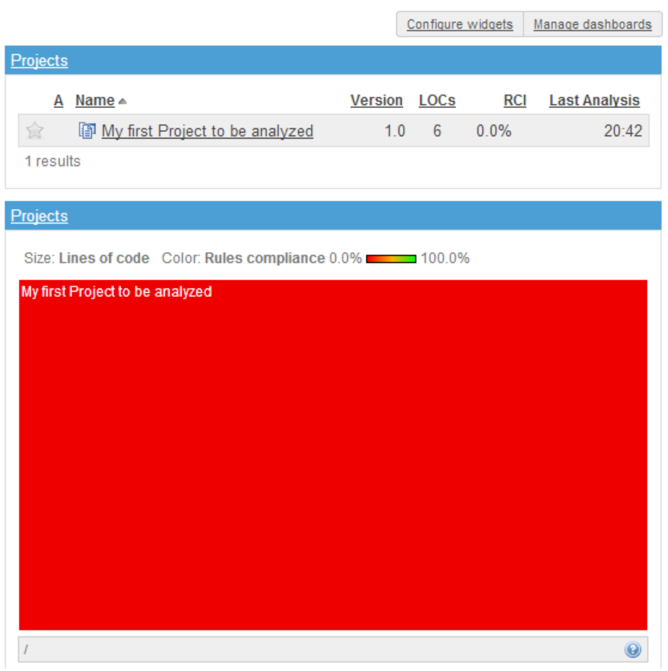
How to use Sonar?



```
21:05:09.990 INFO  - PMD configuration: C:₩sonarqube₩sonar-runner-dist-2.3₩sonar
-runner-2.3₩bin₩.sonar₩pmd.xml
21:05:10.965 INFO  - Execute PMD 4.3 done: 1018 ms
21:05:11.049 INFO  - Sensor PmdSensor done: 1103 ms
21:05:11.050 INFO  - Sensor InitialOpenIssuesSensor...
21:05:11.066 INFO  - Sensor InitialOpenIssuesSensor done: 16 ms
21:05:11.067 INFO  - Sensor ProfileSensor...
21:05:11.270 INFO  - Sensor ProfileSensor done: 203 ms
21:05:11.271 INFO  - Sensor ProfileEventsSensor...
21:05:11.303 INFO  - Sensor ProfileEventsSensor done: 32 ms
21:05:11.304 INFO  - Sensor ProjectLinksSensor...
21:05:11.318 INFO  - Sensor ProjectLinksSensor done: 14 ms
21:05:11.319 INFO  - Sensor VersionEventsSensor...
21:05:11.340 INFO  - Sensor VersionEventsSensor done: 21 ms
21:05:11.341 INFO  - Sensor JaCoCoSensor...
21:05:11.351 INFO  - Project coverage is set to 0% since there is no directories
 with classes.
21:05:11.352 INFO  - Sensor JaCoCoSensor done: 11 ms
21:05:11.842 INFO  - Execute decorators...
21:05:12.198 INFO  - Store results in database
21:05:12.247 INFO  - ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/d
ashboard/index/Hello
21:05:12.303 INFO  - Executing post-job class org.sonar.plugins.core.issue.notif
ication.SendIssueNotificationsPostJob
21:05:12.305 INFO  - Executing post-job class org.sonar.plugins.core.batch.Index
ProjectPostJob
21:05:12.370 INFO  - Executing post-job class org.sonar.plugins.dbcleaner.Projec
tPurgePostJob
21:05:12.382 INFO  - -> Keep one snapshot per day between 2014-02-18 and 2014-03
-17
21:05:12.384 INFO  - -> Keep one snapshot per week between 2013-03-19 and 2014-0
2-18
21:05:12.385 INFO  - -> Keep one snapshot per month between 2009-03-24 and 2013-
03-19
21:05:12.386 INFO  - -> Delete data prior to: 2009-03-24
21:05:12.390 INFO  - -> Clean My first Project to be analyzed [id=1]
21:05:12.394 INFO  - <- Clean snapshot 42
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
Total time: 8.630s
Final Memory: 10M/245M
INFO: ------------------------------------------------------------------------

C:₩sonarqube₩sonar-runner-dist-2.3₩sonar-runner-2.3₩bin>_
```

- C:\sonarqube\sonar-runner-dist-2.3\sonar-runner-2.3\bin
- Execute sonar-runner

How to use Sonar?



