

# 2014 소프트웨어 검증

Eclipse, JUnit, 정적분석도구

T1

200911381 김진현

200911417 정명권

200911418 정세진

1. Eclipse
2. JUnit
3. STATIC ANALYSIS
4. PMD
5. Eclipse TPTP
6. FIND BUGS

# INDEX

IDE

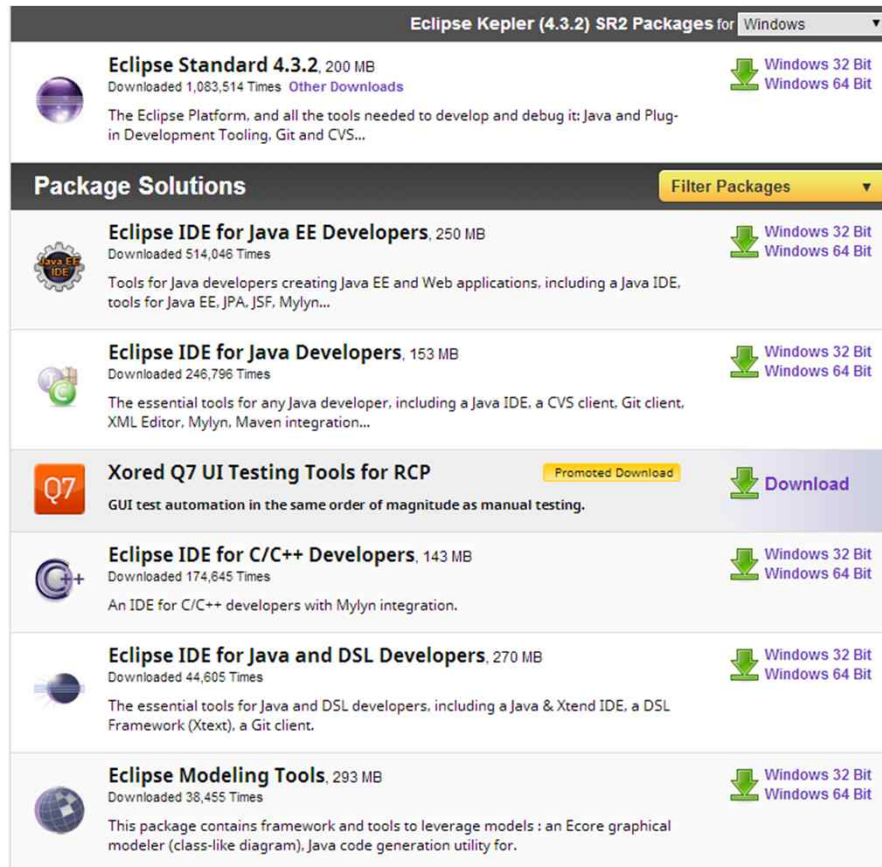
# ECLIPSE

# Eclipse

- ❖ Ide의 한 종류
- ❖ 주로 jdk를 설치하여 java코드 작성에 이용
- ❖ c/c++, android 등의 다양한 언어 지원

# Eclipse - 설치

- ❖ <http://eclipse.org> 에서 다운로드
- ❖ Os 선택 후, 맞는 버전에 따라서 다운로드 후 압축 해제



Eclipse Kepler (4.3.2) SR2 Packages for Windows

**Eclipse Standard 4.3.2**, 200 MB  
Downloaded 1,083,514 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...

**Package Solutions** [Filter Packages](#)

**Eclipse IDE for Java EE Developers**, 250 MB  
Downloaded 514,046 Times  
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

**Eclipse IDE for Java Developers**, 153 MB  
Downloaded 246,796 Times  
The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

**Xored Q7 UI Testing Tools for RCP** [Promoted Download](#)  
GUI test automation in the same order of magnitude as manual testing.

**Eclipse IDE for C/C++ Developers**, 143 MB  
Downloaded 174,645 Times  
An IDE for C/C++ developers with Mylyn integration.

**Eclipse IDE for Java and DSL Developers**, 270 MB  
Downloaded 44,605 Times  
The essential tools for Java and DSL developers, including a Java & Xtend IDE, a DSL Framework (Xtext), a Git client.

**Eclipse Modeling Tools**, 293 MB  
Downloaded 38,455 Times  
This package contains framework and tools to leverage models : an Ecore graphical modeler (class-like diagram), Java code generation utility for.

# Eclipse


- ❖ Jdk 설치
- ❖ <http://oracle.com> 에서 downloads -> java for developer 항목에서 jdk 다운로드 후 설치


Products Solutions Downloads Store Support Training Partners

java > Java SE > Downloads

Overview Downloads Documentation Community Technologies Training

### Java SE Downloads

 **DOWNLOAD** ▾  
Java Platform (JDK) 8

 **DOWNLOAD** ▾  
JDK 8 & NetBeans 8.0

#### Java Platform, Standard Edition

**Java SE 8**  
This new major release contains several new features and enhancements that increase the performance of existing applications, make it easier to develop applications for modern platforms, and increase maintainability of code.  
[Learn more](#) ▶

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
  - JDK ReadMe
  - JRE ReadMe

**JDK 8**  
**DOWNLOAD** ▾

**Server JRE 8**  
**DOWNLOAD** ▾

**JRE 8**  
**DOWNLOAD** ▾

### Java SE Development Kit 8

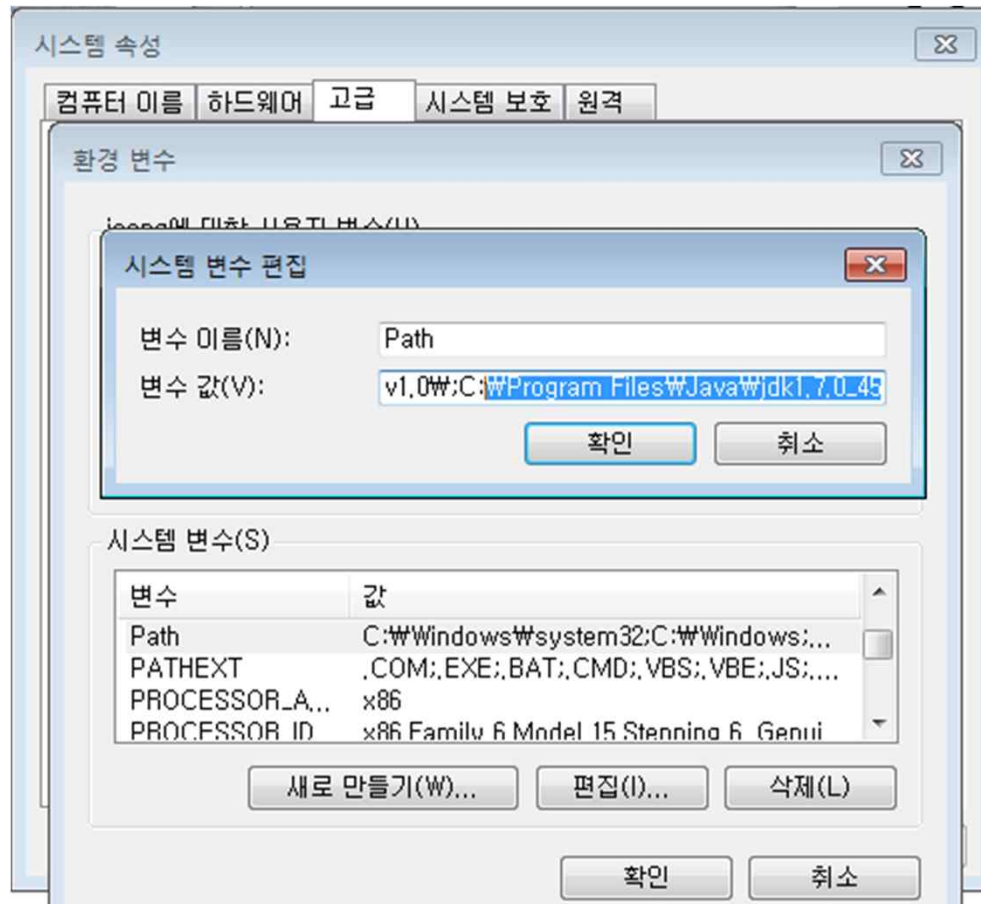
You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement  Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	83.51 MB	 <a href="#">jdk-8-linux-arm-vfp-hflt.tar.gz</a>
Linux x86	133.57 MB	 <a href="#">jdk-8-linux-i586.rpm</a>
Linux x86	152.47 MB	 <a href="#">jdk-8-linux-i586.tar.gz</a>
Linux x64	133.85 MB	 <a href="#">jdk-8-linux-x64.rpm</a>
Linux x64	151.61 MB	 <a href="#">jdk-8-linux-x64.tar.gz</a>
Mac OS X x64	207.72 MB	 <a href="#">jdk-8-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	135.5 MB	 <a href="#">jdk-8-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	95.53 MB	 <a href="#">jdk-8-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	135.78 MB	 <a href="#">jdk-8-solaris-x64.tar.Z</a>
Solaris x64	93.15 MB	 <a href="#">jdk-8-solaris-x64.tar.gz</a>
Windows x86	151.68 MB	 <a href="#">jdk-8-windows-i586.exe</a>
Windows x64	155.14 MB	 <a href="#">jdk-8-windows-x64.exe</a>

