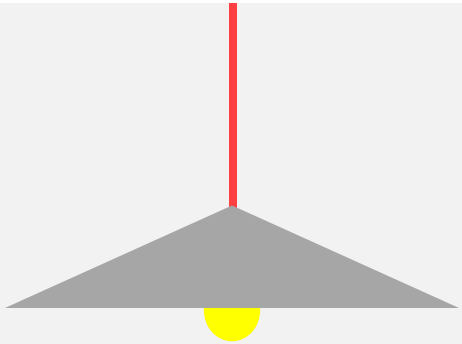
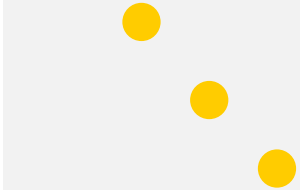
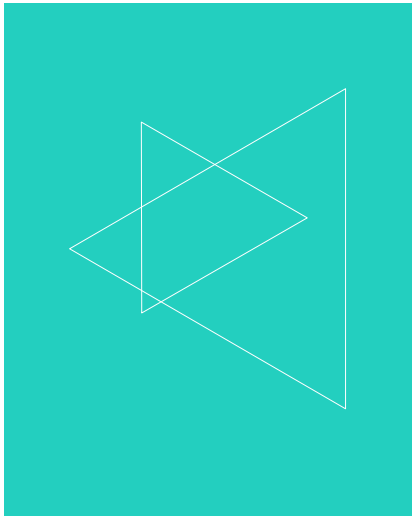


소프트웨어 검증

T3
박준모 200911391
한종철 200911429 *
신민용 201111364





목차

1

정적 분석 도구

1-0 Static Analysis

-정적 분석 이란?

1-1 Static Analysis tool

- 정적 분석 도구 사용시 이점
- 정적 분석의 단점

1-2 Using Static Analysis tool

- PMD
- Check Style
- Eclipsemetrics

정적 분석 이란?

- 소프트웨어를 개발할 시 사용하는 분석 방법으로 코드를 직접 실행해 보지 않고 테스트를 하는 방식이다.(←→ 동적 분석)

- 여러 가지 정적 분석 도구를 사용하여 코드를 실행 하지 않은 상태에서 프로그램의 오류를 찾을 수 있다.

Static Analysis tool

정적 분석 도구 사용시 이점

- 다른 테스트 자동화 보다 테스트 환경 준비 및 실행이 짧다.
- 분석 자료를 객관적인 지표로 사용 할 수 있다.
- 프로그램 결함을 예방 차원에서 접근 가능
- 시간 및 인력 투자 대비 효과(ROI)가 높다.
- 동적 분석으론 발견 할 수 없는 결함을 찾을 수 있다.

정적 분석 도구의 단점

- **허위 경보(false Alarm)이 생긴다.**
- **분석 결과는 쉽게 얻을 수 있으나 이를 사용하기 위해선 해석을 잘해야함.**
- **모든 오류 및 결함을 찾아 낼 순 없다.**

정적 분석 도구 툴 소개

역할	Tool	세부 정보
결함 예방 및 발견	PMD	자바 소스코드에 대한 잠재적인 문제에 대한 분석
	FindBugs	자바 소스코드에 대한 잠재적인 문제에 대한 분석
코딩 표준	CheckStyle	자바 프로그램에 대한 코딩 표준 준수 검사 도구
	StyleCop	C# 프로그램 언어에 대한 코딩 표준 준수 검사 도구
코드 복잡도	eclipsemetrics	소스코드 복잡도 분석 소스 코드 통계 정보 제공 도구
	javancss	자바언어에 대한 소스코드 복잡도 분석 도구, CLI 형태로 지원됨

PMD의 특징

- 오픈 소스 기반 자바소스 코드 정적 분석 도구

- 사용하지 않는 변수, 아무 처리도 안하는 catch 블록, 불필요한 객체 생성 등등을 찾아낸다.

- 소스코드에 바로 적용할 수 있는 Ruleset을 제공하며 사용자 자신의 Rule을 추가

- java, javascript, xml, xsl 등을 지원한다.

PMD

PMD가 자동으로 체크해 주는 내용

번호	특징	문제점
1	가능한 버그	비어있는 try/catch/finally/switch 문장
2	죽은 코드 (Dead code)	사용되지 않는 로컬 변수, 패러미터, private 메소드
3	최적화되지 않은 코드	낭비되는 String/StringBuffer의 사용
4	지나치게 복잡한 표현식	불필요한 if 문장, while 루프가 될 수 있는 for 루프
5	중복된 코드	복사/붙여넣기한 코드 (버그도 복사됨)

PMD Rule set

- PMD Rule set은 소스 코드 검사를 어떻게 할 지 정해주는 것이다.
- 공식적으로 지원하는 기본 Rule set 존재
- Customizing Rule set은 xml로 작성 가능하다
- 다른 사람이 쓴 Rule set을 받아서 적용 가능 하다.

1

2

PMD Rule Set

번호	Ruleset	룰의수	설명	예
1	Android	4	Android SDK 와 관련된 우수 사례(best practice)에 관한 룰	메소드 시작에서 super가 호출되어야 함
2	Basic	34	누구에게나 적용될 수 있는 모범사례(good practice)에 관한 룰	catch block이 비어 있는 경우
3	Brace	4	괄호와 관련된 룰	if 문장은 괄호를 사용해야 함
4	Code size	11	Code size와 관련된 문제를 찾는 룰	메소드의 길이가 너무 긴 경우
5	Clone	3	done() 메소드를 의심스럽게 사용하는 경우를 찾는 룰	불필요한 생성자
6	Controversial	19	논란의 소지가 있는 룰	인자도 바디도 없는 생성자
7	Coupling	3	객체와 패키지 사이에 강한 또는 적절하지 않은 결합(coupling)이 있는 인스턴스를 찾는 룰	지나치게 많은 imports
8	Design	48	의심스러운 설계를 찾는 룰	Singleton 패턴의 고려(static method들만 있는 클래스)
9	Finalizer	6	Finalizer에서 발생할 수 있는 문제를 다루는 룰	Finalize() 메소드가 비어있는 경우
10	Import Statements	5	클래스의 import문장과 관련된 문제를 다루는 룰	중복된 import 문장
11	J2EE	9	J2EE에 관한 룰	적절한 클래스 로더 사용하기

PMD Rule Set

번호	Ruleset	룰의 수	설명	예
12	Javabeans	2	Bean 룰을 따르지 않는 인스턴스를 찾는 룰	클래스가 bean이거나 직/간접적으로 참조되면 직렬화되어야 한다.
13	Junit Tests	10	Junit 테스트를 수행하다 발생할 수 있는 문제를 찾는 룰	Junit테스트에서 suit() 메소드는 public이고 static이어야 함
14	Logging(Java)	4	Logger의 의심스러운 사용을 찾는 룰	각 클래스에서 사용되는 logger는 1개임
15	Logging(Jakarta)	2	Jakarta Common Logging 프레임워크의 의심스러운 사용을 찾는 룰	Stackstrace의 모든 내용을 출력하기 위해서는 logging문자에서 2개의 인자를 사용해야 함
16	Migrating	14	하나의 JDK 버전에서 다른 버전으로 업그레이드 할 때 관련된 룰	Vector 대신 ArrayList의 사용
17	Naming	18	이름이 너무 길거나 짧은지 등 이름에 관련된 룰	필드, 로컬 별수, 패러미터 이름이 너무 짧은 경우를 찾음
18	Optimizations	10	수행성능을 향상 시키는 우수사례를 적용하기 위한 룰	한번만 할당되는 로컬 변수는 final로 선언함
19	Strict Exceptions	10	예외를 발생시키고 잡기 위한 가이드라인에 관련된 룰	Throwable을 catch하는 것을 피함 (너무 다양한 예외)
20	Strings	15	String과 StringBuffer를 사용할 때 발생할 수 있는 문제점을 찾는 룰	동일한 문자열은 상수로 선언하는 것이 바람직함
21	Sun Security	2	Sun의 보안 가이드라인을 체크하기 위한 룰	메소드가 내부 배열을 반환하면 보안상 문제가 발생할 수 있음
22	Unused Code	4	사용되지 않는 코드를 찾는 룰	로컬변수가 선언되었으나 사용되지 않음

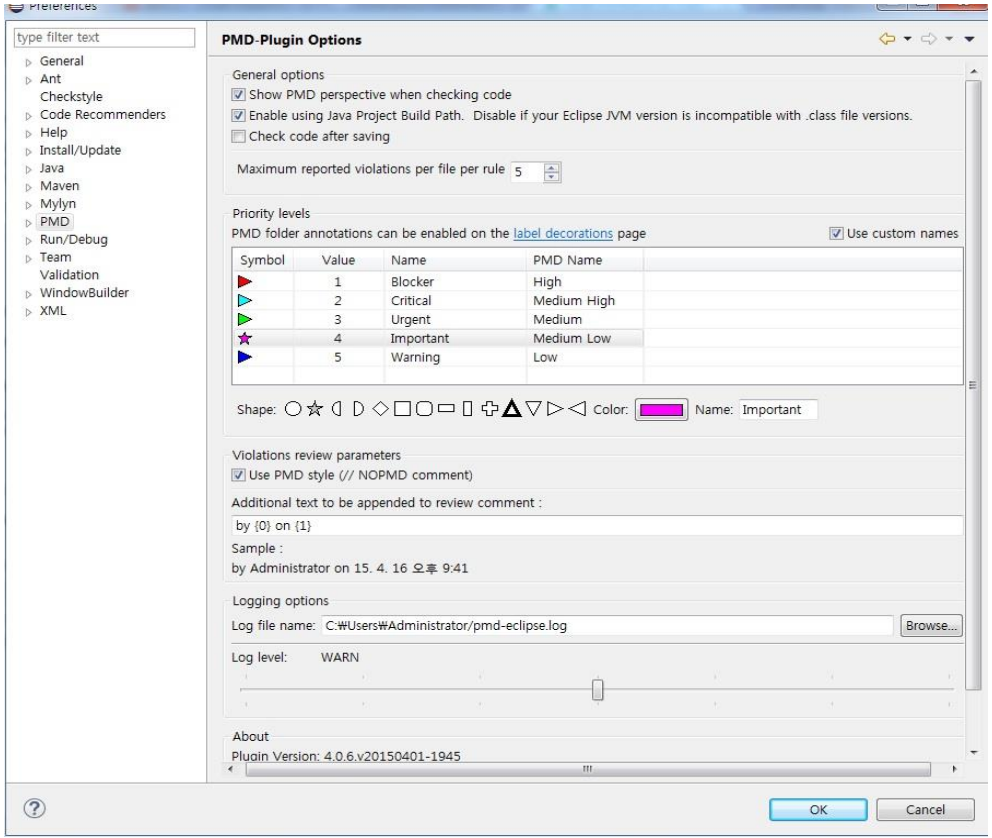
PMD 사용법

- 정해진 규칙(룰) 기반의 정적 분석 도구
- 정해진 규칙을 위반 여부를 체크
- 룰의 priorty설정 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 수정 이 가능