Configure widgets | Manage dashboards

Projects

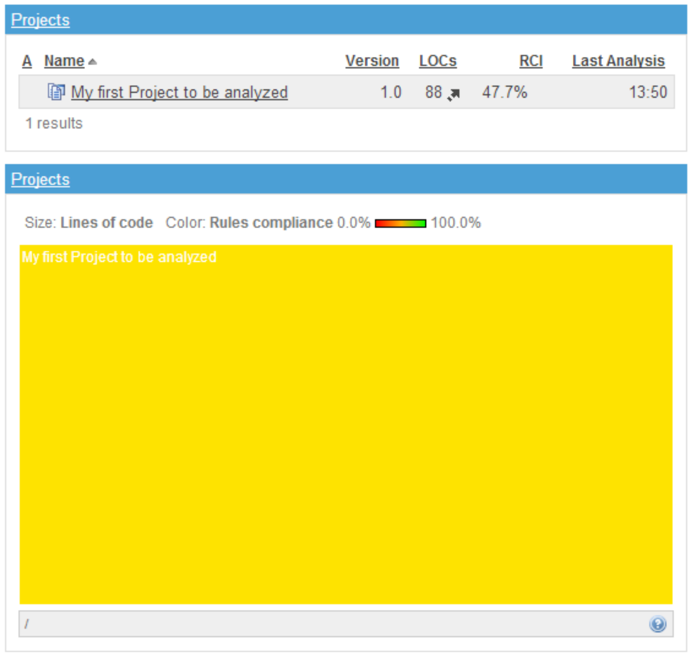
A	Name	Version	LOCs	RCI	Last Analysis
☆	My first Project to be analyzed	1.0	6	0.0%	20:42

1 results

Projects

Size: Lines of code Color: Rules compliance 0.0% 100.0%

My first Project to be analyzed



Projects

A	Name	Version	LOCs	RCI	Last Analysis
	My first Project to be analyzed	1.0	88	47.7%	13:50

1 results

Projects

Size: Lines of code Color: Rules compliance 0.0% 100.0%

My first Project to be analyzed

- open your browser and access to http://localhost:9000

→ Login with ID : admin / PW : admin

- It would be closer to red when there's many error.

How to use Sonar?