# Eclipse

- ❖ 제어판\시스템 및 보안\시스템\고급 시스템 설정에서 jdk설치된 위치로 환경변수 설정

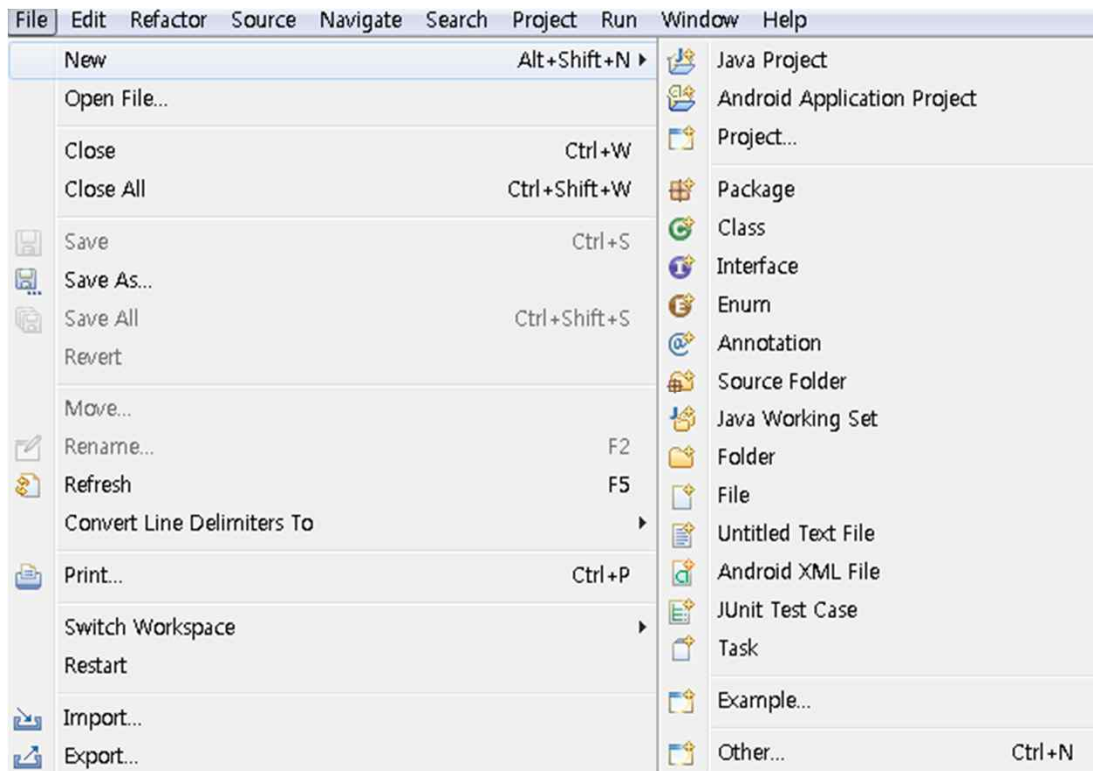


# Eclipse - 사용법

## ❖ 프로젝트 생성

❖ File -> new -> Java project

❖ 같은 방식으로 package, class 등 생성 가능

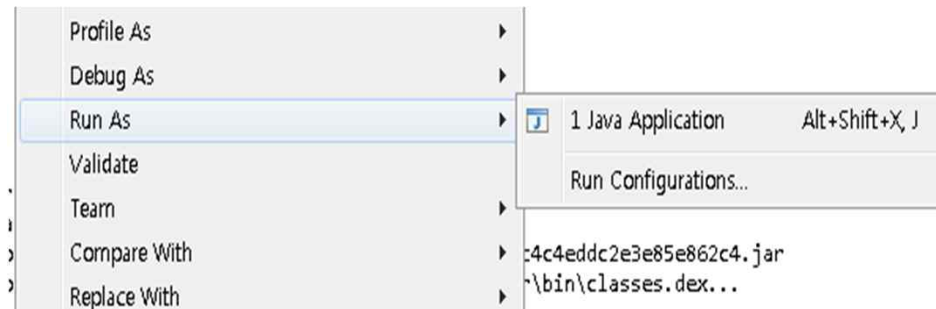
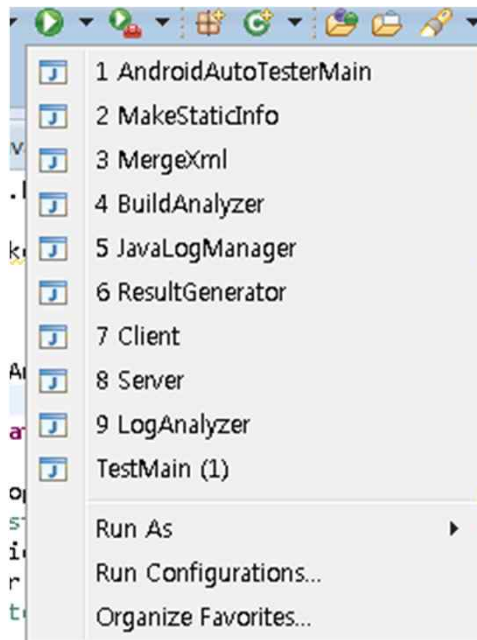




# Eclipse

## ❖ 실행

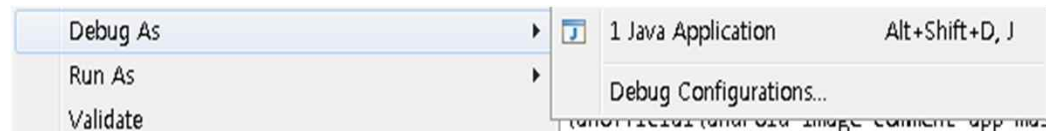
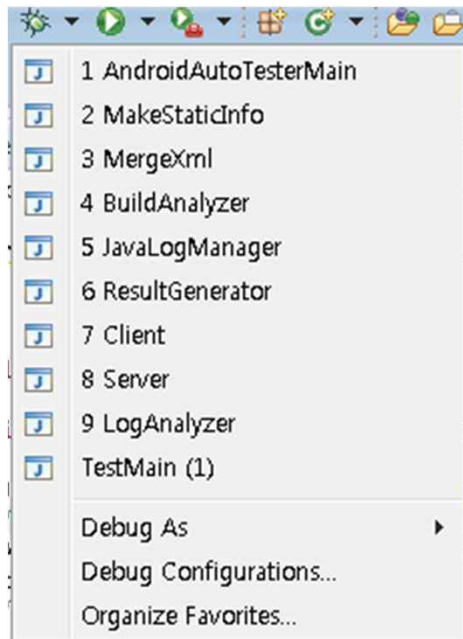
- ❖ 상단의 화살표 모양의 버튼을 누르고 실행시킬 프로젝트 선택
- ❖ 마우스 오른쪽 버튼 누르고 run as 선택



# Eclipse

## ❖ 디버깅

- ❖ 상단의 별레 모양의 버튼을 누르고 실행시킬 프로젝트 선택
- ❖ 마우스 오른쪽 버튼 누르고 debug as 선택

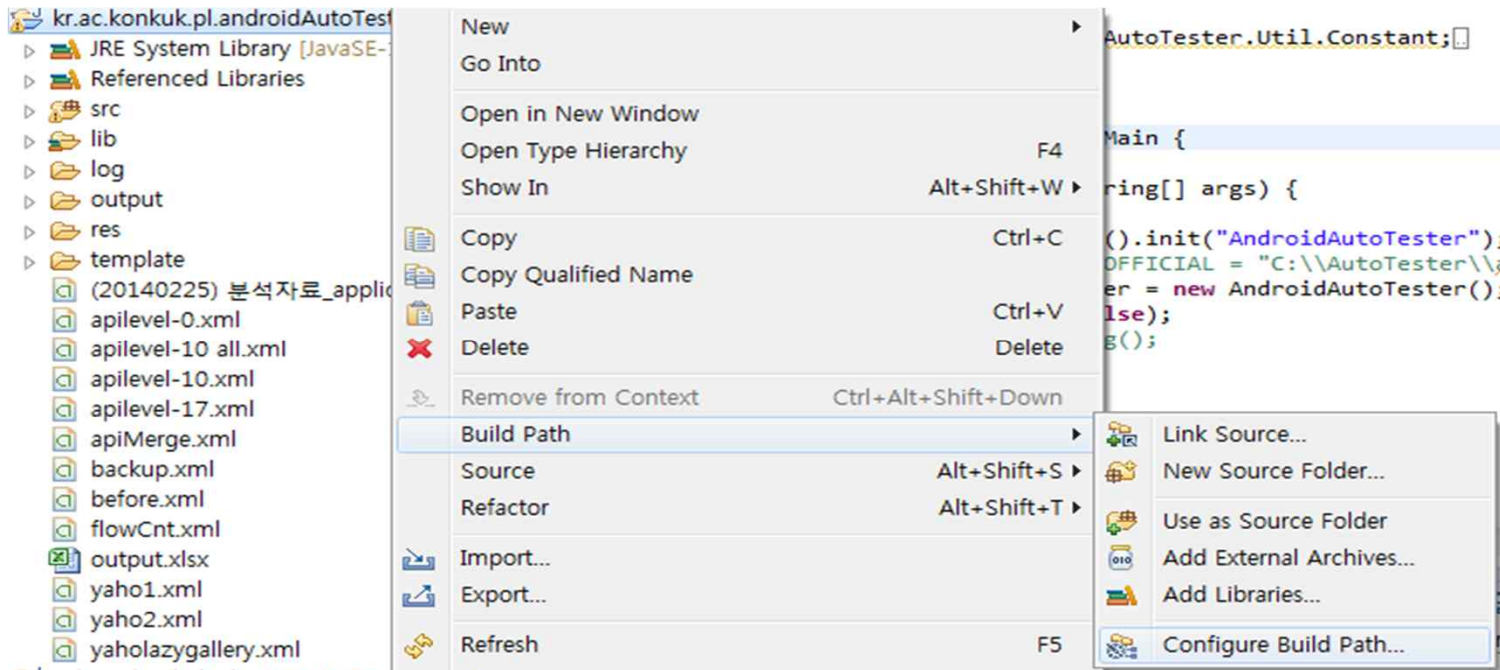


# Eclipse - 라이브러리 사용

## ❖ 외부 라이브러리 사용

❖ Eclipse에서 project에 jar 형태의 외부 라이브러리를 추가하여 사용 가능

❖ 프로젝트 이름에 마우스 오른쪽 클릭 - > buildpath -> configuration  
build path에서 관리



# Eclipse

Properties for kr.ac.konkuk.pl.androidAutoTesterFinal

Java Build Path

JARs and class folders on the build path:

- AndroidMarketApi.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- apache-log4j-extras-1.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- apktool\_2.0.0b7.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- asm-all-3.3.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- commons-beanutils-1.8.2.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- commons-collections-3.2.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- commons-digester-1.7.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- commons-exec-1.2.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- commons-io-2.0.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- commons-logging-1.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- dex-ir-1.12.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- dex-reader-1.15.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- dex-tools-0.0.9.15.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- dex-translator-0.0.9.15.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- diffutils-1.2.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- dom4j-1.6.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- dx.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- freemarker.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- jcommon-1.0.21.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- jfreechart-1.0.17.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- kr.ac.konkuk.pl.autoTesterModel.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- kr.ac.konkuk.pl.javaDebugLib.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- kr.ac.konkuk.pl.portabilityAnalyzer.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- kr.ac.konkuk.pl.testcasegen.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- log4j-1.2.16.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.contenttype\_3.4.200.v20130326-1255.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.jobs\_3.5.300.v20130429-1813.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.jobs.source\_3.5.300.v20130429-1813.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.resources\_3.8.101.v20130717-0806.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.resources.win32.x86\_64\_3.5.0.v20121203-0906.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.runtime\_3.9.0.v20130326-1255.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.runtime.compatibility\_3.2.200.v20130326-1255.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.core.runtime.compatibility.registry.source\_3.5.200.v20130514-1256.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.equinox.common\_3.6.200.v20130402-1505.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.equinox.preferences\_3.5.100.v20130422-1538.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.jdt.core\_3.9.1.v20130905-0837.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.jdt.core.manipulation\_1.5.0.v20130605-1748.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.osgi\_3.9.1.v20130814-1242.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.osgi.services\_3.3.100.v20130513-1956.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.osgi.util\_3.2.300.v20130513-1956.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- org.eclipse.text\_3.5.300.v20130515-1451.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- poi-3.9-20121203.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- poi-ooxml-3.9-20121203.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- poi-ooxml-schemas-3.9-20121203.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib
- protobuf-java-2.4.1.jar - kr.ac.konkuk.pl.androidAutoTesterFinal/lib

Buttons: Add JARs..., Add External JARs..., Add Variable..., Add Library..., Add Class Folder..., Add External Class Folder..., Edit..., Remove, Migrate JAR File...

