

SOFTWARE VERIFICATION

System Test

2005 | 1342 | 이도현
2006 | 1000 | 안병욱
2007 | 10069 | 전창완
2009 | 2432 | 김다영

INDEX



- | Analysis
- | Category-Partition Testing
- | Pair-wise Testing
- | Static Analysis Test
 - | Sonar
 - | JArchitect
 - | Clover
 - | JDepend
 - | Eclipse TPTP

Analysis

Analysis

| Documents

/ T4 - OSP Stage 2040 for ASAP Paint V2.0 사용

/ 기존 1.0에서 삭제된 부분이 있으나 크게 기능에 미치는 영향이 없다고 판단, 기존 Test Case 그대로 적용

Category-Partition Testing

Category-Partition Testing

| Source

/ 매 테스트마다, Snapshot을 Branches안에 생성
6/3, r119에서 추출된 snapshot으로 테스트 실행

```
waniui-Mac-mini:branches wani$ ls
total 0
drwxr-xr-x  6 wani  staff  204  6  3 10:48 .
drwxr-xr-x 10 wani  staff  340  6  3 10:48 ..
drwxr-xr-x  8 wani  staff  272  6  3 10:48 .svn
drwxr-xr-x 12 wani  staff  408  6  3 10:48 snapshot130514
drwxr-xr-x 12 wani  staff  408  6  3 10:48 snapshot130520
drwxr-xr-x 12 wani  staff  408  6  3 10:48 snapshot130603
waniui-Mac-mini:branches wani$
```

Category-Partition Testing

| Mantis

/ 지난번 테스트에서 에러난 부분은 전부 해결된 것을 확인 할 수 있었음

해결됨 [^] (1 - 10 / 12)	
0000037 	이미지 회전 작동 안함 General - 2013-05-27 18:01
0000039 	복사 후 붙여넣기 수행시 General - 2013-05-27 17:37
0000036 	파일 불러온 후 이미지 회전 General - 2013-05-27 08:59
0000040 	이미지 사이즈 변경 할때 정사각형이 아니면 불완전하게 동작 General - 2013-05-27 08:47
0000035 	새파일 때 저장 이후 새파일이 되지 않아요. General - 2013-05-27 08:46
0000038 	실행하자마자 붙여넣기 하면 발생하는 에러 General - 2013-05-27 08:44
0000041 	연필 툴 에러.. General - 2013-05-27 08:40
0000005 	상윤님 너무 고생이 많으세요 π.π General - 2013-05-10 09:49
0000001 	c2 관련 [모든 프로젝트] General - 2013-05-03 04:17
0000004 	날씨가 좋은데 시험기간이네여 General - 2013-05-02 12:14

Category-Partition Testing

| Test Case 1. 시작방식

Test Case	시작방식	저장할까요?	기존파일저장유무	종료	결과
1.1.1	새 파일	O	O	X	Pass
1.1.2	새 파일	O	O	O	Pass
1.1.3	새 파일	O	X	X	Pass
1.1.4	새 파일	O	X	O	Pass
1.1.5	새 파일	X	O	X	Pass
1.1.6	새 파일	X	O	O	Pass
1.1.7	새 파일	X	X	X	Pass
1.1.8	새 파일	X	X	O	Pass
1.2.1	불러오기	O	O	X	Pass
1.2.2	불러오기	O	O	O	Pass
1.2.3	불러오기	O	X	X	Pass
1.2.4	불러오기	O	X	O	Pass
1.2.5	불러오기	X	O	X	Pass
1.2.6	불러오기	X	O	O	Pass
1.2.7	불러오기	X	X	X	Pass
1.2.8	불러오기	X	X	O	Pass

Category-Partition Testing

| Test Case2. 저장하기

Test Case	새 파일로 시작여부	메뉴	결과
2.1	○	저장하기	Pass
2.2	○	다른 이름으로 저장	Pass
2.3	X	저장하기	Pass
2.4	X	다른 이름으로 저장	Pass

Category-Partition Testing

| Test Case3. 이미지기능

Test Case	영역선택유무	메뉴	클립보드유무	결과
3.1.1	O	이미지복사	O	Pass
3.1.2	O	잘라내기	O	Pass
3.1.3	O	이동	O	Pass
3.1.4	O	삭제	O	Pass
3.1.5	O	회전	O	FAIL
3.1.6	O	반전	O	Pass
3.1.7	O	이미지복사	X	Pass
3.1.8	O	잘라내기	X	Pass
3.1.9	O	이동	X	Pass
3.1.10	O	삭제	X	Pass
3.1.11	O	회전	X	FAIL
3.1.12	O	반전	X	Pass
3.2.1	X	이동	O	Pass
3.2.2	X	삭제	O	Pass
3.2.3	X	회전	O	FAIL
3.2.4	X	반전	O	Pass
3.2.5	X	이동	X	Pass
3.2.6	X	삭제	X	Pass
3.2.7	X	회전	X	FAIL
3.2.8	X	반전	X	Pass