01 JUnit

02 Eclipse

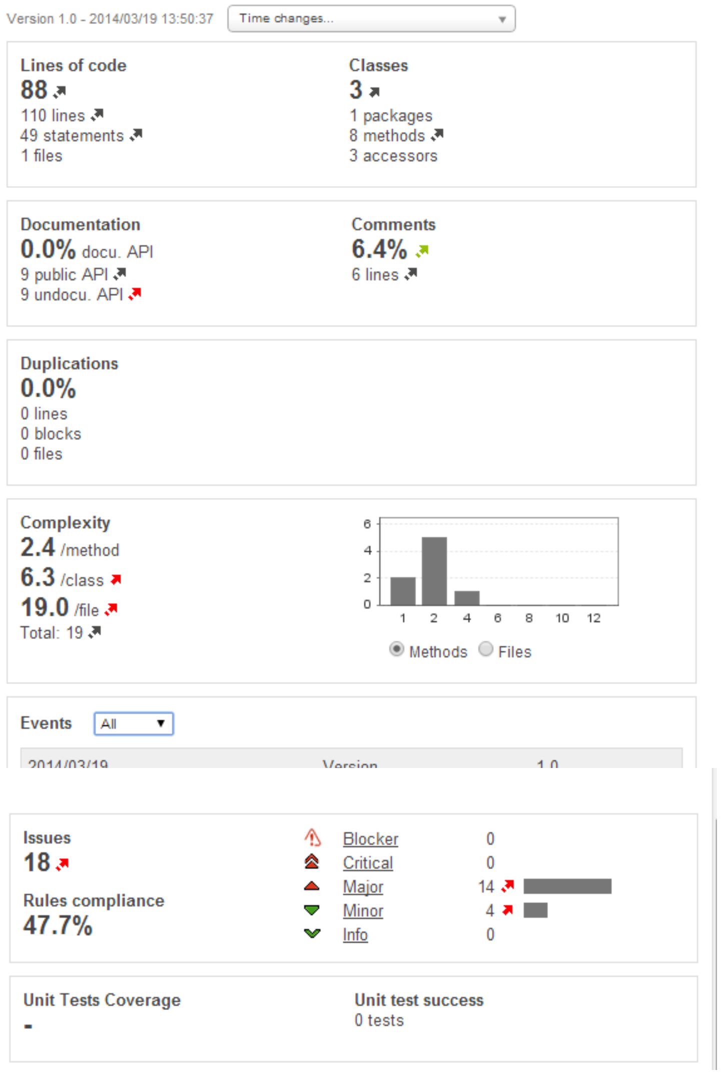
03 CheckStyle

04 PMD

05 FindBugs

06 SONAR

- Dashboard shows the result in terms of 7 standards of quality.
  - potential bugs
  - coding rule
  - test
  - duplication
  - comments
  - architecture and design
  - complexity



Version 1.0 - 2014/03/19 13:50:37 | Time changes...

Lines of code
88
110 lines
49 statements
1 files

Classes
3
1 packages
8 methods
3 accessors

Documentation
0.0% docu. API
9 public API
9 undocu. API

Comments
6.4%
6 lines

Duplications
0.0%
0 lines
0 blocks
0 files

Complexity
2.4 /method
6.3 /class
19.0 /file
Total: 19

Events All

2014/03/19 Version 1.0

Issues
18

Rules compliance
47.7%

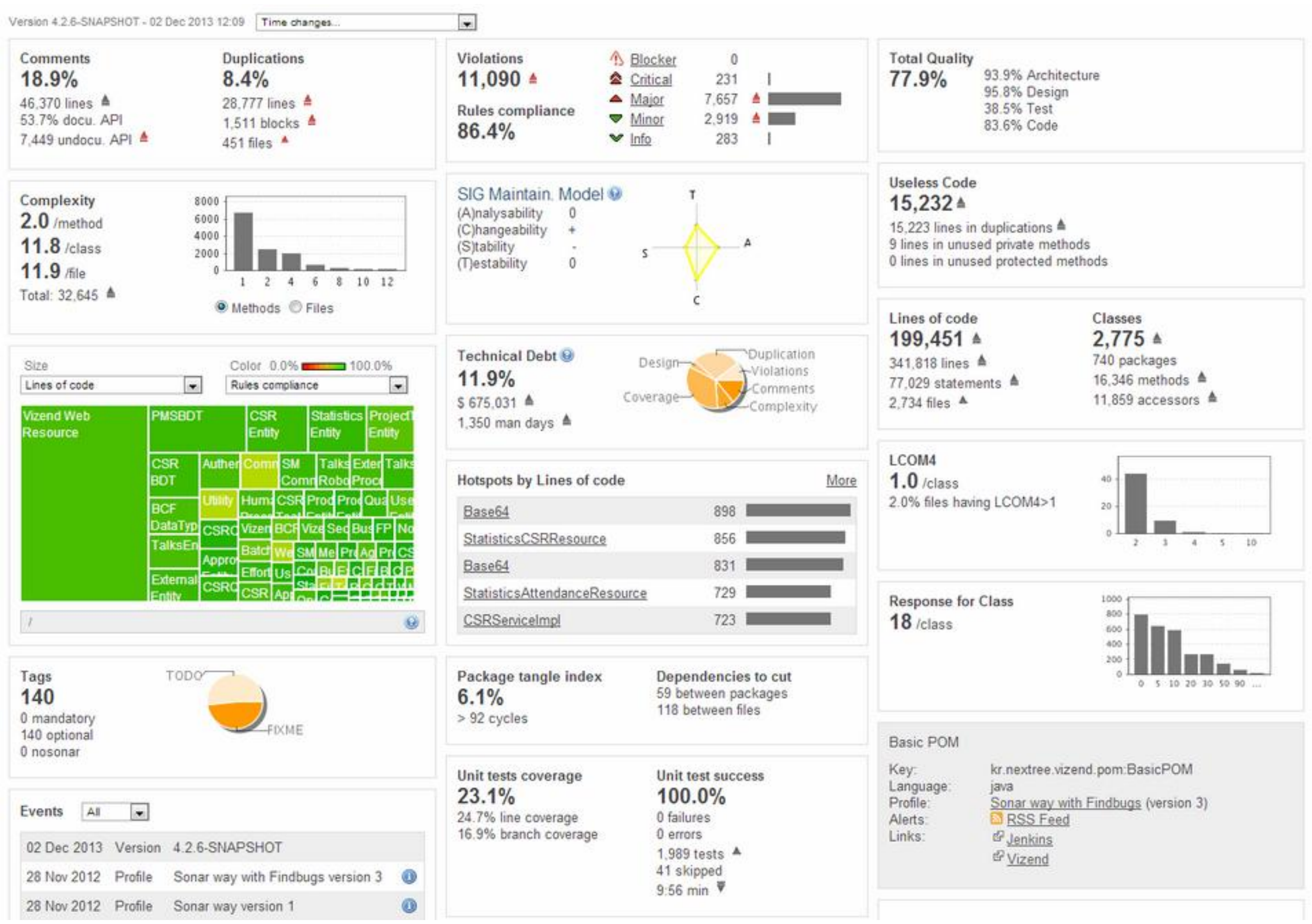
Blocker	0
Critical	0
Major	14
Minor	4
Info	0