2014 Software Verfication T1

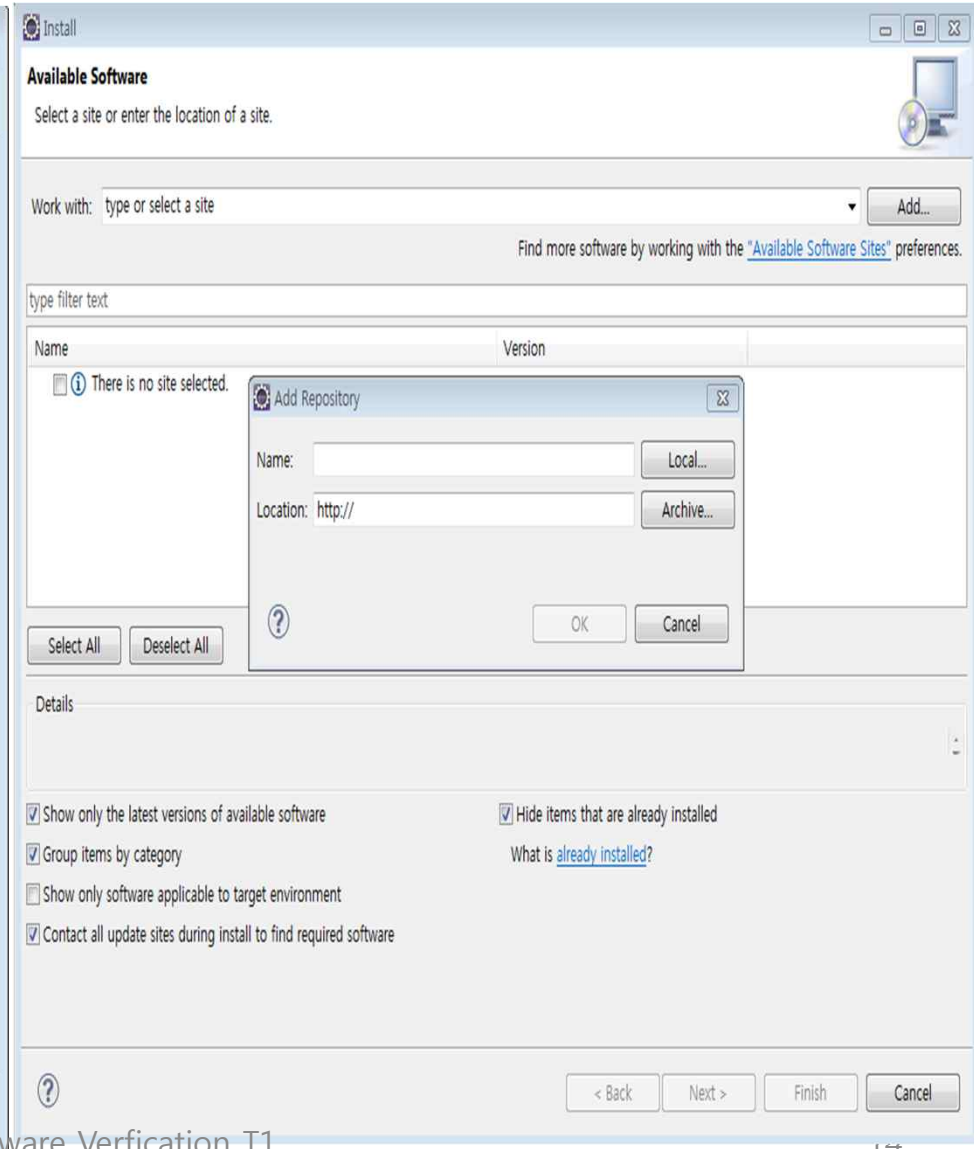
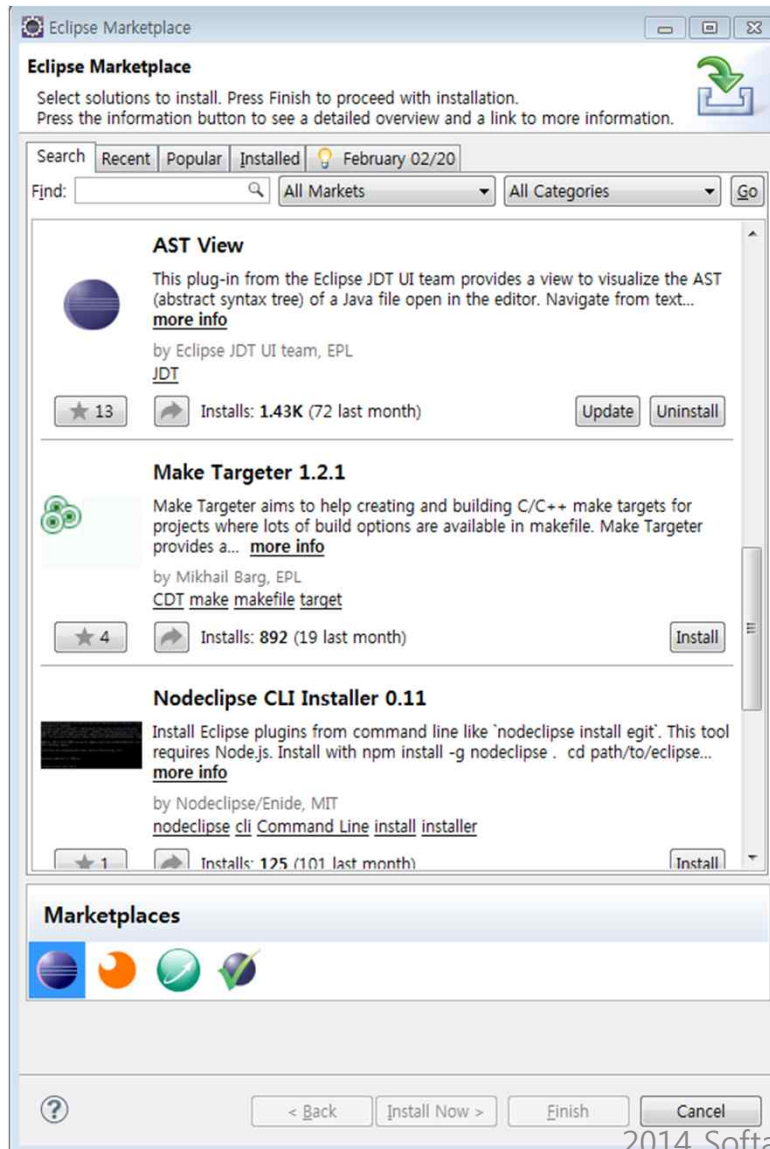
OK Cancel

# Eclipse - 플러그인

## ❖ 플러그인

- ❖ 시스템에 기능을 추가하는 코드/데이터의 구조화된 묶음
- ❖ 추가되는 기능은 코드 라이브러리, 플랫폼 확장, 문서 등의 형태로 제공
- ❖ Help->install new software 메뉴에서 url을 이용하여 설치
- ❖ Help->eclipse marketplace 에서 검색을 통해 plugin을 찾아서 설치 가능
- ❖ 대표적인 예로는 subclipse(svn), ast view, android sdk 등이 있다

# Eclipse



자바를 위한 단위 테스트 작성 프레임워크

**JUNIT**

# JUnit

## ❖ JUnit

- Java Unit Test 작성을 위한 산업 표준 프레임워크

## ❖ Unit Test

- 프로그램을 작은 단위로 쪼개고 각 단위가 제대로 동작하는지 검증하는 것
- 문제가 발생시 그 원인이 되는 단위를 빠르게 찾고 해결할 수 있음

Unit Tests

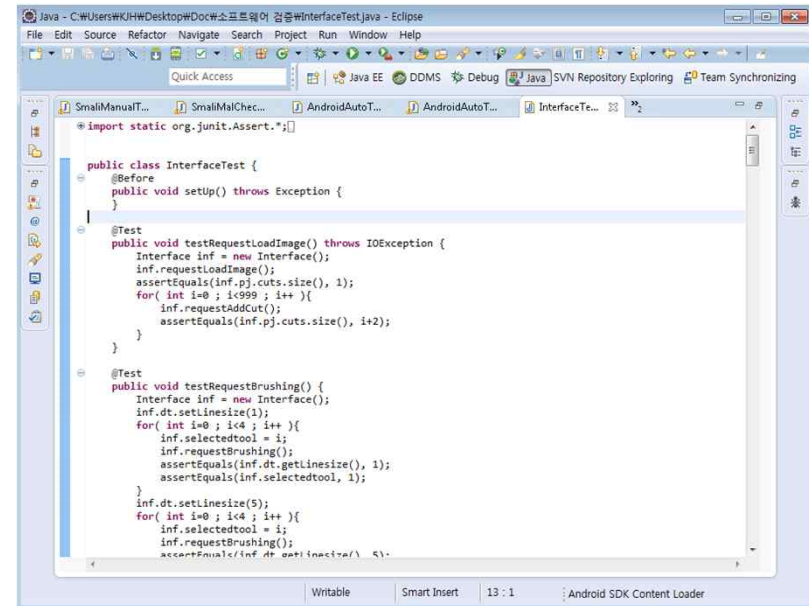
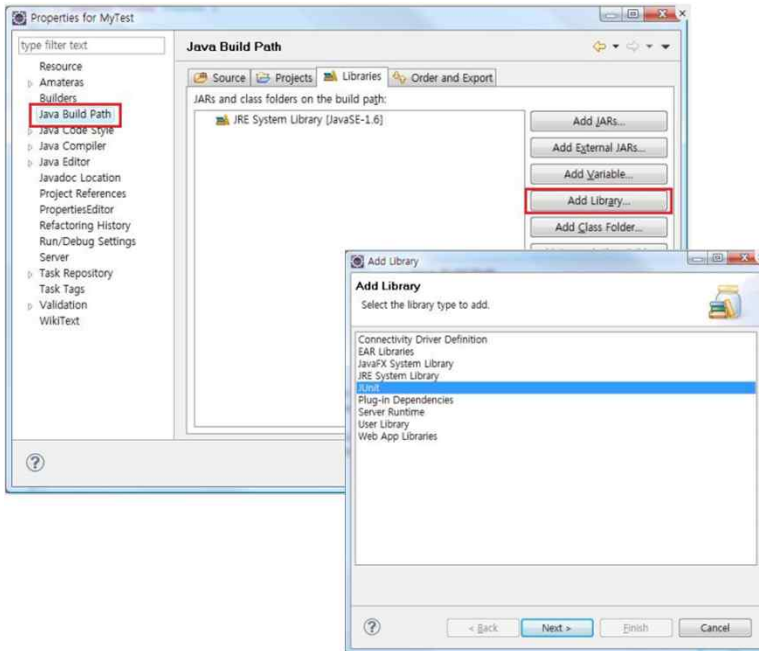




# JUnit

## ❖ 동작 환경 구성

- Eclipse 설치 (JUnit은 기본적으로 포함되어 있음)
- 테스트를 수행할 프로젝트의 Library에 JUnit를 추가
- 테스트 클래스를 생성하고 테스트 코드를 작성



# JUnit

## ❖ 테스트 코드 작성

- 테스트 클래스를 생성
- 단위 테스트 메소드를 생성
- 메소드 내부에서 테스트 대상 클래스의 인스턴스를 선언
- 인자값을 설정하고 메소드를 실행시켜 결과를 가져옴
- JUnit 프레임워크에 포함된 Assert 메소드들을 이용하여 결과를 확인

```
@Test
public void testRequestSetColor() {
    Interface inf = new Interface();
    for( int i=0 ; i<16 ; i++ ){
        inf.color = i;
        for( int j=0 ; j<16 ; j++ ){
            inf.dt.getBrush().setColor(j);
            for( int k=0 ; k<16 ; k++ ){
                inf.requestSetColor(k);
                assertEquals(inf.dt.getBrush().getColor(),k);
            }
        }
    }
}
```

# JUnit

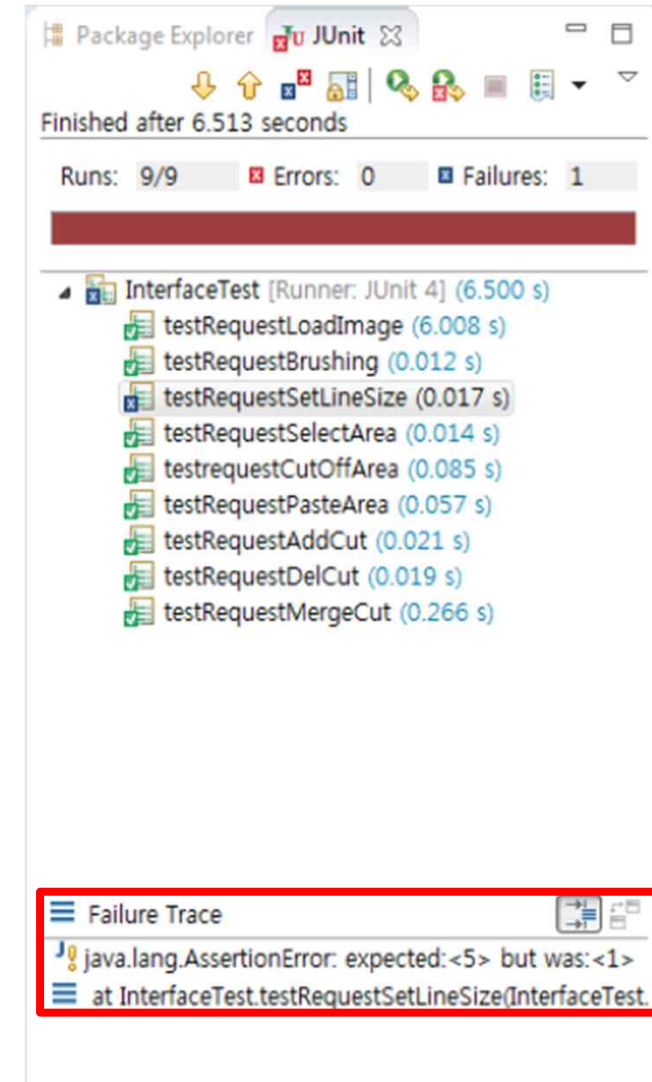
## ❖ JUnit Assert 주요 메소드

- `assertArrayEquals(a, b)` : 배열의 값이 일치하는지 확인
- `assertEquals(a, b)` : 객체 a와 b가 일치하는지 확인
- `assertTrue(a)` : a가 참인지 확인
- `assertNotNull(a)` : a가 Null이 아닌지 확인
- etc : <http://junit.sourceforge.net/javadoc/org/junit/Assert.html>

# JUnit

## ❖ 결과 확인

- 문제가 되는 단위만 확인하여 빠른 오류 해결이 가능
- 한번 테스트 케이스 작성 후 지속적으로 활용이 가능
- 오류 발생 시 목표 값과 실제 값을 비교하여 그 원인을 쉽게 파악할 수 있음
- 테스트 코드 작성시 gui와 연관된 부분에 대해서 테스트 케이스 작성이 쉽지 않음



Introduce Static Analysis

# STATIC ANALYSIS

# Static Analysis

## ❖ What is the Static Analysis?

- 컴퓨터 소프트웨어를 분석하는 방법 중 하나로서 프로그램을 실제로 실행해보지 않고 분석하는 방법이다.
- 여러 정적 분석 도구들을 사용하여 치명적일 수 있는 오류들을 미리 검출한다.
- 프로그램 내부의 논리적 오류를 찾아내는 데 도움을 줄 수 있다.