1

2

PMD 예제

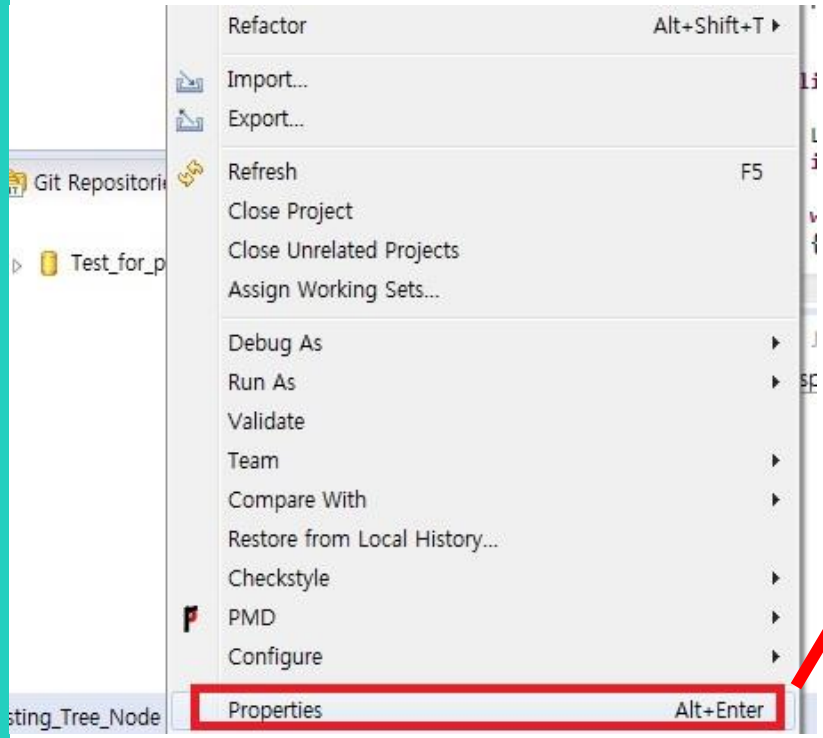


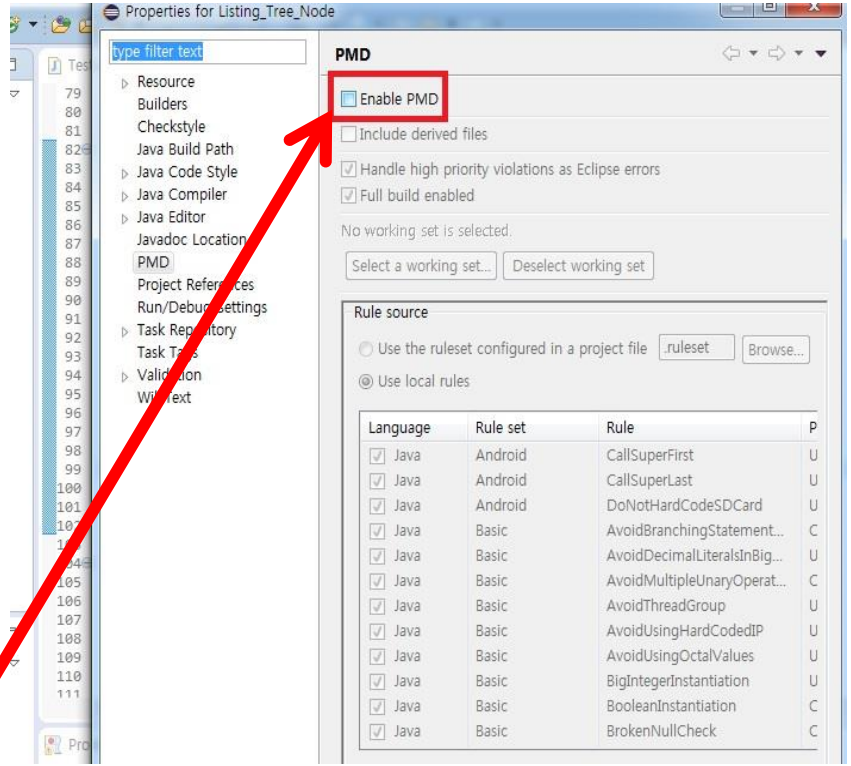
-Window → Preference →PMD

1

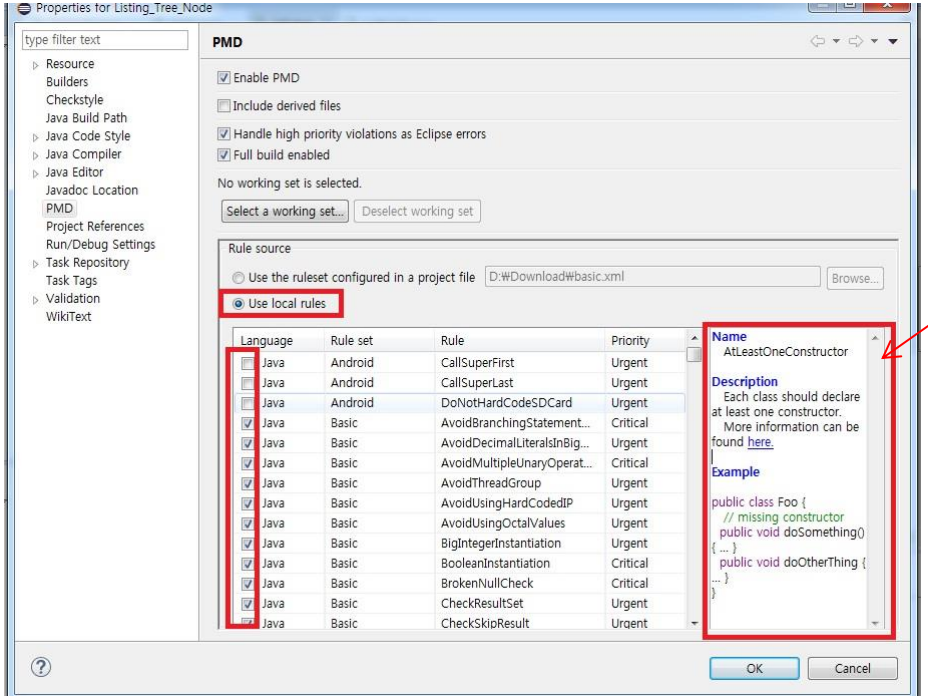
2

PMD 예제


Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Debug As
Run As
Validate
Team
Compare With
Restore from Local History...
Checkstyle
PMD
Configure
Properties Alt+Enter
Git Repositori
Test_for_p
sting_Tree_Node


Properties for Listing_Tree_Node
type filter text
Resource
Builders
Checkstyle
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
PMD
Project References
Run/Debug Settings
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
Browse...
Use local rules
Language
Rule set
Rule
Java Android CallSuperFirst
Java Android CallSuperLast
Java Android DoNotHardCodeSDCard
Java Basic AvoidBranchingStatement...
Java Basic AvoidDecimalLiteralsInBig...
Java Basic AvoidMultipleUnaryOperat...
Java Basic AvoidThreadGroup
Java Basic AvoidUsingHardCodedIP
Java Basic AvoidUsingOctalValues
Java Basic BigIntegerInstantiation
Java Basic BooleanInstantiation
Java Basic BrokenNullCheck

PMD 예제

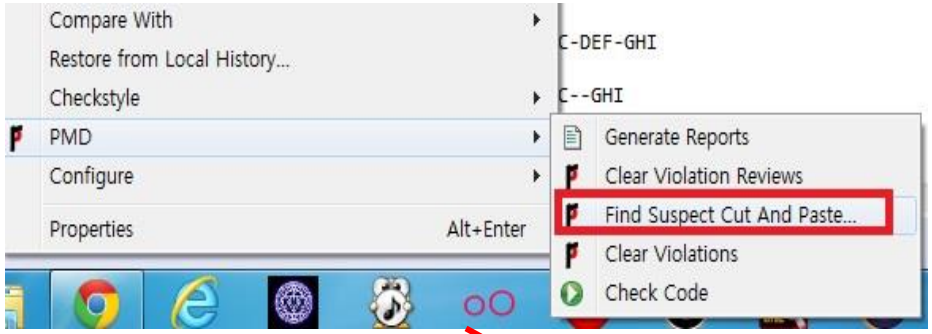


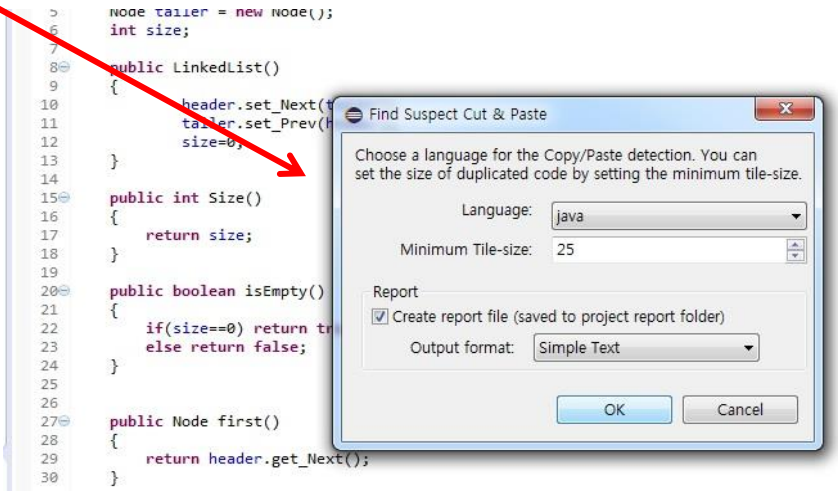
Rule Set 설명부분

1

2

PMD 예제

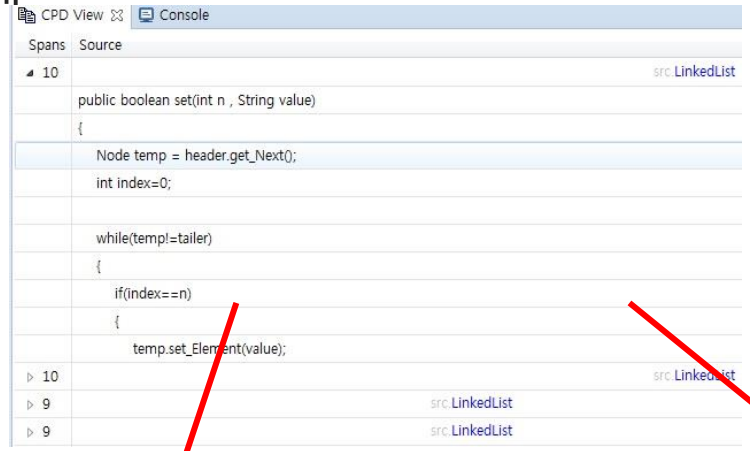

Compare With
Restore from Local History...
Checkstyle
PMD
Configure
Properties
Alt+Enter
Generate Reports
Clear Violation Reviews
Find Suspect Cut And Paste...
Clear Violations
Check Code


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
OK
Cancel

1

2

PMD 예제



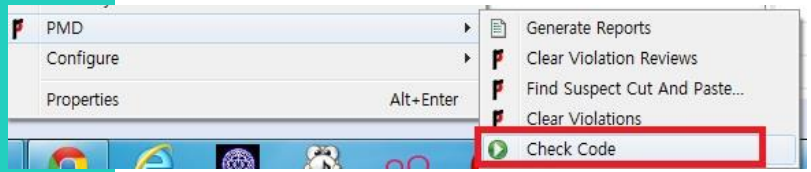

```
public boolean addAfter(int n , String value)
{
    Node temp = header.get_Next();
    int index=0;

    while(temp!=tailer)
    {
        if(index==n)
        {
            Node node = new Node(temp,temp.get_Next(),value);
            temp.get_Next().set_Prev(node);
            temp.set_Next(node);
            return true;
        }

        temp=temp.get_Next();
        index ++;
    }
```

```
        return false;
    }

    public boolean remove(int n)
    {
        Node temp = header.get_Next();
        int index=0;

        while(temp!=tailer)
        {
            if(index==n)
            {
                temp.get_Prev().set_Next(temp.get_Next());
                temp.get_Next().set_Prev(temp.get_Prev());
                temp.set_Next(null);
                temp.set_Prev(null);
                return true;
            }

            temp=temp.get_Next();
            index ++;
        }
```