Unit Tests Coverage
-

Unit test success
0 tests

How to use Sonar?

- Dashboard (with plugin example)



Version 4.2.6-SNAPSHOT - 02 Dec 2013 12:09 | Time changes...

Comments
18.9%
46,370 lines
53.7% docu. API
7,449 undocu. API

Duplications
8.4%
28,777 lines
1,511 blocks
451 files

Violations
11,090

Rules compliance
86.4%

Blocker	0	
Critical	231	
Major	7,657	
Minor	2,919	
Info	283	

Total Quality
77.9%
93.9% Architecture
95.8% Design
38.5% Test
83.6% Code

Complexity
2.0 /method
11.8 /class
11.9 /file
Total: 32,645

SIG Maintain. Model
(A)nalysability 0
(C)hangeability +
(S)tability -
(T)estability 0

Useless Code
15,232
15,223 lines in duplications
9 lines in unused private methods
0 lines in unused protected methods

Size: Lines of code
Color 0.0% 100.0%: Rules compliance

Technical Debt
11.9%
$ 675,031
1,350 man days

Lines of code
199,451
341,818 lines
77,029 statements
2,734 files

Classes
2,775
740 packages
16,346 methods
11,859 accessors

Hotspots by Lines of code — More

Base64	898
StatisticsCSRResource	856
Base64	831
StatisticsAttendanceResource	729
CSRServiceImpl	723

LCOM4
1.0 /class
2.0% files having LCOM4>1

Response for Class
18 /class

Tags
140
0 mandatory
140 optional
0 nosonar

Package tangle index
6.1%
> 92 cycles

Dependencies to cut
59 between packages
118 between files

Basic POM

Key:	kr.nextree.vizend.pom:BasicPOM
Language:	java
Profile:	Sonar way with Findbugs (version 3)
Alerts:	RSS Feed
Links:	Jenkins
	Vizend