# Static Analysis

## ❖ Benefit of Static Analysis

- 테스트를 실행하기 전에 조기 결함을 발견할 수 있어 시간 낭비를 줄일 수 있다.
- 코드와 설계의 의심스러운 부분을 조기에 발견할 수 있다.
- 동적 분석으로는 발견하기 어려운 결함을 발견할 수 있다.
- 소프트웨어 모델상의 의존도와 불일치성을 발견할 수 있다.
- 코드와 설계에 있어서 유지 보수성을 향상시킬 수 있다.
- 잠재적인 치명적 결함을 예방할 수 있다.

# Static Analysis

## ❖ Defects found through Static Analysis

- 정의되지 않은 값으로 변수 참조
- 모듈과 컴포넌트 간에 일관되지 않은 인터페이스
- 사용되지 않는 변수나 코드
- 코딩 표준 위반
- 보안의 취약성
- 코드와 모델의 syntax 위반



# Static Analysis

## ❖ Example of Static Analysis Tools

- Eclipse TPTP
- PMD
- Findbugs
- etc...

Static Analysis Tool for Source code













**PMD**

# PMD

- ❖ Source code를 분석하는 정적 분석 도구
- ❖ 코드의 중복성을 체크하는 cpd를 내포하고 있음
- ❖ Unused variables, empty catch block, unnecessary object creation 등 여러 programming flaw를 분석한다
- ❖ 지원 언어로는 java, javascript, xml, xsl등이 있다

# PMD - 설치법

- ❖ <http://pmd.sourceforge.net/> - pmd의 source code 및 release 버전 다운로드 주소
- ❖ Plugin 설치 - 다양한 IDE 지원

IDE	Source Code	Maintainers
<a href="#">BlueJ</a>	<a href="#">pmd-bluej</a> 	<a href="#">Tom Copeland</a> 
<a href="#">CodeGuide</a>	N/A	Austin Moore
<a href="#">Eclipse</a>	<a href="#">pmd-eclipse</a> 	Philippe Herlin
<a href="#">Emacs</a>	<a href="#">pmd-emacs</a> 	Nascif Abousalh Neto
<a href="#">Gel</a>	<a href="#">pmd-gel</a> 	Andrei Lumianski
<a href="#">IntelliJ IDEA</a>	N/A	<a href="#">Amit Dev</a> , <a href="#">Tom Copeland</a> 
<a href="#">IntelliJ IDEA - QAPlug</a>	N/A	Jakub Stawiński
<a href="#">JBuilder</a>	<a href="#">pmd-jbuilder</a> 	<a href="#">Tom Copeland</a> 
<a href="#">JCreator</a>	N/A	Brant Gurganus
<a href="#">JDeveloper</a>	<a href="#">pmd-jdeveloper</a> 	<a href="#">Torsten Kleiber</a> 
<a href="#">JEdit</a>	<a href="#">pmd-jedit</a> 	Jiger Patel
<a href="#">Maven</a>	N/A	N/A
<a href="#">Maven 2</a>	N/A	<a href="#">apache.org / maven</a> 
<a href="#">NetBeans</a>	<a href="#">SQE</a> 	N/A
<a href="#">TextPad</a>	N/A	Jeff Epstein
<a href="#">WebLogic Workshop 8.1.x</a>	N/A	Kevin Conaway

# PMD - 사용법

- ❖ 정해진 규칙(룰) 기반의 정적 분석 도구
- ❖ 정해진 규칙을 위반 여부를 체크
- ❖ 룰의 priority 설정 및 priority 확인이 가능
- ❖ 기존에 정의된 룰을 사용 할 수도 있고 Xml기반의 XPath 방법과 javaRuleSet을 상속받아서 java로 사용자 정의 규칙 생성 가능
- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능

# PMD

- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능

type filter text

- ▶ General
- ▶ Ant
- ▶ 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
- ▶ Team
- Terminal
- Validation
- ▶ Web
- ▶ Web Services
- ▶ XML

### Rule Configuration

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

Rule	Rule set	Type
<input checked="" type="checkbox"/> UselessOperationOnImmutable	Unnecessary	T
<input checked="" type="checkbox"/> UnusedNullCheckInEquals	Unnecessary	X
<input checked="" type="checkbox"/> UselessParentheses	Unnecessary	X
<input checked="" type="checkbox"/> UnusedPrivateField	Unused Code	T
<input checked="" type="checkbox"/> UnusedLocalVariable	Unused Code	T
<input checked="" type="checkbox"/> UnusedLocalVariable	Unused Code	T
<input checked="" type="checkbox"/> UnusedFormalParameter	Unused Code	T
<input checked="" type="checkbox"/> UnusedModifier	Unused Code	T
<input checked="" type="checkbox"/> UseConcatOnce	XPath in XSL	X
<input checked="" type="checkbox"/> AvoidAxisNavigation	XPath in XSL	X

Summary Rule Exclusions

**Name**  
UnusedPrivateField

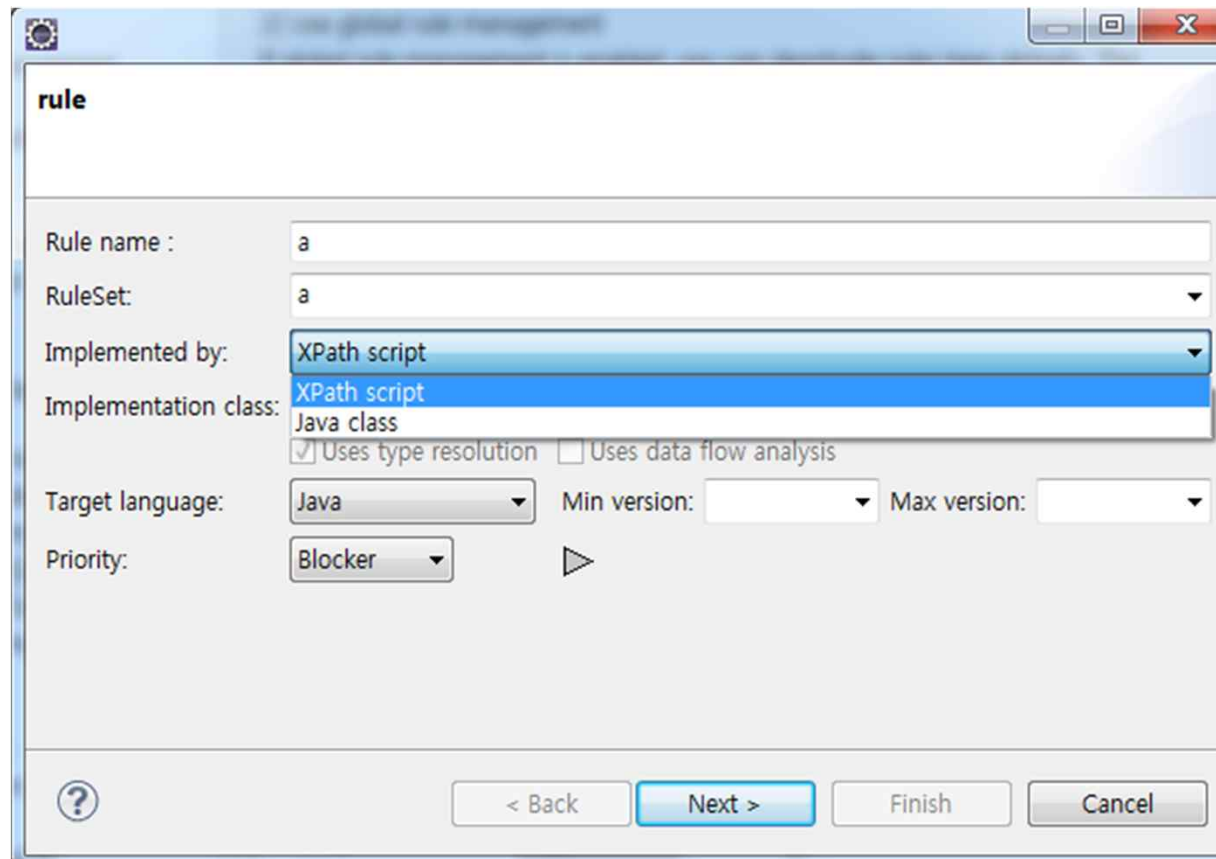
**Description**  
Detects when a private field is declared and/or assigned a value, but not used.  
More information can be found [here](#).

**Example**

```
public class Something {  
    private static int FOO = 2; // Unused  
    private int i = 5; // Unused  
    private int j = 6;  
    public int addOne() {  
        return j++;  
    }  
}
```