1

2

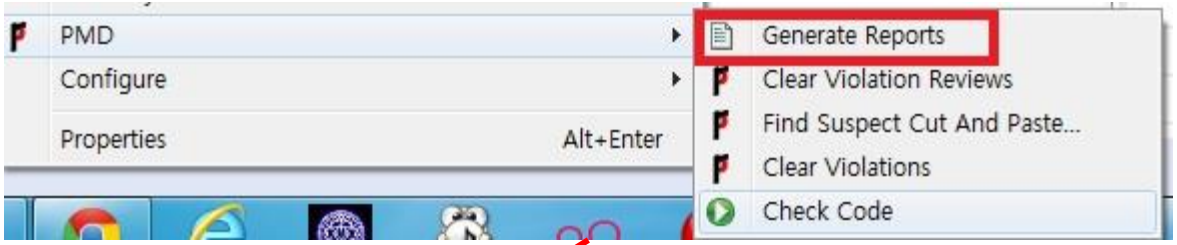
PMD 예제

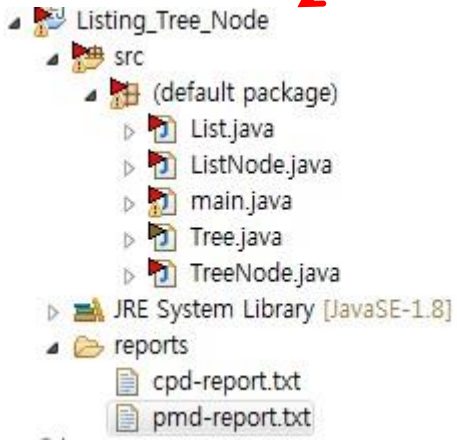


Violations Outline

P	Line	created	Rule	Error Message
▶	4	Thu Apr 16 22:36:3...	ClassNamingConventions	Class names should begin with an uppercas...
▶	9	Thu Apr 16 22:36:3...	VariableNamingConventions	Variables should start with a lowercase char...
▶	4	Thu Apr 16 22:36:3...	UseUtilityClass	All methods are static. Consider using a uti...
▶	15	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	12	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	6	Thu Apr 16 22:36:3...	MethodWithSameNameAsEncl...	Classes should not have non-constructor m...
▶	10	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	9	Thu Apr 16 22:36:3...	ShortVariable	Avoid variables with short names like T
▶	6	Thu Apr 16 22:36:3...	CommentRequired	publicMethodCommentRequirement Requir...
▶	12	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	13	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	11	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	4	Thu Apr 16 22:36:3...	CommentRequired	headerCommentRequirement Required
▶	6	Thu Apr 16 22:36:3...	MethodArgumentCouldBeFinal	Parameter 'args' is not assigned and could ...
▶	15	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	9	Thu Apr 16 22:36:3...	LocalVariableCouldBeFinal	Local variable 'T' could be declared final
▶	13	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	4	Thu Apr 16 22:36:3...	NoPackage	All classes and interfaces must belong to a ...
▶	14	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
▶	14	Thu Apr 16 22:36:3...	LawOfDemeter	Potential violation of Law of Demeter (met...
★	4	Thu Apr 16 22:36:3...	ShortClassName	Avoid short class names like main
★	6	Thu Apr 16 22:36:3...	UseVarargs	Consider using varargs for methods or cons...

PMD 예제





Test Report 텍스트 파일 생성

Checkstyle 이란?

- 코딩 하면서 소스 코드 내에서 다양한 위반 사항에 대해 알 수 있고,
  개발자들이 체크인 전에 위반 사항을 고칠 수 있다.
  또한 정해놓은 코딩 규칙에 따라 팀원들이 보다 쉽게 규칙을 적용할 수
  있게 도와주는 도구

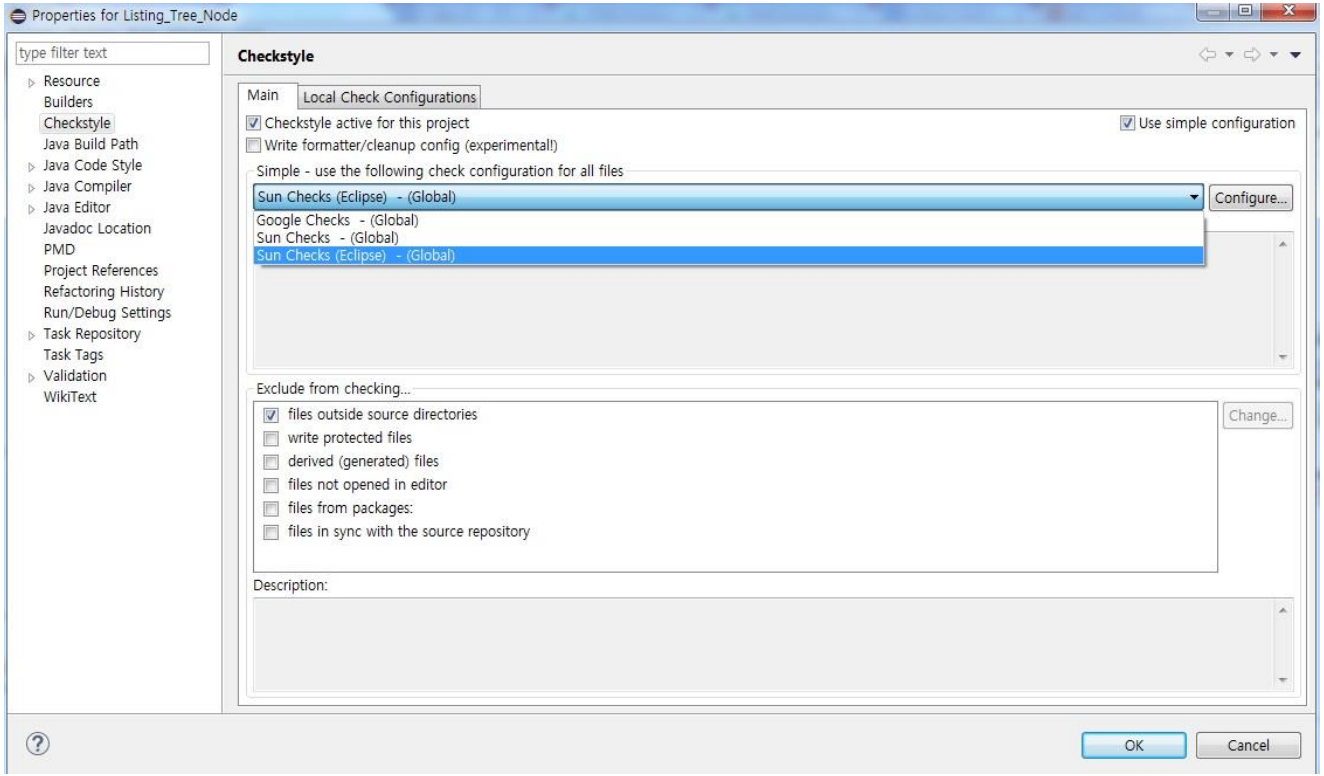
Checkstyle 의 특징

- 여러 사람과 작업 시 손쉽게 코딩 스타일을 맞출 수 있음.
- 표준 코딩 스타일을 손쉽게 프로젝트에 적용.
- 개발 초기에 소스 코드의 잠재적 결함을 발견

1

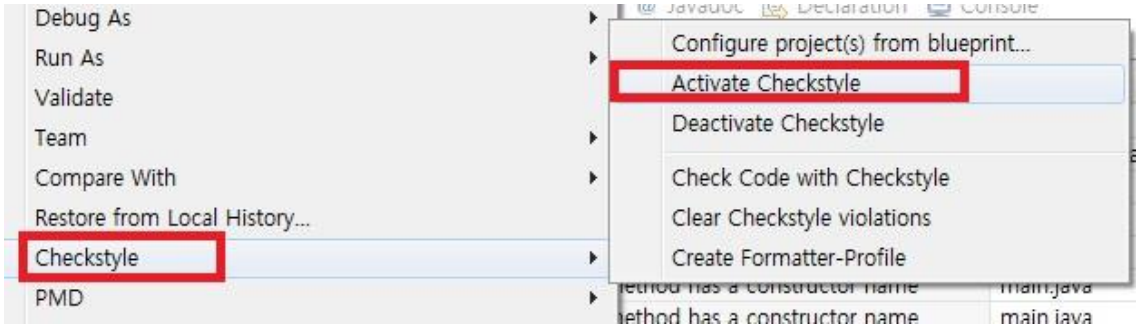
2

Check Style 사용 예제



해당 프로젝트에서 마우스 우 클릭 → Properties → CheckStyle

Check Style 예제



이름	설명
Configure projects(s) from blueprint...	Blueprint로 부터 선택된 프로젝트에 Checkstyle을 적용
Active Checkstyle	Checkstyle 적용
Deactive Checkstyle	Checkstyle 해지
Check Code with Checkstyle	Checkstyle로 code 검사
Clear Checkstyle violations	Checkstyle 에 의해 발견된 에러 제거
Create Formatter-Profile	Formatter-Profile 생성

Check Style 예제

```
25
26
27    public ListNode first()
28    {
29        return header.get_Next();
30    }
31
32    public ListNode last()
33    {
```

Problems | Javadoc | Declaration | Console

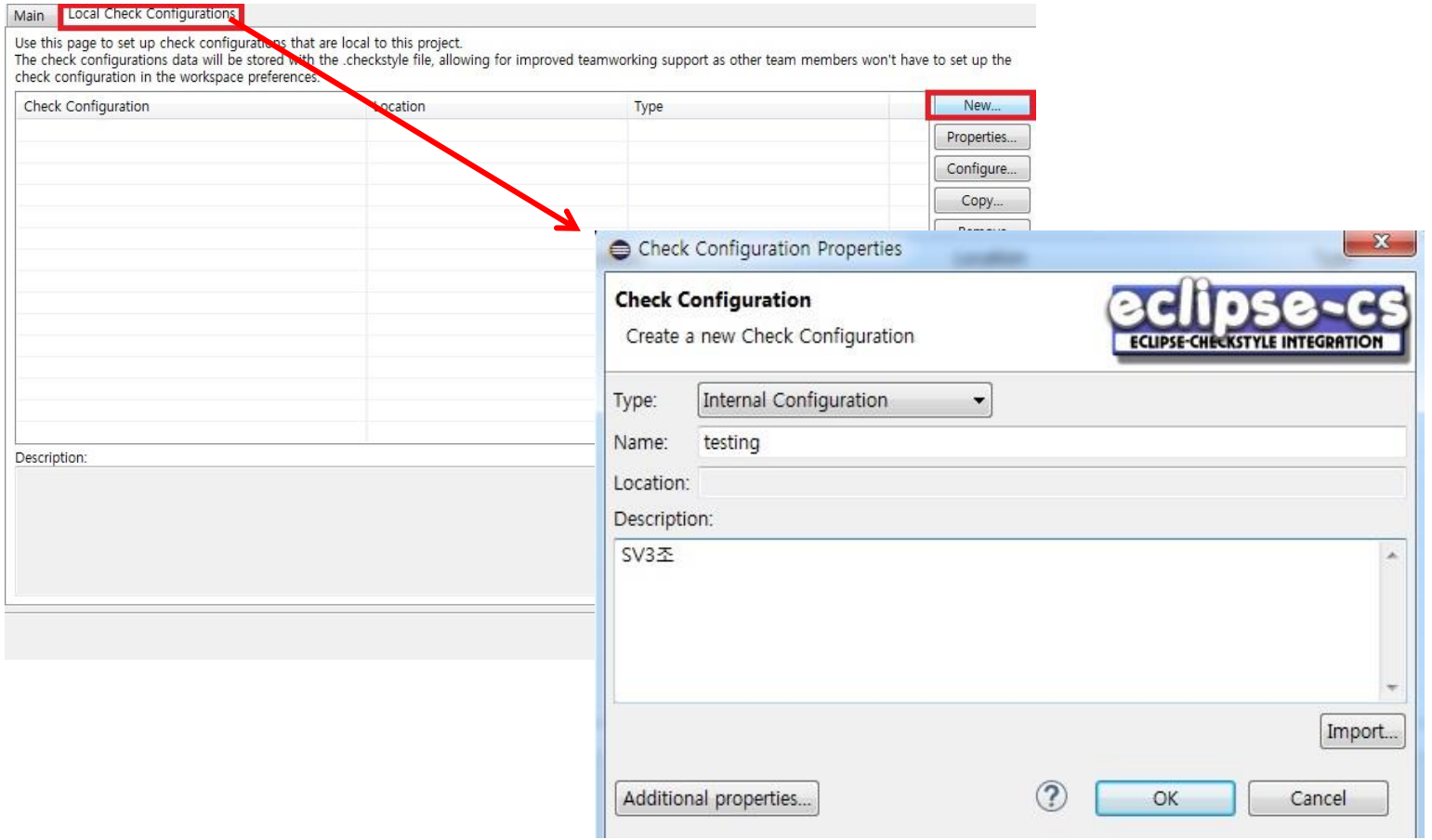
0 errors, 354 warnings, 0 others (Filter matched 100 of 354 items)

Description	Resource	Path	Location	Type
Warnings (100 of 354 items)				
',' is not followed by whitespace.	List.java	/Listing_Tree_Node...	line 49	Checkstyle Pr...
',' is not followed by whitespace.	List.java	/Listing_Tree_Node...	line 49	Checkstyle Pr...
',' is not followed by whitespace.	List.java	/Listing_Tree_Node...	line 57	Checkstyle Pr...
',' is not followed by whitespace.	List.java	/Listing_Tree_Node...	line 57	Checkstyle Pr...
',' is not followed by whitespace.	List.java	/Listing_Tree_Node...	line 91	Checkstyle Pr...
',' is not followed by whitespace.	List.java	/Listing_Tree_Node...	line 91	Checkstyle Pr...