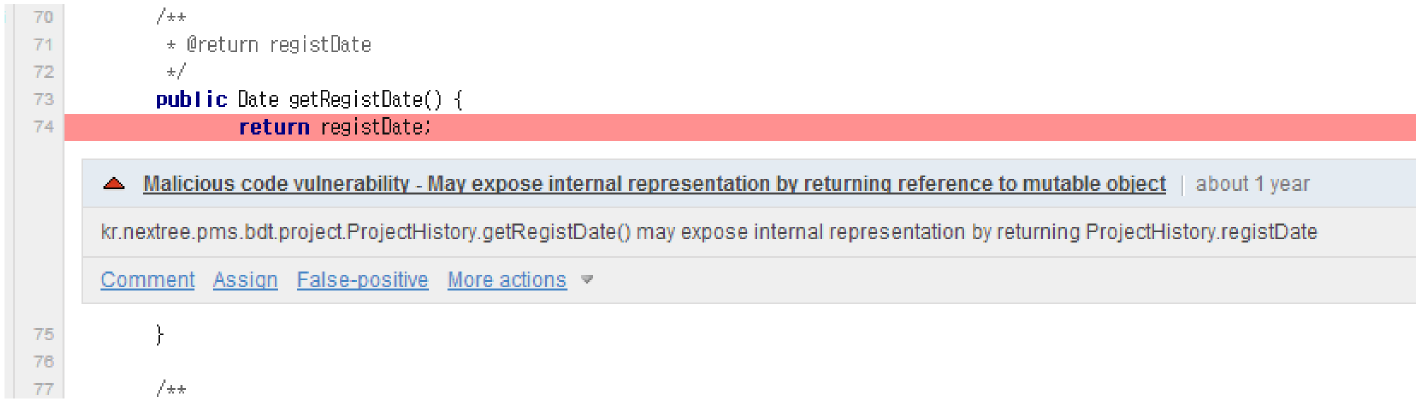
Events | All

02 Dec 2013	Version	4.2.6-SNAPSHOT	
28 Nov 2012	Profile	Sonar way with Findbugs version 3	
28 Nov 2012	Profile	Sonar way version 1	

Unit tests coverage
23.1%
24.7% line coverage
16.9% branch coverage

Unit test success
100.0%
0 failures
0 errors
1,989 tests
41 skipped
9:56 min

How to use Sonar?



```
    /**
     * @return registDate
     */
    public Date getRegistDate() {
        return registDate;
```

Malicious code vulnerability - May expose internal representation by returning reference to mutable object | about 1 year

kr.nextree.pms.bdt.project.ProjectHistory.getRegistDate() may expose internal representation by returning ProjectHistory.registDate

```
    }

    /**
```

- Code comment

sonarqube

[How to solve the error…]

one of your sonarqube server cannot be reached. please check your connection settings.	서버가 열리지 않아서 생기는 에러 StartSonar.bat으로 서버 실행
authentication error	이클립스 windows-preferences에서 ID/pw가 일치하지 않으면 나는 에러 서버 로그인 후 admin/admin으로 로그인
The folder ‘C:devworkspaceHellosrc’ does not exist for ‘Hello‘	Sonar-runner.properties 수정시 \가 아닌 /로 경로 설정