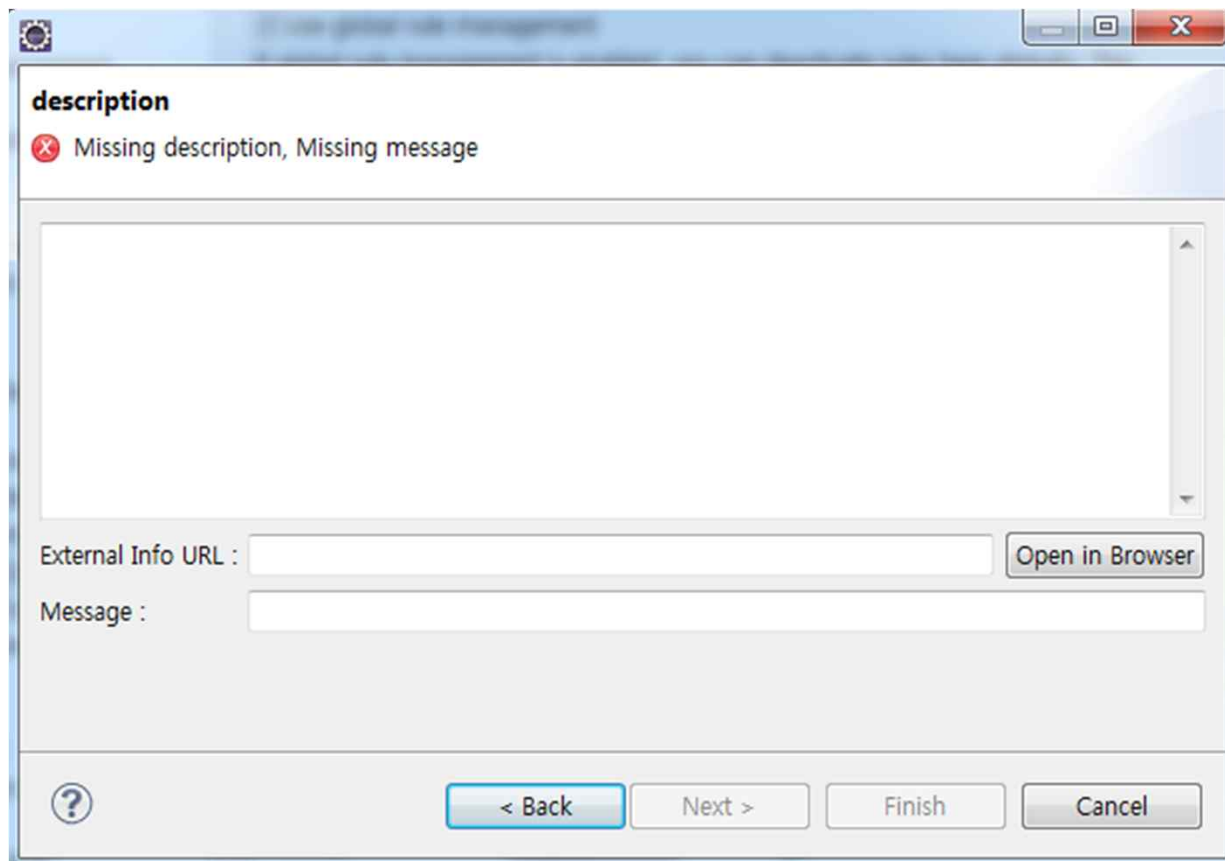
# PMD

- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능



# PMD

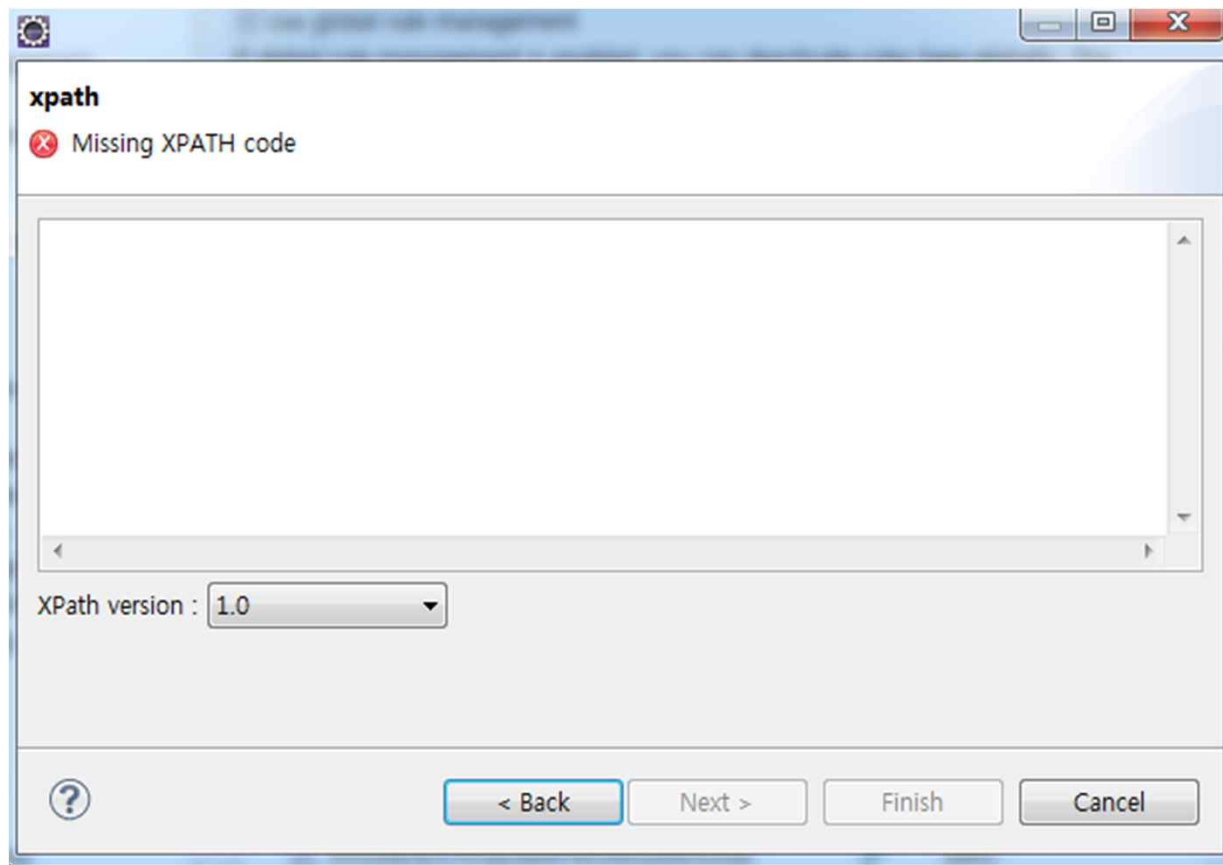
- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능





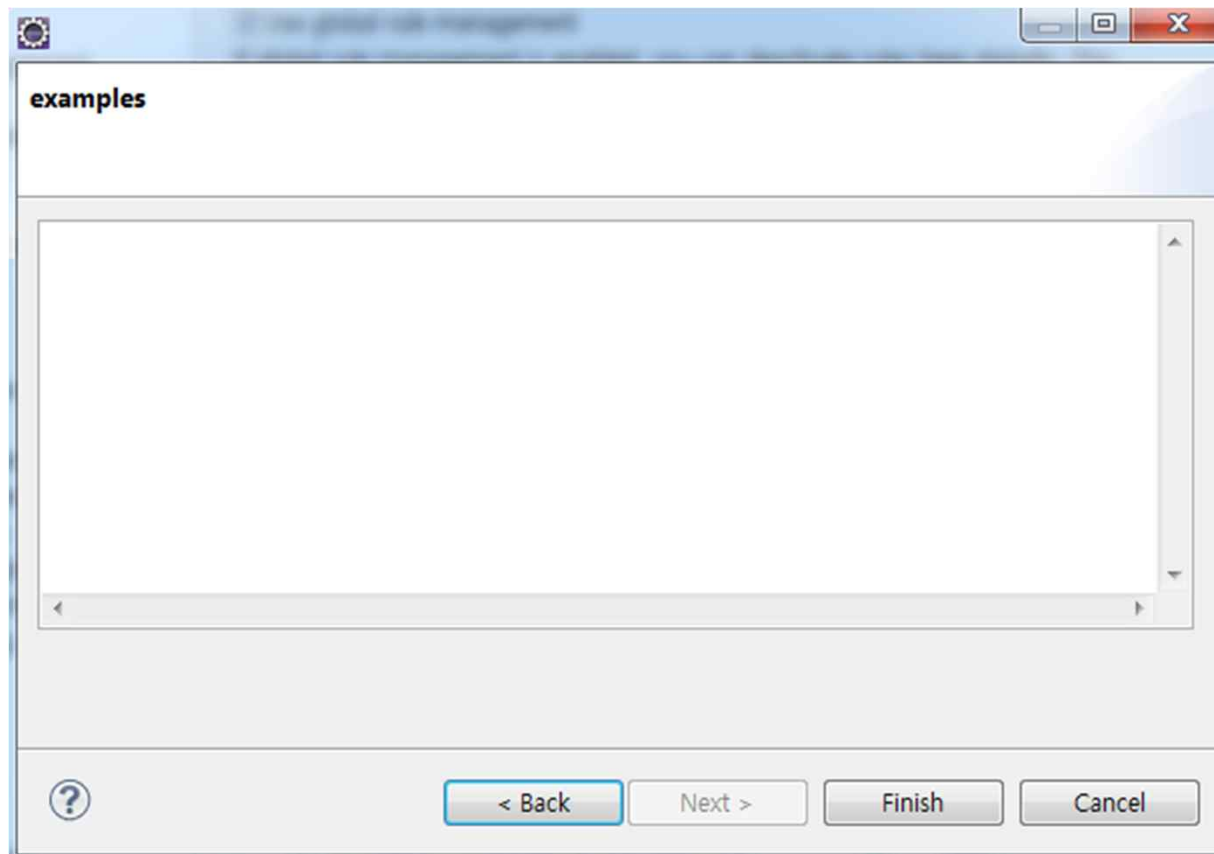
# PMD

- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능



# PMD

- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능



# PMD

- ❖ Windows -> preference-> pmd 에서 룰의 추가, 삭제 및 수정이 가능

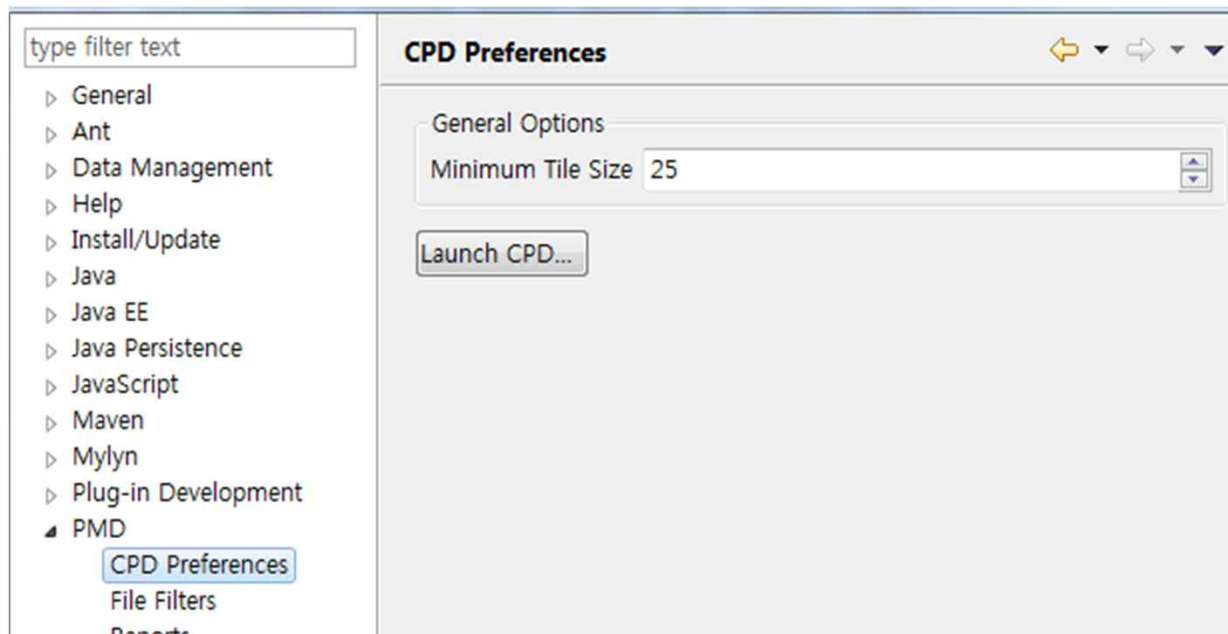
```
//CatchStatement
[count(Block/BlockStatement) = 0 and ($allowCommentedBlocks != 'true' or Block/@containsComment = 'false')]
[FormalParameter/Type/ReferenceType
 /ClassOrInterfaceType[@Image != 'InterruptedException' and @Image != 'CloneNotSupportedException']
]
```

xample:

```
public void doSomething() {
    try {
        FileInputStream fis = new FileInputStream("/tmp/bugger");
    } catch (IOException ioe) {
        // not good
    }
}
```