1

2

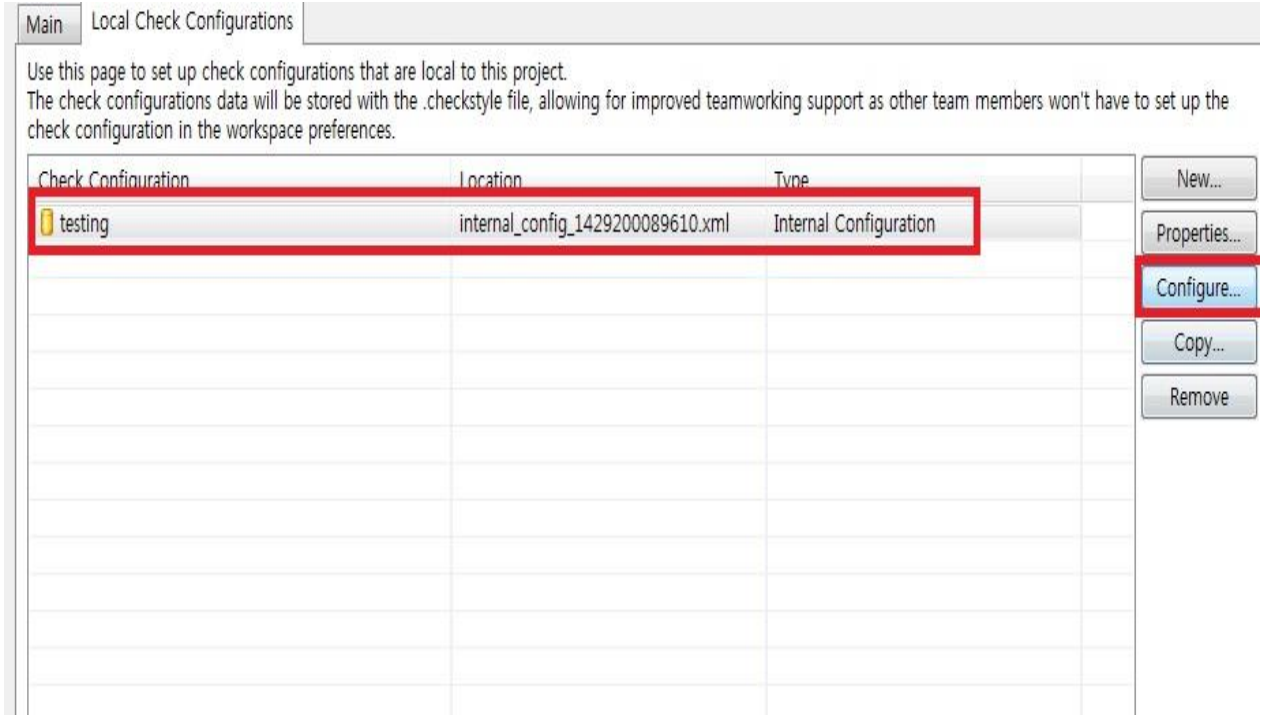
Check Style 예제



1

2

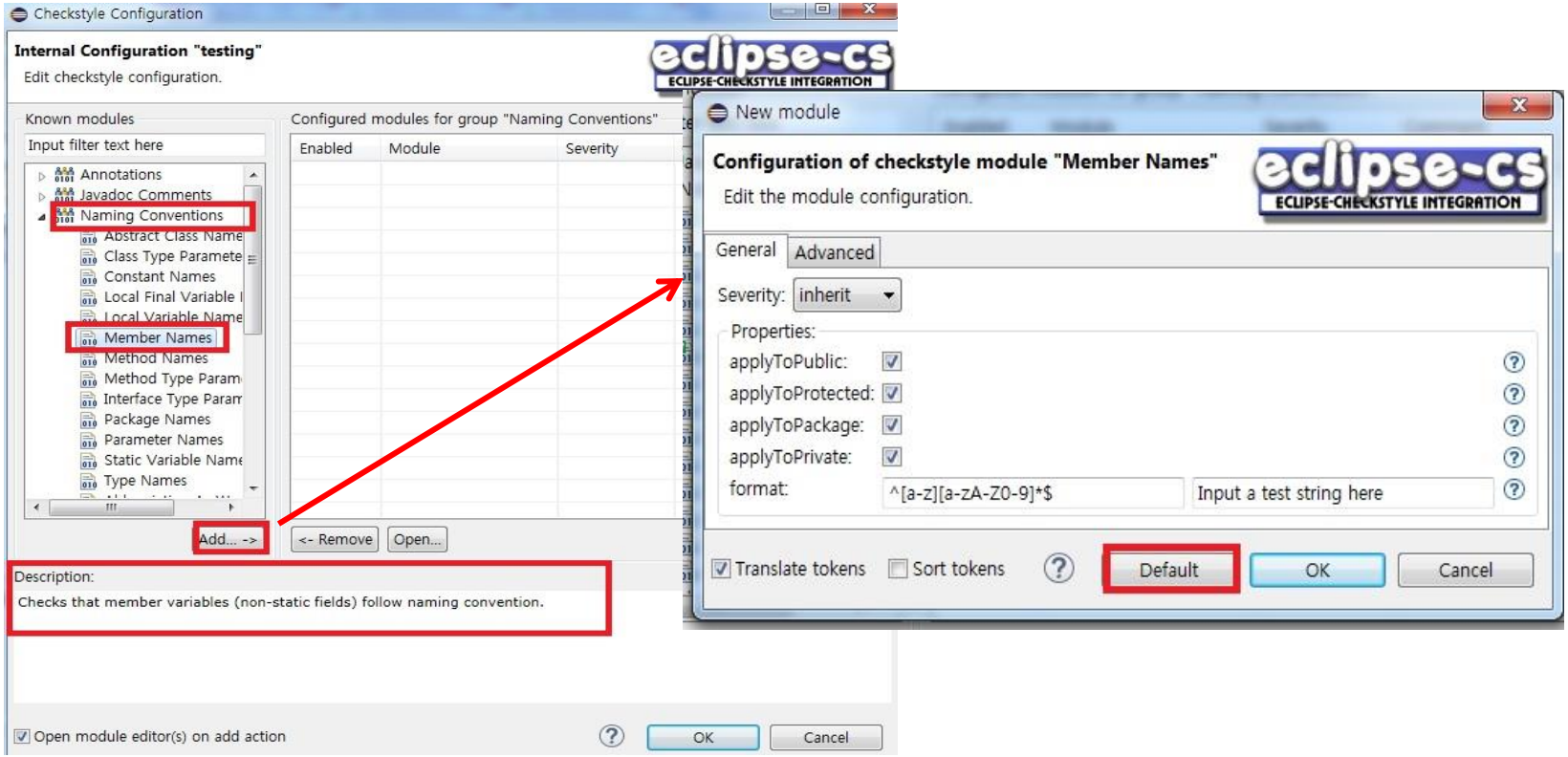
Check Style 예제



1

2

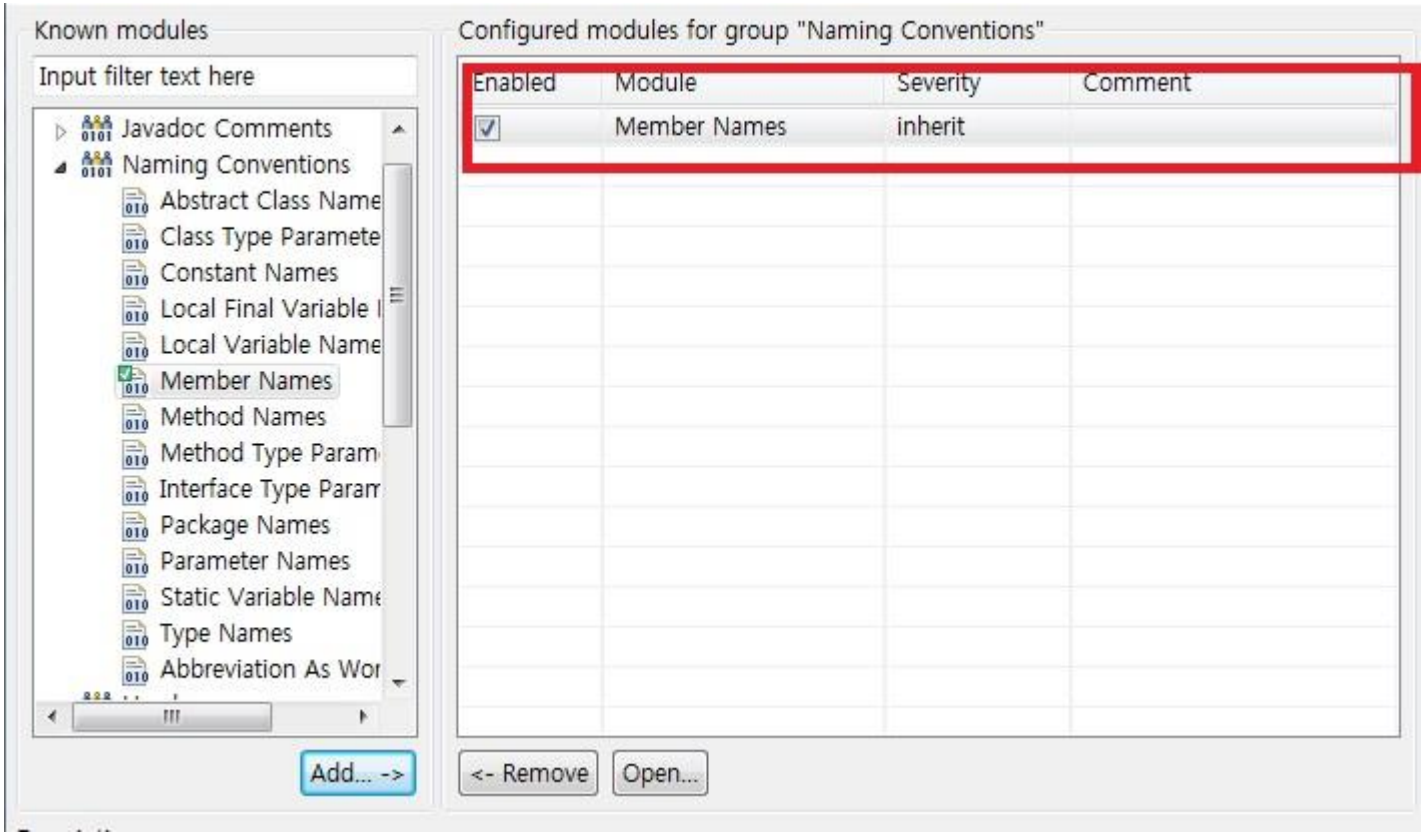
Check Style 예제



1

2

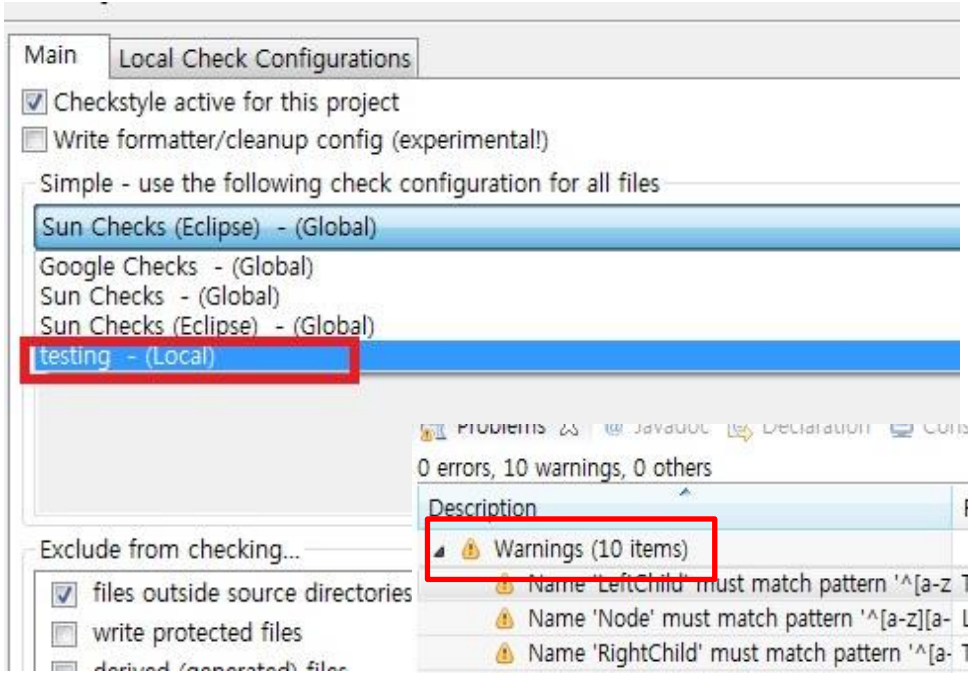
Check Style 예제



1

2

Check Style 예제



0 errors, 10 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (10 items)				
Name 'LeftChild' must match pattern '^[a-z	TreeNode.java	/Listing_Tree_Node...	line 6	Checkstyle Pr...
Name 'Node' must match pattern '^[a-z][a-	ListNode.java	/Listing_Tree_Node...	line 3	Checkstyle Pr...
Name 'RightChild' must match pattern '^[a-	TreeNode.java	/Listing_Tree_Node...	line 7	Checkstyle Pr...
The import java.util is never used	LinkedList.java	/Linked_list/src	line 1	Java Problem