sonarqube

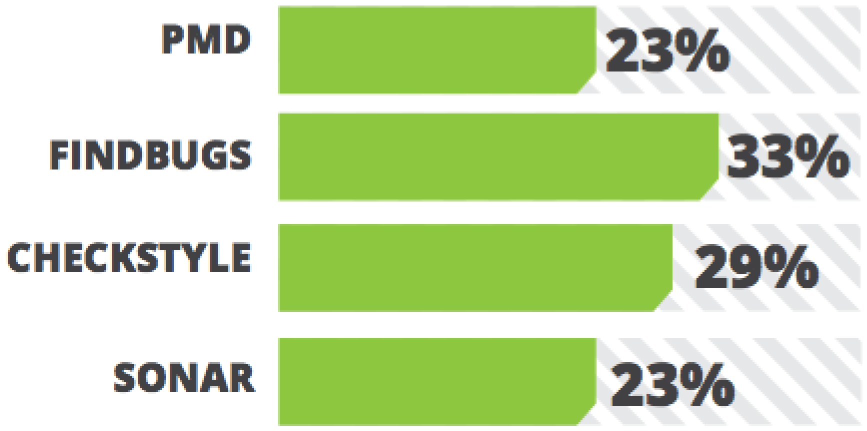
After using sonar…

- **SonarQube**의 장점
  - **dashboard**가 항목별로 정리되어있어 이해하고 알아보기 쉬움
  -다른 **plugin**을 추가 설치한다면 더욱 많은 분석 결과를 확인할 수 있음
  -또한 코드별로 코멘트를 달아 팀원 별로 의견 공유를 할 수 있음

- **SonarQube**의 단점
  -eclipse 내에서 분석 결과를 확인 할 수 있는 것이 아니라 따로 인터넷 페이지에서 확인해야함
  -server를 켜놓고 **cmd**창에서 **runner**를 돌려야 하는 점 때문에 불편함

- 어려웠던 점 및 소감
  -설치의 어려움

Comparisons between S.A tools



sonarqube

Comparisons between S.A tools

	Checkstyle	PMD	FindBugs	SONAR
Good	-지속적으로 업데이트 -**Code convention**을 지키게끔 연습하는데 가장 유용. -한글로 된 자료가 많음	-엄청나게 꼼꼼 -분석에 오랜 시간이 소요되지 않음 -**CPD** 분석기능 제공 -여러가지 리포트 포맷을 제공	-툴이 가벼움 -자체 **UI** 제공으로 사용이 편리 -잠재적 버그를 찾는데 가장 유용 -바이트코드 분석으로 호출되지 않는 영역에러 찾음	-오라클,**MySQL** 같은 **DB**와 연동 가능 -복잡도 분석 같은 다른 툴에는 없는 기능 제공 -일자별 로 변동을 파악가능
Bad	-진짜 버그는 못찾을 수도 있음(**code convention**을 준수했나를 가장 중점적으로 보는 툴!)	-툴이 조금 무거운편 -자체 **UI**를 제공하지 않아 이클립스 없이 쓰려면 힘들다 -쓰면서 가장 에러가 많이 난 툴.	-툴이 가벼움에도 불구하고 분석에 **PMD**보다 오랜 시간이 소요됨 -리포트 포맷이 다양하지 않음	-인스톨 어려움 -서버가 항상 틀어져 있어야함. -이클립스가 아닌 웹상에서 확인해야함.

Q&A

Thank you

JUnit Reference

JUnit in Action 단위테스트의 모든 것**(**인사이트**)**

이클립스 프로젝트 필수 유틸리티 **Subversion, Ant, JUnit, Trac(**한빛미디어**)**

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

http://junit.org/

http://www.javajigi.net/pages/viewpage.action?pageId=278

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

CheckStyle Reference

http://checkstyle.sourceforge.net/
http://eclipse-cs.sourceforge.net/
http://netframework.tistory.com/363
http://jseaknu.egloos.com/466646
http://zeroturnaround.com/rebellabs/code-quality-tools-review-for-2013-sonar-findbugs-pmd-and-checkstyle/

PMD Reference

자바 개발자도 쉽고 즐겁게 배우는 테스팅 이야기(한빛미디어)

http://cafe.naver.com/tbed/16

http://stackoverflow.com/questions/16778062/importing-rules-for-pmd-eclipse

http://cafe.naver.com/tbed/16

http://www.swbank.kr/html/tools/toolsFile/PMD_05_functionIntro.pdf

FindBugs Reference

자바 개발자도 쉽고 즐겁게 배우는 테스팅 이야기(한빛미디어)

http://findbugs.sourceforge.net/manual/gui.html

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

http://stackoverflow.com/questions/6395546
/findbugs-not-showing-the-bugs-found

Sonar Reference

http://www.sonarqube.org/

http://www.slideshare.net/allnewangel/sonar-23151331

http://dryang.egloos.com/viewer/4005366