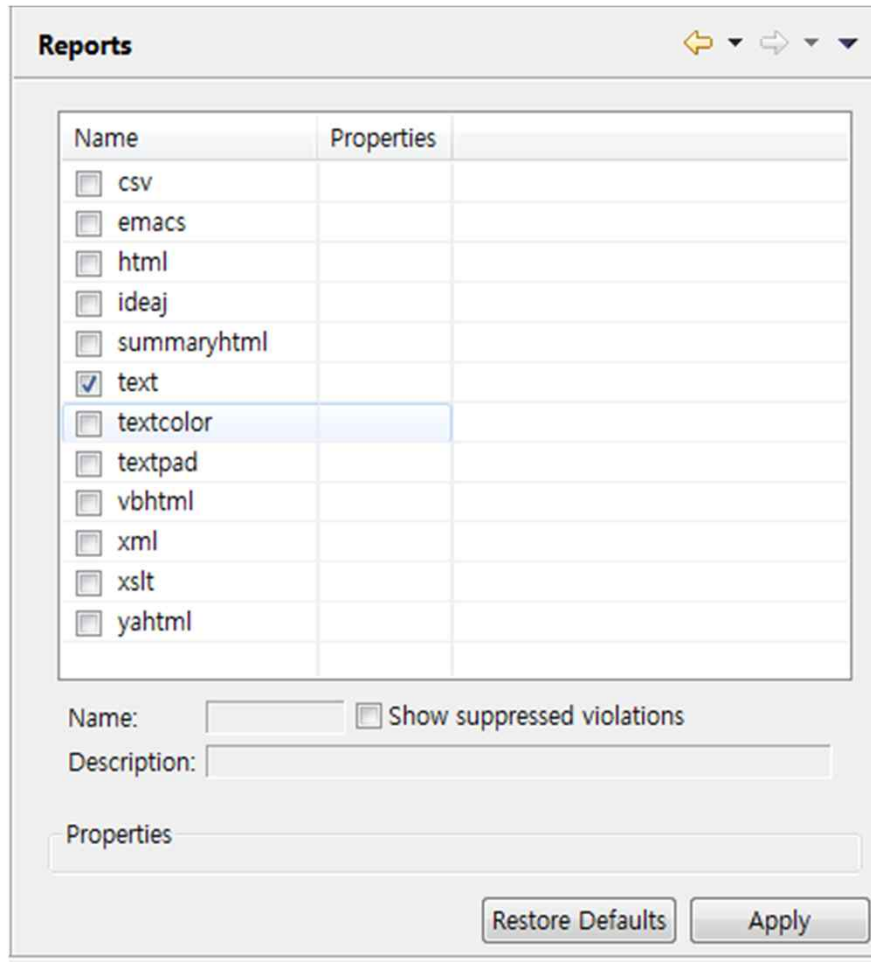
# PMD

- ❖ CPD를 통해 중복 코드를 확인 가능
- ❖ 검사하는 최소 tile size를 지정 (tile = token)



# PMD

❖ 다양한 format의 reports 기능을 제공



# PMD

## ❖ 미리 정의된 rule

- Naming Rule
  - 짧은 변수, 메소드 이름 사용
- Code Size Rule
  - Toomanyfields 같이 구조화 하지 않고 코드 사이즈만 증가시키는 룰
- Optimization Rule
  - 최적화를 위한 룰
- 등 이외에도 많은 룰셋이 존재
  
- 각 룰셋에는 수십 가지의 세부 rule들이 존재
  - unreachable code도 검사 가능

# PMD

- ❖ 적용할 규칙과 적용하지 않을 규칙 선택 해서 세팅 할 수 있다
- ❖ Project-> properties->pmd

Properties for xcount

type filter text

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

**PMD**

Enable PMD

Include derived files

Handle high priority violations as Eclipse errors

Full build enabled

No working set is selected.

Select a working set... Deselect working set

Rule source

Use the ruleset configured in a project file .ruleset

Use local rules

Language	Rule set	Rule	Priority
<input checked="" type="checkbox"/> Java	Naming	MethodNamingConventions	Blocker
<input checked="" type="checkbox"/> Java	Naming	MethodWithSameNameAs...	Urgent
<input checked="" type="checkbox"/> Java	Naming	MisleadingVariableName	Urgent
<input checked="" type="checkbox"/> Java	Naming	NoPackage	Urgent
<input checked="" type="checkbox"/> Java	Naming	PackageCase	Urgent
<input checked="" type="checkbox"/> Java	Naming	ShortClassName	Important
<input checked="" type="checkbox"/> Java	Naming	ShortMethodName	Urgent
<input checked="" type="checkbox"/> Java	Naming	ShortVariable	Urgent
<input checked="" type="checkbox"/> Java	Naming	SuspiciousConstantFieldNa...	Urgent
<input checked="" type="checkbox"/> Java	Naming	SuspiciousEqualsMethodN...	Critical
<input checked="" type="checkbox"/> Java	Naming	SuspiciousHashCodeMetho...	Urgent
<input checked="" type="checkbox"/> Java	Naming	VariableNamingConventions	Blocker

**Name**  
ShortClassName

**Description**  
Classnames with fewer than five characters are not recommended. More information can be found [here](#).

**Example**  

```
public class Foo {  
}
```

# PMD

## ❖ 실행 화면

The screenshot displays the PMD IDE interface. On the left, the Package Explorer shows the project structure with 'xcount'. The Violations Outline table lists various PMD rules and their corresponding error messages. On the right, the code editor shows the source code for 'Counter.java' and 'AutoTesterUtil.java'. The Violations Overview table at the bottom provides a summary of the detected violations.

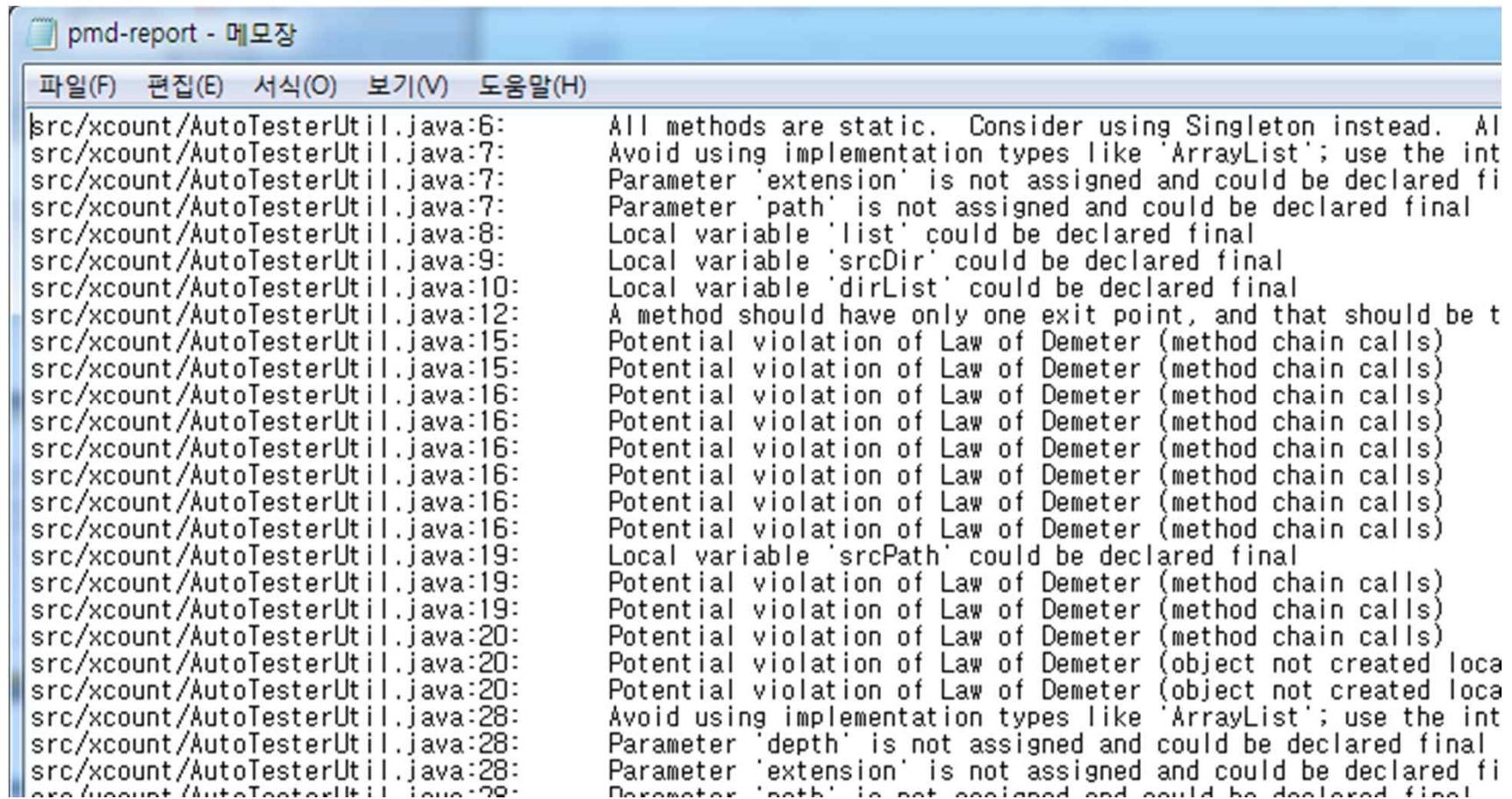
Priority	Line	created	Rule	Error Message
▶	139	Wed ...	Lo...	Local variable 'pathTok' could be declared final
▶	94	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	38	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	110	Wed ...	Lo...	Local variable 'target' could be declared final
▶	16	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	138	Wed ...	La...	Potential violation of Law of Demeter (object not created loc...
▶	38	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	101	Wed ...	Lo...	Local variable 'list' could be declared final
▶	16	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	52	Wed ...	Lo...	Local variable 'srcDir' could be declared final
▶	158	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	61	Wed ...	O...	A method should have only one exit point, and that should ...
▶	93	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	111	Wed ...	La...	Potential violation of Law of Demeter (object not created loc...
▶	10	Wed ...	Lo...	Local variable 'dirList' could be declared final
▶	158	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	16	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	94	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	59	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	136	Wed ...	La...	Potential violation of Law of Demeter (object not created loc...
▶	94	Wed ...	La...	Potential violation of Law of Demeter (method chain calls)
▶	43	Wed ...	La...	Potential violation of Law of Demeter (object not created loc...

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
PrematureDeclaration	1	8.7	0.09	xcount
UnusedLocalVariable	1	8.7	0.09	xcount
MethodArgumentCouldBeFinal	16	139.1	1.45	xcount
AvoidInstantiatingObjectsInLoops	1	8.7	0.09	xcount
LawOfDemeter	68	591.3	6.18	xcount
UseSingleton	1	8.7	0.09	xcount
UseStringBufferForStringAppends	1	8.7	0.09	xcount
DataflowAnomalyAnalysis	5	43.5	0.45	xcount
LocalVariableCouldBeFinal	28	243.5	2.55	xcount
OnlyOneReturn	7	60.9	0.64	xcount
Counter.java	18	818.2	9.00	xcount
SystemPrintln	2	90.9	1.00	xcount
DefaultPackage	1	45.5	0.50	xcount
ShortVariable	1	45.5	0.50	xcount



# PMD

## ❖ 기본적인 형태의 txt형식의 리포트



```
pmd-report - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
src/xcount/AutoTesterUtil.java:6: All methods are static. Consider using Singleton instead. Al
src/xcount/AutoTesterUtil.java:7: Avoid using implementation types like 'ArrayList'; use the int
src/xcount/AutoTesterUtil.java:7: Parameter 'extension' is not assigned and could be declared final
src/xcount/AutoTesterUtil.java:7: Parameter 'path' is not assigned and could be declared final
src/xcount/AutoTesterUtil.java:8: Local variable 'list' could be declared final
src/xcount/AutoTesterUtil.java:9: Local variable 'srcDir' could be declared final
src/xcount/AutoTesterUtil.java:10: Local variable 'dirList' could be declared final
src/xcount/AutoTesterUtil.java:12: A method should have only one exit point, and that should be t
src/xcount/AutoTesterUtil.java:15: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:15: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:16: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:16: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:16: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:16: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:16: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:16: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:19: Local variable 'srcPath' could be declared final
src/xcount/AutoTesterUtil.java:19: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:19: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:20: Potential violation of Law of Demeter (method chain calls)
src/xcount/AutoTesterUtil.java:20: Potential violation of Law of Demeter (object not created loca
src/xcount/AutoTesterUtil.java:20: Potential violation of Law of Demeter (object not created loca
src/xcount/AutoTesterUtil.java:28: Avoid using implementation types like 'ArrayList'; use the int
src/xcount/AutoTesterUtil.java:28: Parameter 'depth' is not assigned and could be declared final
src/xcount/AutoTesterUtil.java:28: Parameter 'extension' is not assigned and could be declared final
src/xcount/AutoTesterUtil.java:28: Parameter 'path' is not assigned and could be declared final
```