The import org.junit.Ignore is never used	TestCase.java	/Junit_Test/src/pk1	line 9	Java Problem
The value of the local variable a is not used	main.java	/pmd_for_hudson/...	line 7	Java Problem
The value of the local variable b is not used	main.java	/pmd_for_hudson/...	line 7	Java Problem

Eclipse metrics 란?

자바 프로젝트의 코드에 대한 정보(라인 수, 인자 수, 구문 수 등의 통계) 및

각 클래스의 결합도, 응집도, 복잡도 등을 쉽게 파악할 수 있도록 도와주는 도구.

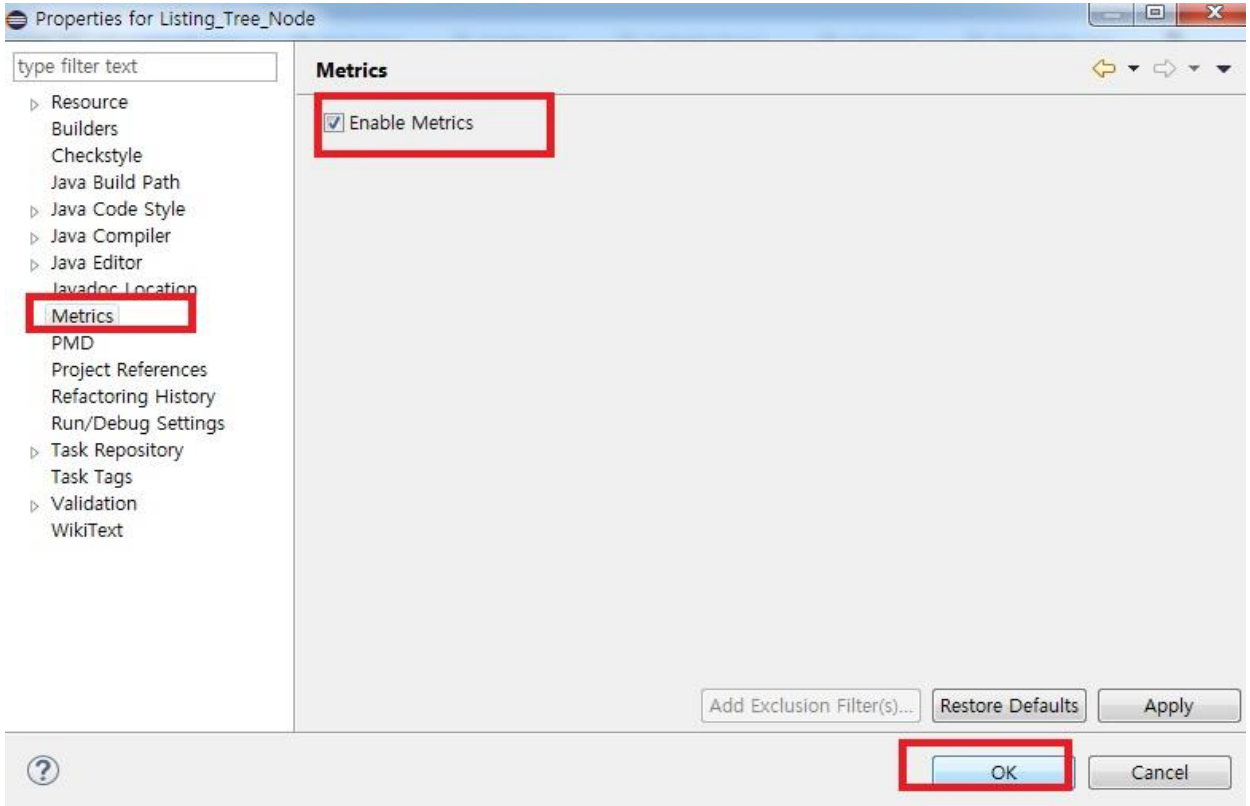
Eclipse metrics 의 특징

- 자바 프로젝트에 대한 각종 Metrics를 편리하게 산출해준다.
- Metrics 결과를 시각적으로 출력.
- Metrics 결과를 HTML, CSV 등의 다양한 방식으로 출력할 수 있다.
- 출력된 Metrics는 객체 지향적인 코드인지 파악하는데 참조 가능
- Quality Assurance에 효과적입니다.

1

2

Metrics 예제

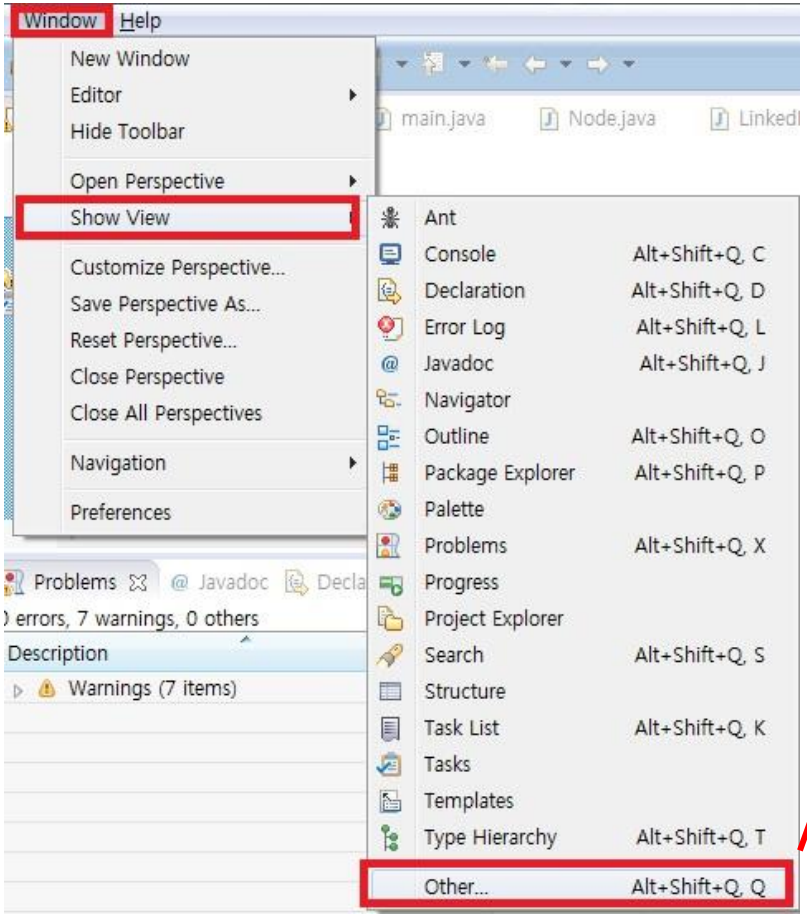


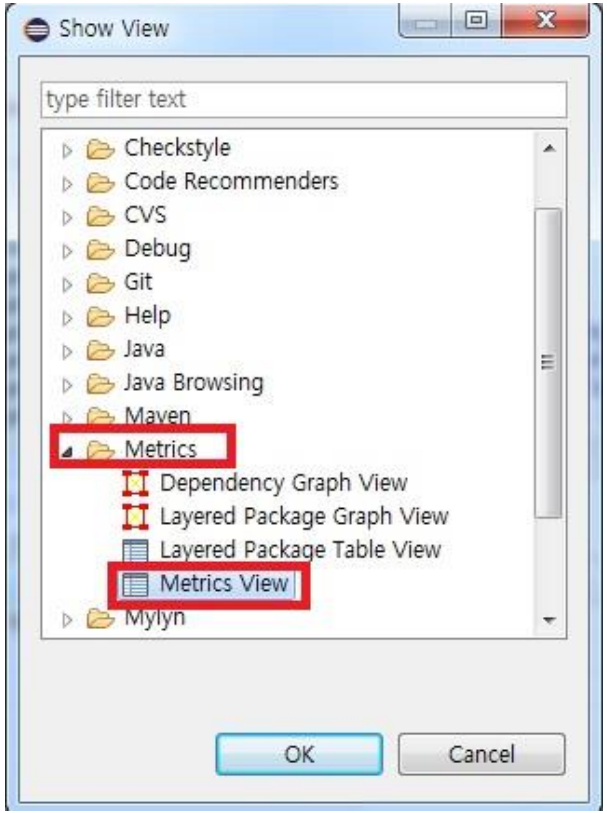
해당 프로젝트에서 마우스 우 클릭 → Properties → Metrics

1

2

Metrics 예제





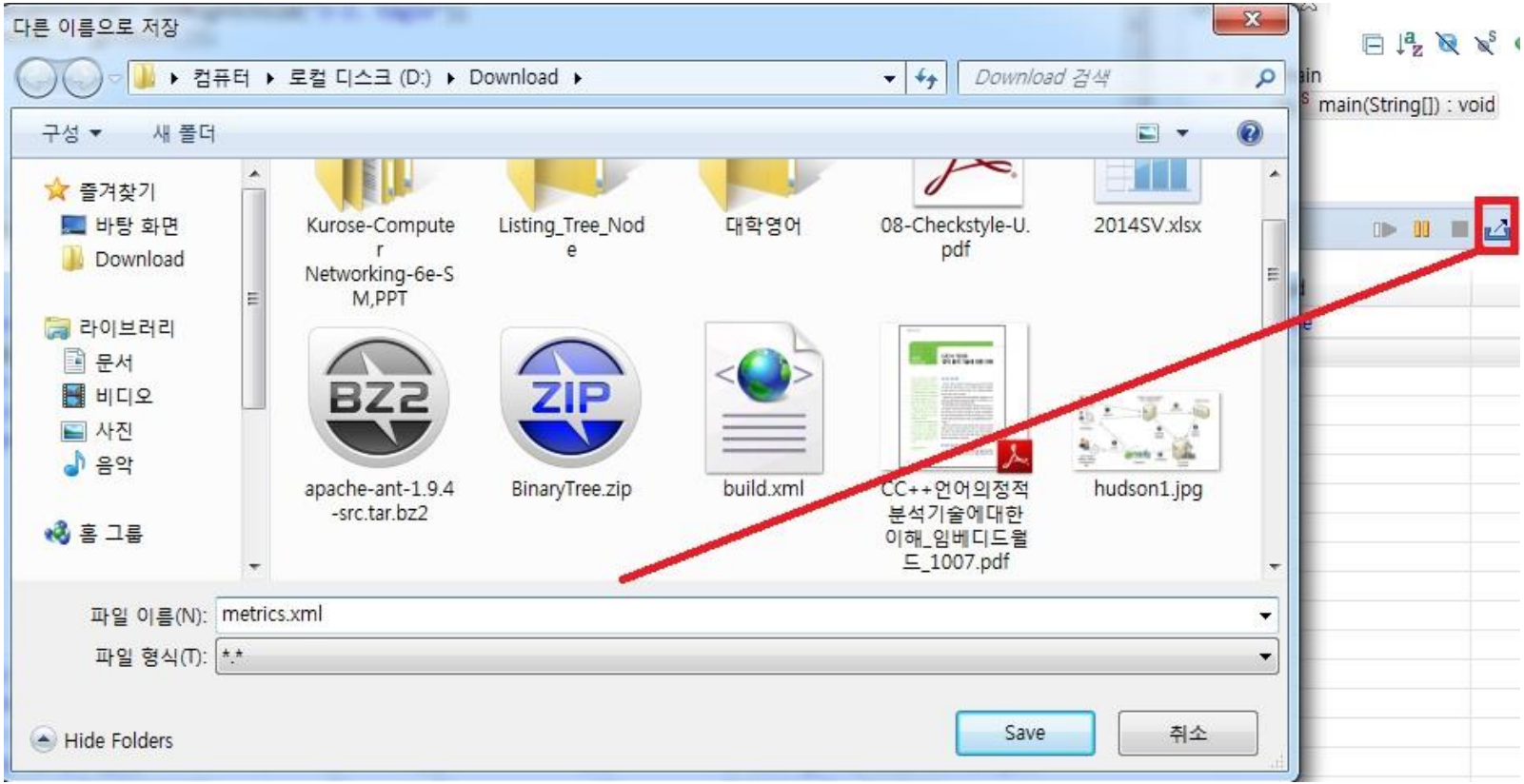
Metrics 예제

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
Number of Parameters (avg/max per method)		0.643	0.75	3	/Listing_Tree_Node/src/ListNode.java	ListNode
Number of Static Attributes (avg/max per type)	0	0	0	0	/Listing_Tree_Node/src/main.java	
Efferent Coupling (avg/max per packageFragment)		0	0	0	/Listing_Tree_Node/src	
Specialization Index (avg/max per type)		0	0	0	/Listing_Tree_Node/src/main.java	
Number of Classes (avg/max per packageFragment)	5	5	0	5	/Listing_Tree_Node/src	
src	5	5	0	5	/Listing_Tree_Node/src	
(default package)	5					
main.java	1					
List.java	1					
Tree.java	1					
TreeNode.java	1					
ListNode.java	1					
Number of Attributes (avg/max per type)	12	2.4	1.356	4	/Listing_Tree_Node/src/TreeNode.java	
Abstractness (avg/max per packageFragment)		0	0	0	/Listing_Tree_Node/src	
Normalized Distance (avg/max per packageFragment)		0	0	0	/Listing_Tree_Node/src	
Number of Static Methods (avg/max per type)	1	0.2	0.4	1	/Listing_Tree_Node/src/main.java	
Number of Interfaces (avg/max per packageFragment)	0	0	0	0	/Listing_Tree_Node/src	
Total Lines of Code	293					
Weighted methods per Class (avg/max per type)	57	11.4	8.754	26	/Listing_Tree_Node/src/List.java	
Number of Methods (avg/max per type)	41	8.2	5.741	15	/Listing_Tree_Node/src/List.java	
Depth of Inheritance Tree (avg/max per type)		1	0	1	/Listing_Tree_Node/src/main.java	

1

2

Metrics 예제



1

2

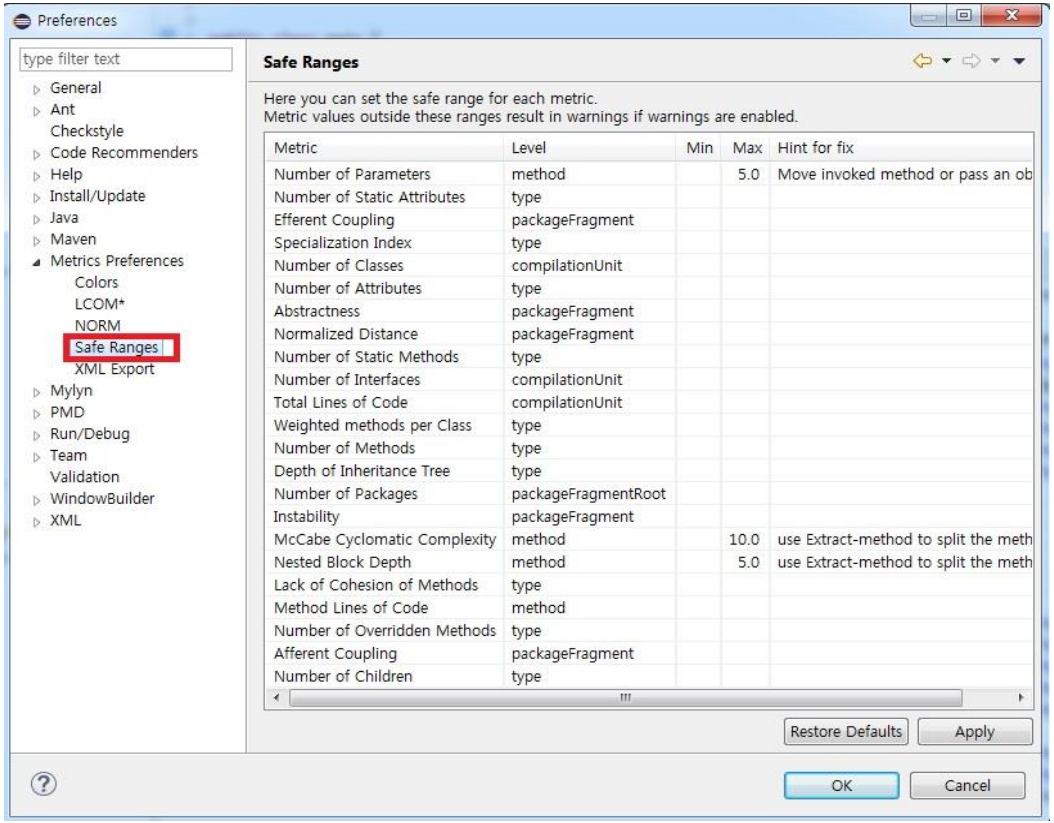
Metrics 예제

```
<?xml version="1.0" encoding="UTF-8"?>
- <Metrics xmlns="http://metrics.sourceforge.net/2003/Metrics-First-Flat" date="2015-04-17" type="Project" scope="Listing_Tree_Node">
  - <Metric hint="Move invoked method or pass an object" max="5" description="Number of Parameters" id="PAR">
    - <Values max="3" stddev="0.75" avg="0.643" per="method">
        <Value value="3" package="(default package)" source="ListNode.java" name="ListNode"/>
        <Value value="2" package="(default package)" source="List.java" name="addAfter"/>
        <Value value="2" package="(default package)" source="List.java" name="addBefore"/>
        <Value value="2" package="(default package)" source="List.java" name="set"/>
        <Value value="2" package="(default package)" source="TreeNode.java" name="TreeNode"/>
        <Value value="1" package="(default package)" source="List.java" name="add_first"/>
        <Value value="1" package="(default package)" source="List.java" name="add_last"/>
        <Value value="1" package="(default package)" source="List.java" name="next"/>
        <Value value="1" package="(default package)" source="List.java" name="prev"/>
        <Value value="1" package="(default package)" source="List.java" name="remove"/>
        <Value value="1" package="(default package)" source="ListNode.java" name="set_Element"/>
        <Value value="1" package="(default package)" source="ListNode.java" name="set_Next"/>
        <Value value="1" package="(default package)" source="ListNode.java" name="set_Prev"/>
        <Value value="1" package="(default package)" source="Tree.java" name="setNodeListPreoder"/>
        <Value value="1" package="(default package)" source="TreeNode.java" name="addLeftChild"/>
        <Value value="1" package="(default package)" source="TreeNode.java" name="addRightChild"/>
        <Value value="1" package="(default package)" source="TreeNode.java" name="setElement"/>
        <Value value="1" package="(default package)" source="TreeNode.java" name="setLeftChild"/>
        <Value value="1" package="(default package)" source="TreeNode.java" name="setParent"/>
        <Value value="1" package="(default package)" source="TreeNode.java" name="setRightChild"/>
        <Value value="1" package="(default package)" source="main.java" name="main"/>
        <Value value="0" package="(default package)" source="List.java" name="List"/>
        <Value value="0" package="(default package)" source="List.java" name="Size"/>
        <Value value="0" package="(default package)" source="List.java" name="first"/>
        <Value value="0" package="(default package)" source="List.java" name="isEmpty"/>
```

1

2

Metrics 예제

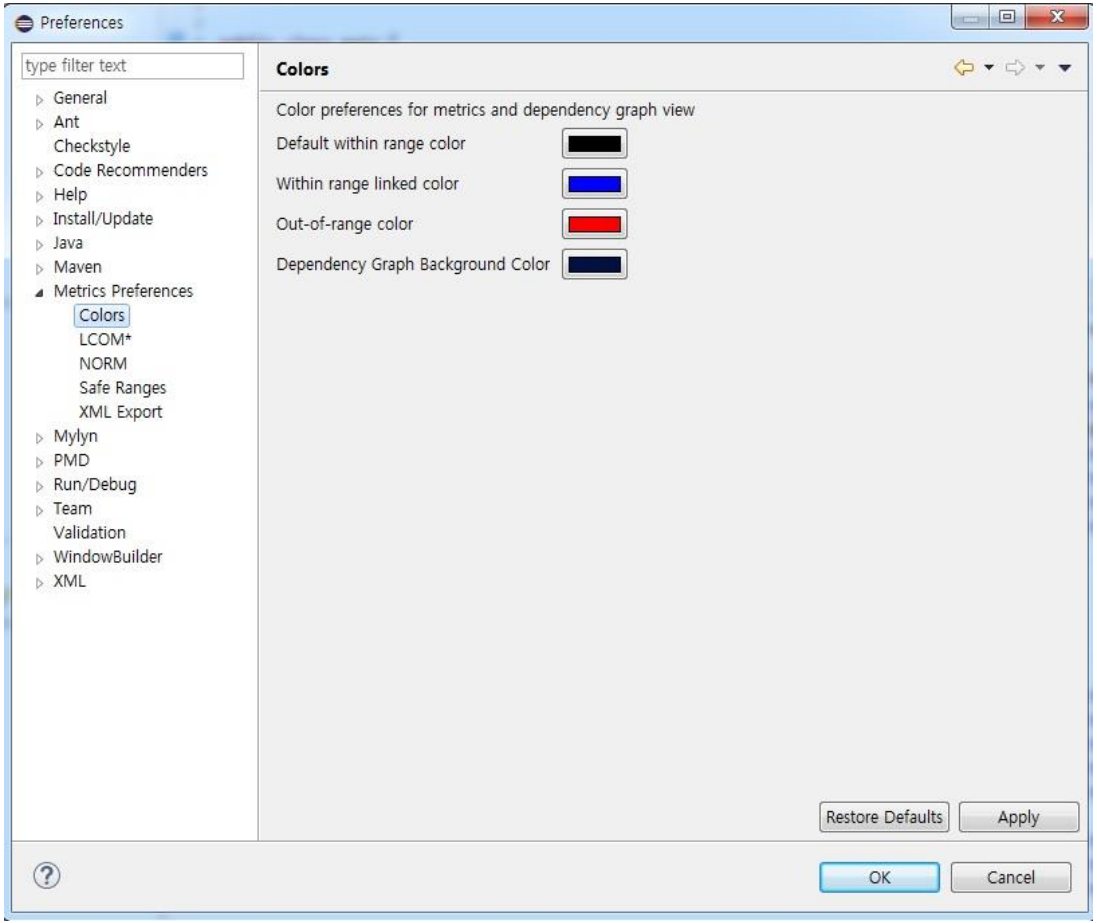


Window→ Properties → Metrics

1

2

Metrics 예제



2

시스템테스트 도구

2-0 Pairwise Test

- Pairwise Test란?