# PMD

## ❖ pmd

- ❖ 사용자가 규칙을 설정 가능하므로 프로젝트 진행시 팀에서 정한 네이밍 룰, 코드 스타일들을 적용하여 사용 할 수 있다
- ❖ 공용적으로 통용되는 코드 스타일들을 적용하여 코딩 스타일 형성에 도움
- ❖ Command line으로 실행과 ant script로의 실행을 지원해서 ant script 작성 후 컴파일시 자동으로 pmd가 수행되도록 만들 수 있다
- ❖ 하지만 새 규칙 생성시에 코드로 구성되어 쉽게 작성하기가 어렵다

# PMD

## ❖ Pmd ant script 예시

```
<target name="pmd">
  <taskdef name="pmd" classname="net.sourceforge.pmd.ant.PMDTask" />
  <pmd shortFileNames="true">
    <ruleset>rulesets/favorites.xml</ruleset>
    <ruleset>basic</ruleset>
    <formatter type="html" toFile="pmd_report.html" linkPrefix="http://pmd.sourceforge.net/xref/" />
    <fileset dir="/usr/local/j2sdk1.4.1_01/src/">
      <include name="java/lang/*.java" />
    </fileset>
  </pmd>
</target>
```

프로파일링을 통한 분석 도구

# ECLIPSE TPTP

# Eclipse TPTP

## ❖ Eclipse TPTP

- Eclipse Test & Performance Tool Platform Project
- 모니터링, 테스트 자동화, 프로파일링 등 문제점을 찾고 해결하는 기능을 제공
- 이클립스 플러그인으로 제공되기 때문에 설치 및 분석이 간편
- 정적 분석을 통한 지속적인 모니터링 방식

## ❖ 설치방법

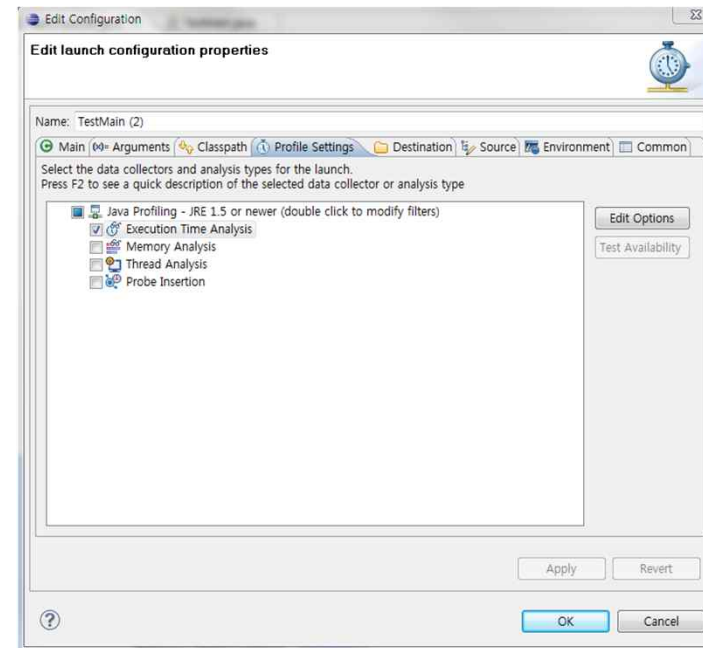
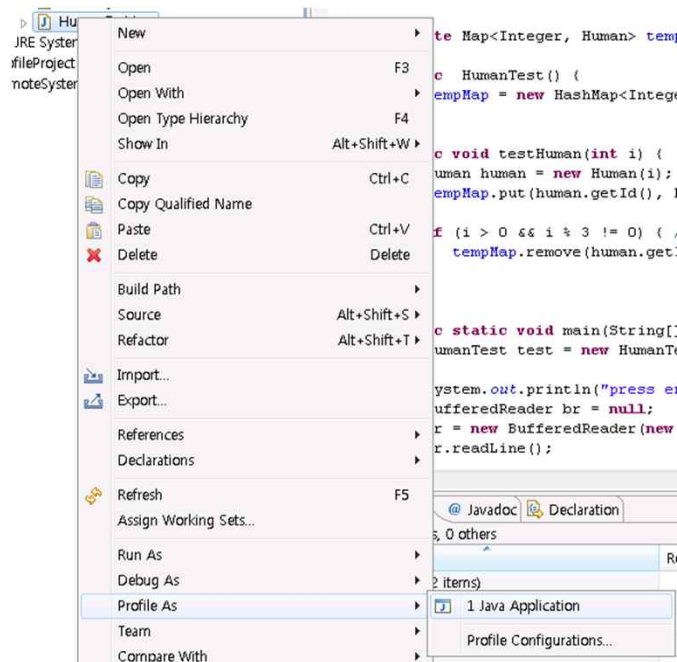
- 참고 : <http://www.eclipse.org/tptp/home/downloads/>



# Eclipse TPTP

## ❖ 실행 및 분석

- 플러그인이 정상적으로 설치 되면 이클립스에서 관련 메뉴가 생성
- 분석 대상 클래스를 Profile As로 실행하여 분석 시작



# Eclipse TPTP

## ❖ 객체 사용 현황 분석

- 현재 활성화된 객체와 총 생성된 객체, 그 크기 등의 정보를 알 수 있음
- App 동작에 따라서 지속적인 모니터링 가능
- 어떤 객체가 더 이상 사용되지 않을 시점에 활성화된 객체가 남아있을 경우 메모리 누수가 발생했음을 알 수 있음
- 메모리 누수 발생 객체는 객체 참조 그래프를 통해 추가 분석이 필요

### Memory Statistics

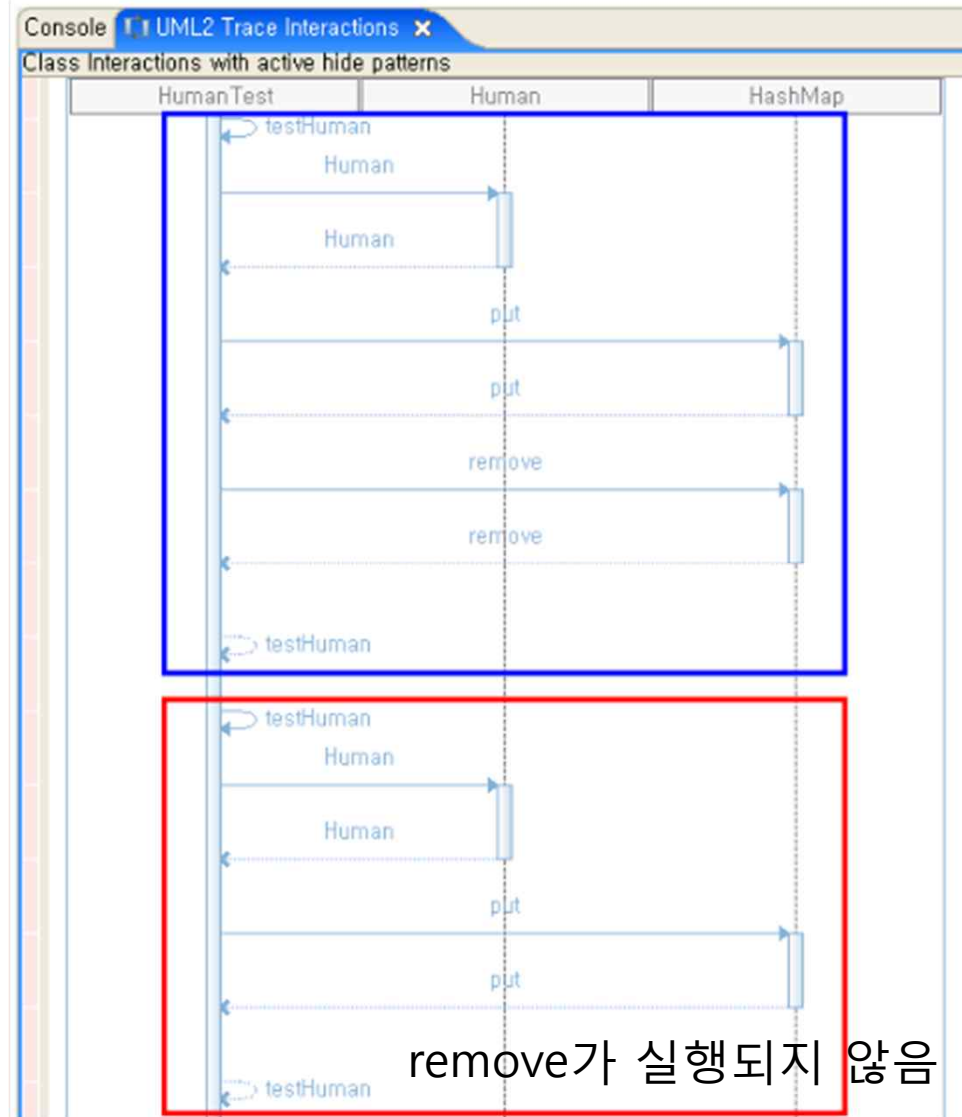
Filter: Default Filter. Click [here](#) to set filter

>Class Name	Package	Live Instances	Active Size (bytes)	Total Instances	Total Size (bytes)	Avg. Age
Human	(default package)	10,000	160,000	10,000	160,000	0
HumanTest	(default package)	1	16	1	16	0

# Eclipse TPTP

## ❖ 객체 참조 그래프

- 특정 객체가 참조되는 과정을 그래프로 표현
- 누수되는 객체를 참조하는 클래스를 발견할 수 있음
- 누락된 메소드나 논리적 오류를 발견할 수 있음





# Eclipse TPTP

## ❖ 수행 시간 관련 분석

- 어플리케이션의 수행 시간, 호출 횟수 등의 정보도 알 수 있음
- 특정 method 에서만 수행시간이 지연될 경우 문제 점이 존재한다고 생각 할 수 있음

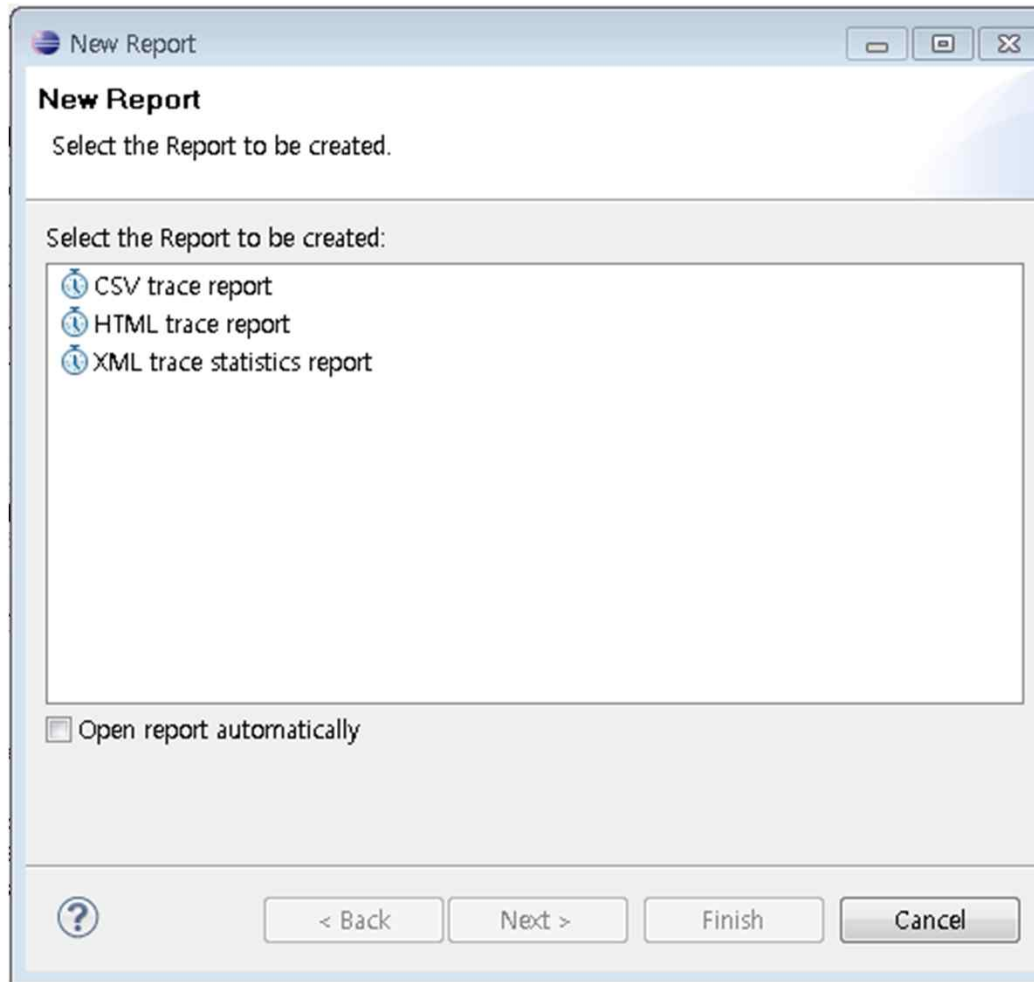
Package	<	Base Time (seconds)	Average Base Time (sec...	Cumulative Time (secon...	Calls
(default package)	◆	12.156381	0.000332	12.156381	36,670
HumanTest	◆	11.226161	0.001122	12.156381	10,003
main(java.lang.String[])	◆	10.329594	10.329594	12.156381	1
yaho() void	◆	0.681598	0.681598	0.681598	1
testHuman(int) void	◆	0.214891	0.000021	1.145111	10,000
HumanTest()	◆	0.000078	0.000078	0.000078	1
Human	◆	0.930220	0.000035	0.930220	26,667
getId() int	◆	0.497025	0.000030	0.497025	16,667
Human(int)	◆	0.433195	0.000043	0.433195	10,000