2-1 Allpairs

- Allpairs란?
- Allpairs 설치
- Allpairs 예제

2-2 PICT

- PICT란?
- PICT 설치
- PICT 예제

시스템테스트 도구

Pairwise Test

Pairwise Test란?

- 모든 입력 값들의 조합을 테스트하면 케이스가 너무 많아져서 비현실적이다.

   테스트 범위를 줄이는 대신 테스트에 드는 시간과 노력을 줄일 수 있다!
- 많은 결함이 1개, 2개 또는 3개의 입력 값들의 상호 작용에 의해 발생한다.

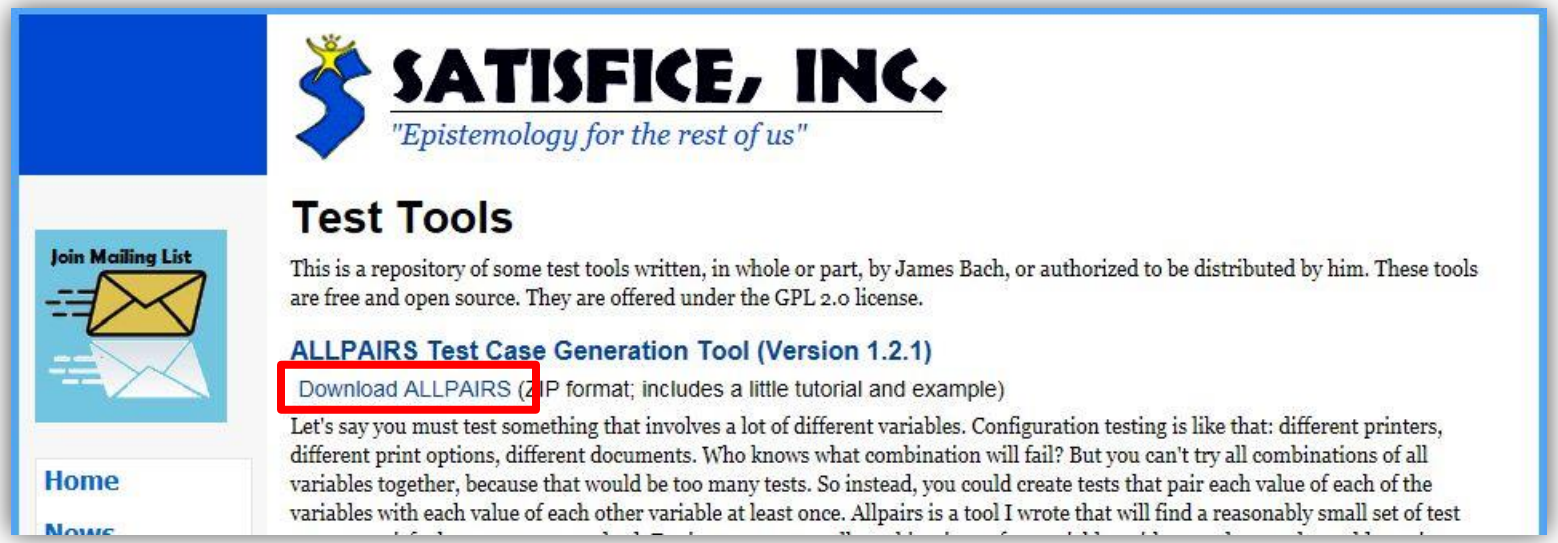
  짝들의 조합을 테스트하여 비슷한 효율의 테스팅 가능!
- 대신 모든 결함을 다 찾아낼 순 없다.

Allpairs란?

- Allpairs은 적용 범위 기준을 만족하는 테스트 케이스의 작은 크기의 세트를 합리적으로 찾을 수 있는 공개 도구 (이때의 적용 범위 기준은 Pairwise 기법)
- Perl 스크립트를 기반으로 실행할 수 있는 command-line
- Perl 스크립트로 작성되어 복잡한 다차원 배열 계산에 시간이 걸림
- Pair 계산만 가능(Triple 계산 불가)

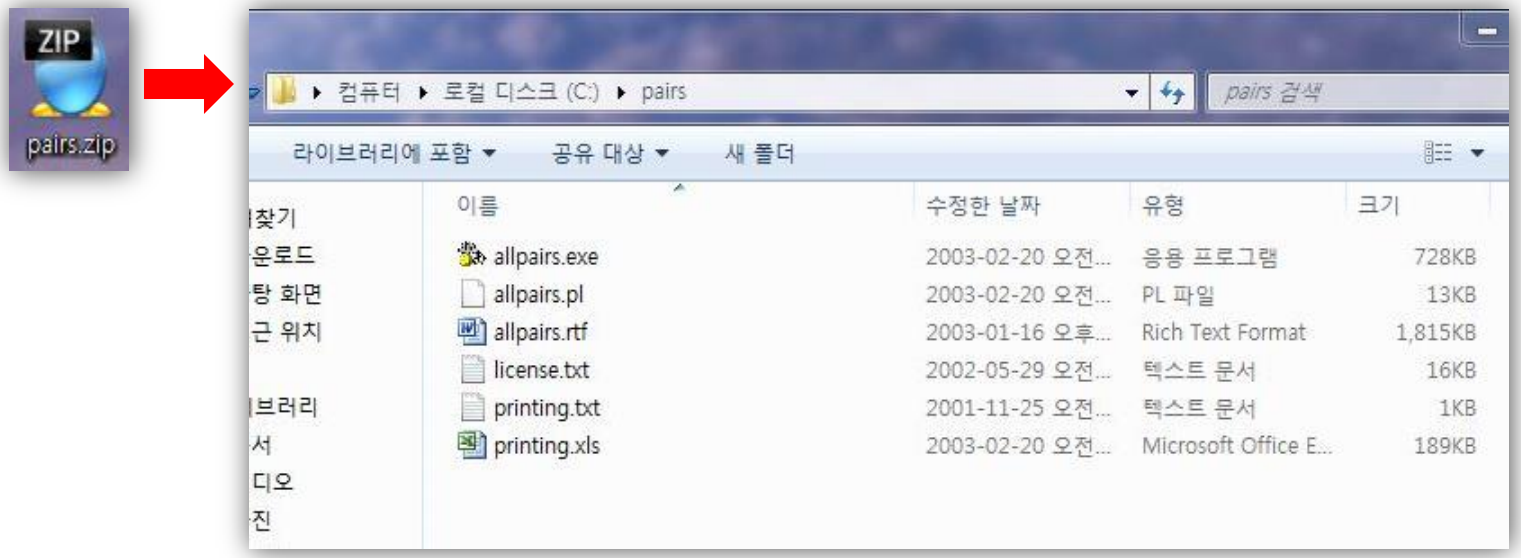
Allpairs 설치(1/2)

- http://www.satisfice.com/tools.shtml



Allpairs 설치(2/2)

- 압축파일 ➜ 원하는 위치에 압축 해제



시스템테스트 도구

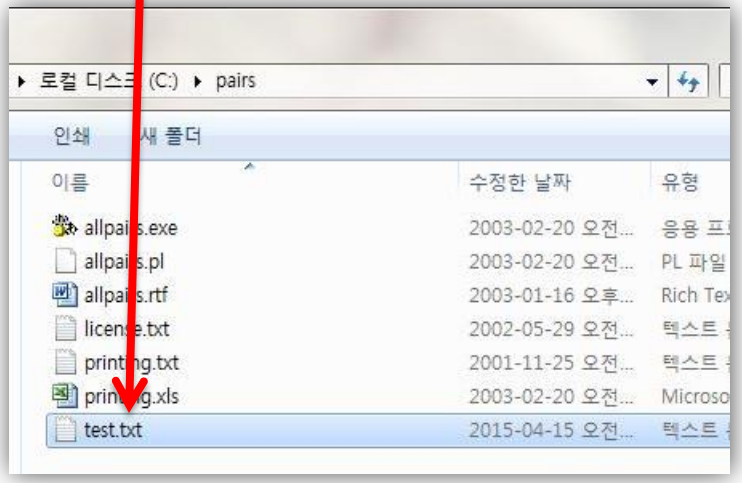
Allpairs

Allpairs 예제(1/5)

변수 테이블 준비

- 텍스트 파일로 pairs 폴더에 저장
- 엑셀로 작성하면 편리함

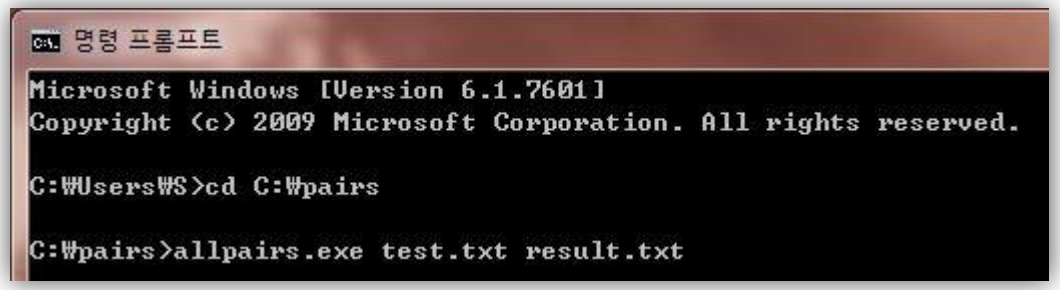
OS	Browser	Website
WinXP	IE9	Google
Win7	IE10	Naver
Win8	IE11	Daum
OS X	Crome(39)	Wikipedia
	FireFox(35)	



Allpairs 예제(2/5)

조합 생성

- cmd 실행 > pairs 디렉터리 접근 > 'allpairs.exe table.txt result.txt' 같은 형식으로 입력



시스템테스트 도구

Allpairs

Allpairs 예제(3/5)

조합 생성 결과

- 오른쪽과 같이 결과를 보여줌

TEST CASES

case	OS	Browser	Website	pairings
1	WinXP	IE9	Google	3
2	Win7	IE9	Naver	3
3	Win8	IE9	Daum	3
4	OS X	IE9	Wikipedia	3
5	WinXP	IE10	Naver	3
6	Win7	IE10	Google	3
7	Win8	IE10	Wikipedia	3
8	OS X	IE10	Daum	3
9	WinXP	IE11	Daum	3
10	Win7	IE11	Wikipedia	3
11	Win8	IE11	Google	3
12	OS X	IE11	Naver	3
13	WinXP	Crome(39)	Wikipedia	3
14	Win7	Crome(39)	Daum	3
15	Win8	Crome(39)	Naver	3
16	OS X	Crome(39)	Google	3
17	WinXP	FireFox(35)	Google	2
18	Win7	FireFox(35)	Naver	2
19	Win8	FireFox(35)	Daum	2
20	OS X	FireFox(35)	Wikipedia	2

PAIRING DETAILS

var1	var2	value1	value2	appearances	cases
Browser	OS	IE9	WinXP	1	1
Browser	OS	IE9	Win7	1	2
Browser	OS	IE9	Win8	1	3
Browser	OS	IE9	OS X	1	4
Browser	OS	IE10	WinXP	1	5
Browser	OS	IE10	Win7	1	6
Browser	OS	IE10	Win8	1	7
Browser	OS	IE10	OS X	1	8
Browser	OS	IE11	WinXP	1	9
Browser	OS	IE11	Win7	1	10
Browser	OS	IE11	Win8	1	11
Browser	OS	IE11	OS X	1	12
Browser	OS	Crome(39)	WinXP	1	13
Browser	OS	Crome(39)	Win7	1	14
Browser	OS	Crome(39)	Win8	1	15
Browser	OS	Crome(39)	OS X	1	16
Browser	OS	FireFox(35)	WinXP	1	17

시스템테스트 도구

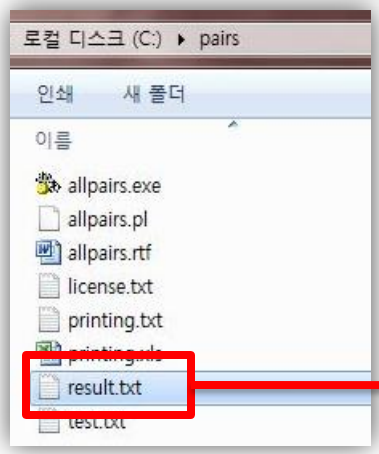
Allpairs

Allpairs 예제(4/5)

텍스트 파일로 조합 생성

- 'allpairs.exe table.txt > result.txt' 같은 형식으로 입력

```
C:₩pairs>allpairs.exe test.txt > result.txt
```



result.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

TEST CASES

case	OS	Browser	Website	pairings	
1	WinXP	IE9	Google	3	
2	Win7	IE9	Naver	3	
3	Win8	IE9	Daum	3	
4	OS X	IE9	Wikipedia		3
5	WinXP	IE10	Naver	3	
6	Win7	IE10	Google	3	
7	Win8	IE10	Wikipedia		3
8	OS X	IE10	Daum	3	
9	WinXP	IE11	Daum	3	
10	Win7	IE11	Wikipedia		3
11	Win8	IE11	Google	3	
12	OS X	IE11	Naver	3	
13	WinXP	Crome(39)	Wikipedia		3
14	Win7	Crome(39)	Daum	3	
15	Win8	Crome(39)	Naver	3	
16	OS X	Crome(39)	Google	3	
17	WinXP	FireFox(35)	Google	2	
18	Win7	FireFox(35)	Naver	2	
19	Win8	FireFox(35)	Daum	2	
20	OS X	FireFox(35)	Wikipedia		2

PAIRING DETAILS

var1	var2	value1	value2	appearances	cases
Browser	OS	IE9	WinXP	1	1

시스템테스트 도구

Allpairs

Allpairs 예제(5/5)

조합

- pairings
  : 유일한 쌍의 개수
  (한 쌍은 두 값의 pair)
- var : variable
- value : 변수의 값
  appearances : 해당 조합이 나타나는 횟수
- cases : 해당 조합이 들어간 case

TEST CASES

case	OS	Browser	Website	pairings
1	WinXP	IE9	Google	3
2	Win7	IE9	Naver	3
3	Win8	IE9	Daum	3
4	OS X	IE9	Wikipedia	3
5	WinXP	IE10	Naver	3
6	Win7	IE10	Google	3
7	Win8	IE10	Wikipedia	3
8	OS X	IE10	Daum	3
9	WinXP	IE11	Daum	3
10	Win7	IE11	Wikipedia	3
11	Win8	IE11	Google	3
12	OS X	IE11	Naver	3
13	WinXP	Crome(39)	Wikipedia	3
14	Win7	Crome(39)	Daum	3
15	Win8	Crome(39)	Naver	3
16	OS X	Crome(39)	Google	3
17	WinXP	FireFox(35)	Google	2
18	Win7	FireFox(35)	Naver	2
19	Win8	FireFox(35)	Daum	2
20	OS X	FireFox(35)	Wikipedia	2

PAIRING DETAILS

var1	var2	value1	value2	appearances	cases
Browser	OS	IE9	WinXP	1	1
Browser	OS	IE9	Win7	1	2
Browser	OS	IE9	Win8	1	3
Browser	OS	IE9	OS X	1	4
Browser	OS	IE10	WinXP	1	5
Browser	OS	IE10	Win7	1	6
Browser	OS	IE10	Win8	1	7
Browser	OS	IE10	OS X	1	8
Browser	OS	IE11	WinXP	1	9
Browser	OS	IE11	Win7	1	10
Browser	OS	IE11	Win8	1	11
Browser	OS	IE11	OS X	1	12
Browser	OS	Crome(39)	WinXP	1	13
Browser	OS	Crome(39)	Win7	1	14
Browser	OS	Crome(39)	Win8	1	15
Browser	OS	Crome(39)	OS X	1	16
Browser	OS	FireFox(35)	WinXP	1	17
Browser	OS	FireFox(35)	Win7	1	18
Browser	OS	FireFox(35)	Win8	1	19
Browser	OS	FireFox(35)	OS X	1	20
Browser	Website	IE9	Google	1	1
Browser	Website	IE9	Naver	1	2
Browser	Website	IE9	Daum	1	3
Browser	Website	IE9	Wikipedia	1	4
Browser	Website	IE10	Google	1	6
Browser	Website	IE10	Naver	1	5
Browser	Website	IE10	Daum	1	8
Browser	Website	IE10	Wikipedia	1	7
Browser	Website	IE11	Google	1	11
Browser	Website	IE11	Naver	1	12
Browser	Website	IE11	Daum	1	9
Browser	Website	IE11	Wikipedia	1	10
Browser	Website	Crome(39)	Google	1	16
Browser	Website	Crome(39)	Naver	1	15
Browser	Website	Crome(39)	Daum	1	14
Browser	Website	Crome(39)	Wikipedia	1	13
Browser	Website	FireFox(35)	Google	1	17
Browser	Website	FireFox(35)	Naver	1	18
Browser	Website	FireFox(35)	Daum	1	19
Browser	Website	FireFox(35)	Wikipedia	1	20
OS	Website	WinXP	Google	2	1, 17
OS	Website	WinXP	Naver	1	5
OS	Website	WinXP	Daum	1	9
OS	Website	WinXP	Wikipedia	1	13
OS	Website	Win7	Google	1	6
OS	Website	Win7	Naver	2	2, 18
OS	Website	Win7	Daum	1	14
OS	Website	Win7	Wikipedia	1	10
OS	Website	Win8	Google	1	11
OS	Website	Win8	Naver	1	15
OS	Website	Win8	Daum	2	3, 19
OS	Website	Win8	Wikipedia	1	7
OS	Website	OS X	Google	1	16
OS	Website	OS X	Naver	1	12
OS	Website	OS X	Daum	1	8
OS	Website	OS X	Wikipedia	2	4, 20

시스템테스트 도구

PICT

PICT란?

- Allpairs같은 Testcase generation tool
- Command-line에서 실행 가능
- 엑셀 파일로 결과 생성 가능
- Allpairs보다 value table 작성이 좀 더 쉽다

시스템테스트 도구

PICT

설치

http://download.microsoft.com/download/f/5/5/f55484df-8494-48fa-8dbd-8c6f76cc014b/pict33.msi

PICT 설치파일 다운받아 설치하면, cmd에서 자동 실행.

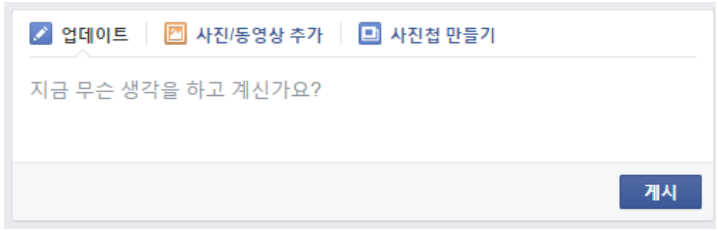
시스템테스트 도구

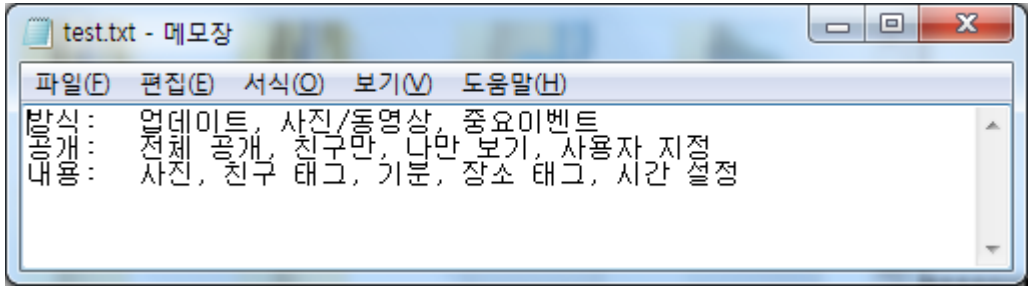
PICT

PICT 예제(1)

Value table 작성

형식 : <CategoryName>:<*TAB*><Value1>, <Value2>, <Value3>, ...

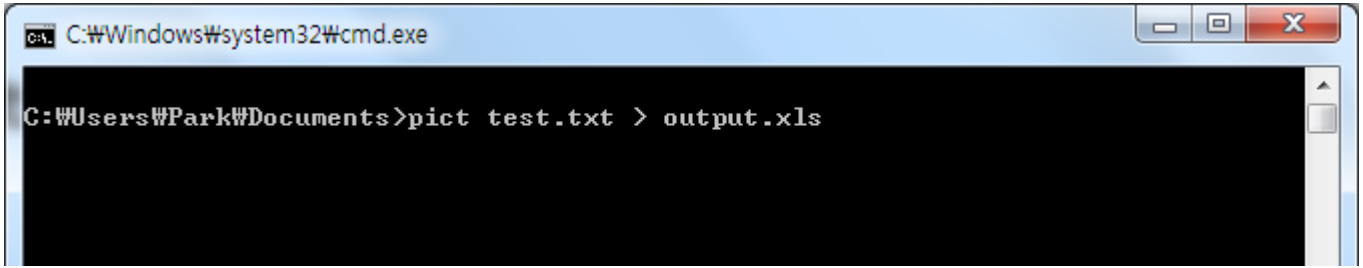




PICT 예제(2)

pict “input file” > “output file”

결과 파일은 엑셀, 텍스트 파일 가능.



시스템테스트 도구

PICT

PICT 예제(3)

60개의 테스트 케이스

➡ 21개의 테스트 케이스 생성!

1	방식	공개	내용
2	업데이트	나만 보기	시간 설정
3	업데이트	친구만	친구 태그
4	사진/동영상	나만 보기	장소 태그
5	중요이벤트	친구만	장소 태그
6	중요이벤트	전체 공개	시간 설정
7	중요이벤트	나만 보기	사진
8	사진/동영상	사용자 지정	시간 설정
9	업데이트	사용자 지정	기분
10	중요이벤트	사용자 지정	친구 태그
11	사진/동영상	친구만	기분
12	업데이트	친구만	사진
13	중요이벤트	전체 공개	기분
14	사진/동영상	사용자 지정	사진
15	사진/동영상	나만 보기	친구 태그
16	업데이트	사용자 지정	장소 태그
17	사진/동영상	전체 공개	사진
18	업데이트	친구만	시간 설정
19	업데이트	전체 공개	장소 태그
20	업데이트	전체 공개	친구 태그
21	업데이트	나만 보기	기분

참고·참조

출처: http://techbard.tistory.com/3

출처: http://www.satisfice.com/tools.shtml
출처: http://gyanni.tistory.com/47

출처: http://iamsungeun.blog.me/100098723655

출처: http://mryou.tistory.com/449

Q&A

Question and Answer

감사합니다 ☺