### Call Tree

Thread name	<Percent Per Thread	Cumulative Time (seco...	Min Time	Avg Time	Max Time	Calls
main[1188]	100.00%	12.288362				
main(java.lang.String[]) v	98.93%	12.156381	12.156381	12.156381	12.156381	1
testHuman(int) void	9.32%	1.145111	0.000021	0.000115	0.149168	10,000
getId() int	4.04%	0.497025	0.000003	0.000030	0.149141	16,667
Human(int)	3.53%	0.433195	0.000003	0.000043	0.113973	10,000
yaho() void	5.55%	0.681598	0.681598	0.681598	0.681598	1
HumanTest()	0.00%	0.000078	0.000078	0.000078	0.000078	1

# Eclipse TPTP

❖ 분석 수행 결과를 3가지의 형태로 리포트 도출이 가능



# Eclipse TPTP

- ❖ 정확한 문제 발생 지점을 확인하기 위해선 지속적인 모니터링이 필요
- ❖ 코드가 완전히 동작 가능한 상태에서만 분석 및 테스트가 가능

Introduce Findbugs

# FINDBUGS

# Findbugs

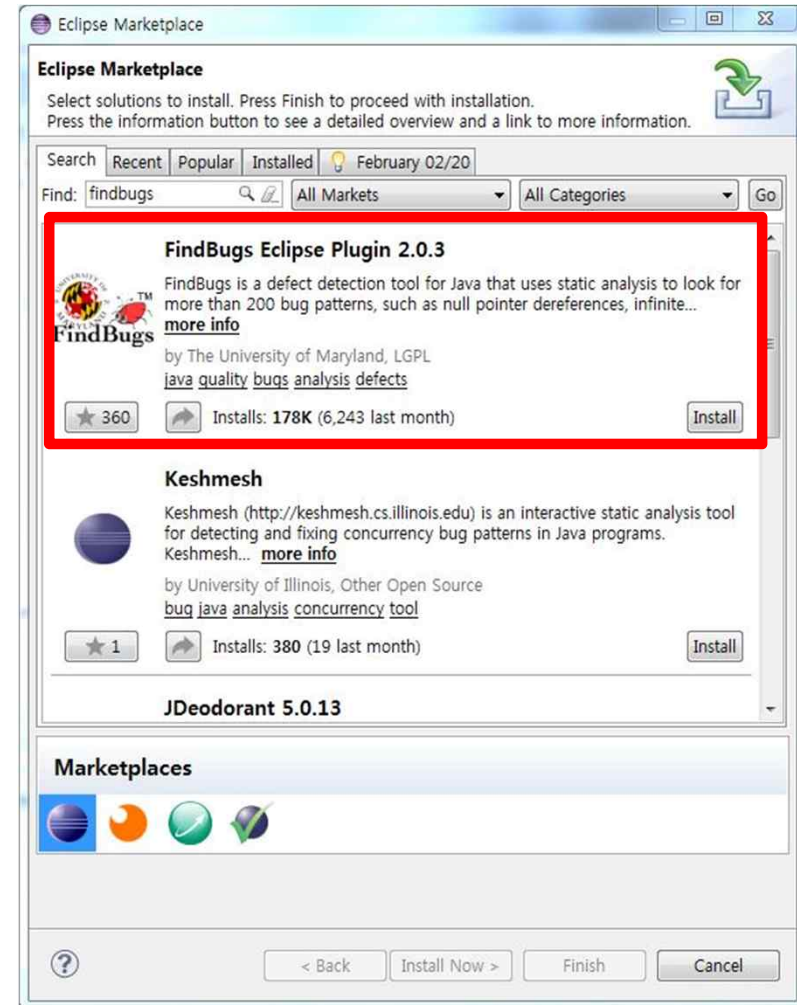
## ❖ What is the Findbugs?

- 자바 코드에 대한 정적 분석 툴
- 소스 코드에서 잠재적으로 버그가 발생할 수 있는 부분에 대한 경고를 출력한다.
- 컴파일된 코드 단계에서 실시된다.
- 프로젝트 or 클래스 단위로 체크가 가능하다.
- 코드의 건전성이 아닌 버그의 가능성에 대해서 검출한다.

# Findbugs

## ❖ How to install for Eclipse

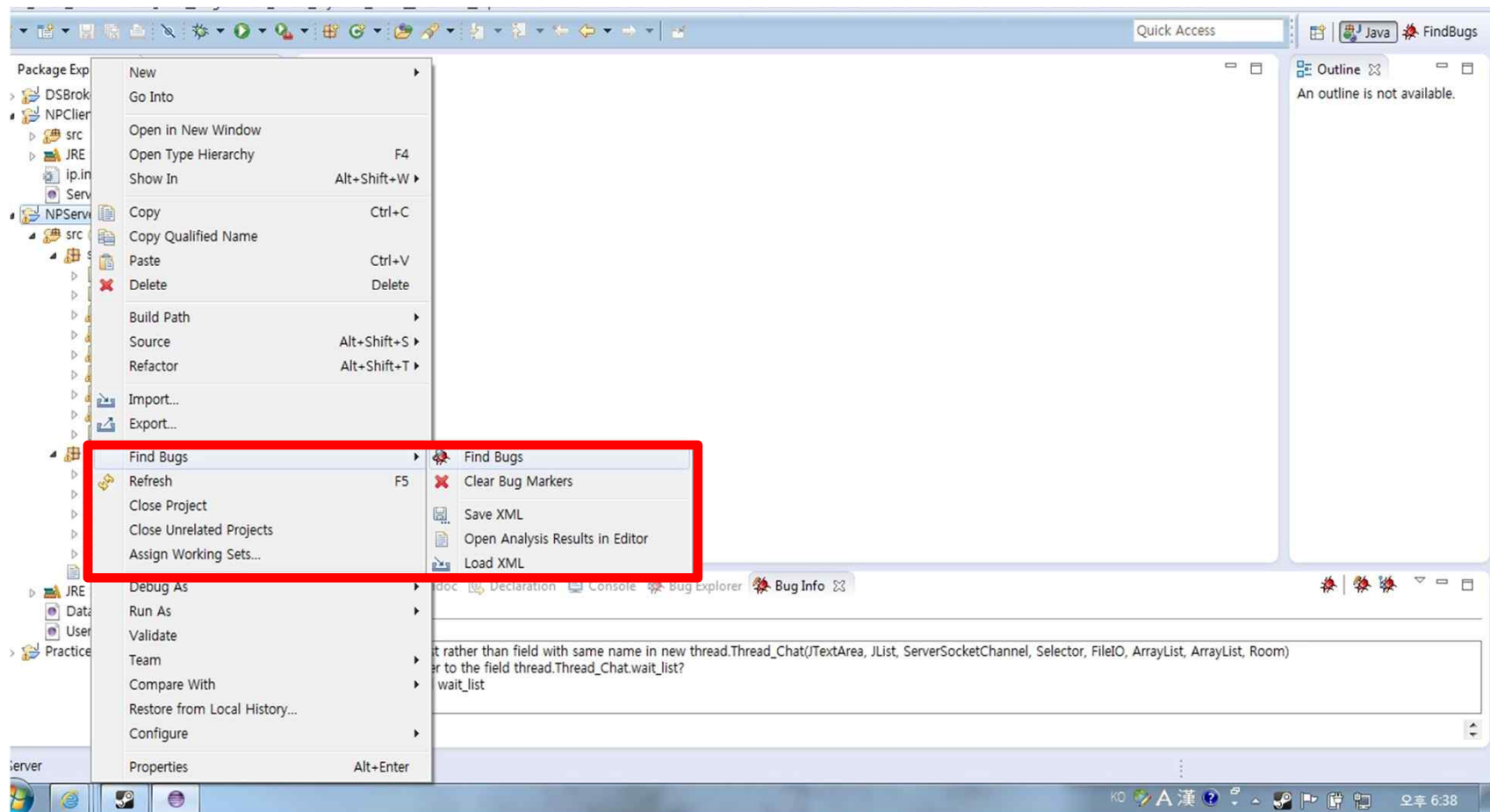
- Eclipse Marketplace에서 Findbugs를 검색하여 다운받아 설치한다.



# Findbugs

## ❖ How to use

- 검사하고자 하는 프로젝트 또는 클래스 우클릭 -> Find Bugs



# Findbugs

## ❖ Result and Debugging

- Windows -> Open Perspective -> Other -> Findbugs

발생 가능한 버그들의 목록과 원인

해당 버그의 상세 정보와 심각도 및 해결방안 제시

```
this.selector = selector;
this.io = io;
this.user_list = user_list;
this.room_list = room_list;
this.wait_list = wait_list;

charset = Charset.forName("EUC-KR");
decoder = charset.newDecoder();

io.loadUserList(user_list);

wait_list = new Room();
}

public void run()
{
    jta.append("Thread_Chat - run() start\n");
    try
```

Dead store to wait\_list rather than field with same name in new thread.Thread\_Chat(JTextArea, JList, ServerSocketChannel, Selector, FileIO, ArrayList, ArrayList, Room) [Scary(9), High confidence]

**Bug:** Dead store to wait\_list rather than field with same name in new thread.Thread\_Chat(JTextArea, JList, ServerSocketChannel, Selector, FileIO, ArrayList, ArrayList, Room)

This instruction assigns a value to a local variable, but the value is not read or used in any subsequent instruction. Often, this indicates an error, because the value



# Findbugs

## ❖ Benefit of Findbugs

- IDE와 밀접히 통합되어 있어 언제든지 쉽게 적용할 수 있다.
- 무료이다(Open Source).
- 강력하고 정확하다.
- 정적 분석 도구의 한계인 가능성만을 제시

# REFERENCES

1. JUnit, <http://using.tistory.com/54>
2. 유닛 테스트, [http://ko.wikipedia.org/wiki/유닛 테스트](http://ko.wikipedia.org/wiki/유닛_테스트)
3. Eclipse TPTP, <http://www.eclipse.org/tptp/home/>
4. Eclipse TPTP 예제, <http://javacan.tistory.com/entry/125>
5. <http://pmd.sourceforge.net/>

Q & A

**END**