Category-Partition Testing

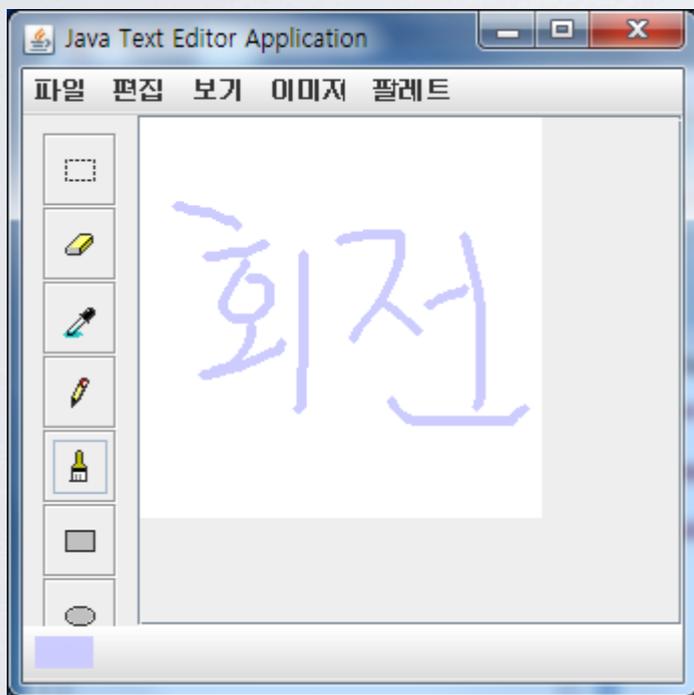
I Test Case3. 이미지기능

- / 이미지 회전 기능은 회전기능만 정상작동
 - / 각도 지정(Function 4.1.2) 이후 이미지열화현상 발생
 - / Test Case : 3.1.5, 3.1.11, 3.2.3, 3.2.7
 - / 멘티스에도 보고된 내용이라 일단 보류

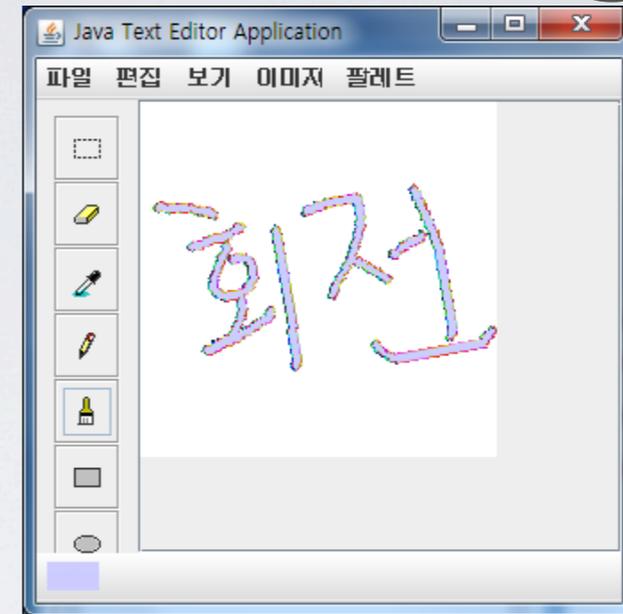
☐ 메모	
(0000033) uranium (관리자) 2013-05-27 08:41 편집 삭제 비공개로 하기	원인 확인, 회전 각도가 int값으로 넘겨져 소수점 연산이 안되는 문제로 발견. 해당 부분에 대해 충분히 해결 가능함.
(0000041) administrator (관리자) [비공개] 2013-05-27 18:00 편집 삭제 공개로 하기	issue 0000037 완료됨.
(0000042) uranium (관리자) 2013-05-27 18:01 편집 삭제 비공개로 하기	이미지 회전이 정상적으로 작동하지 않는 문제. 알고리즘을 완전히 들어내고 새로 집어넣음으로 해결함. 기존의 알고리즘에는 점 보간과 관련된 내용이 존재하지 않아 이를 추가하였음. 그래도 여전히 이미지 열화현상이 일부 존재함.

Category-Partition Testing

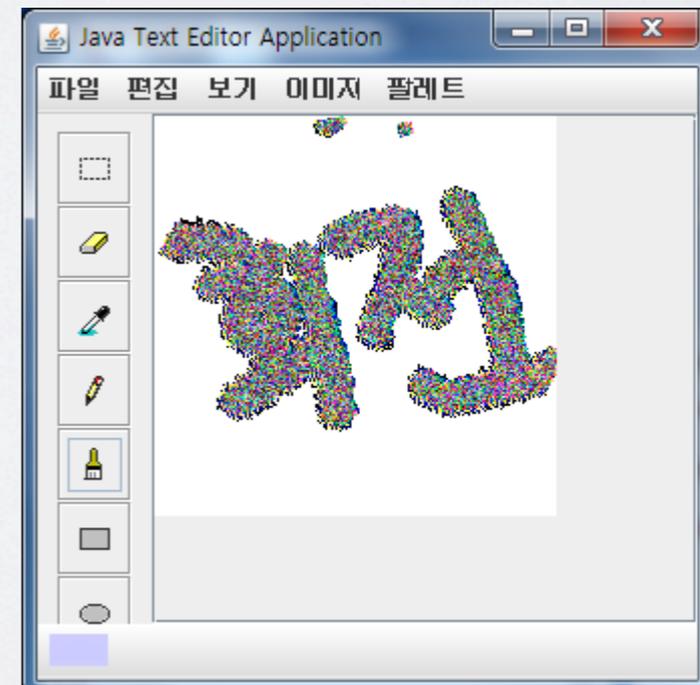
| Test Case3. 이미지기능 (After)



10도 회전 시 이미지 열화확인



1도씩 10번 수행했을 때 결과



Category-Partition Testing

| Test Case4. 붙여넣기

Test Case	이전동작	클립 보드 유무	메뉴	결과
4.1.1	복사	○	붙여 넣기	Pass
4.1.2	잘라내기	○	붙여 넣기	Pass
4.1.3	-	X	붙여 넣기	Pass

Category-Partition Testing

| Test Case5. 이미지 리사이즈

Test Case	Width	비교	height	결과
5.1.1	숫자	>	숫자	Pass
5.1.2	숫자	=	숫자	Pass
5.1.3	숫자	<	숫자	Pass
5.2.1	문자/특수문자		문자/특수문자	Pass

Category-Partition Testing

| Test Case6. 그리기

Test Case	색상 선택 방법	도구	결과
6.1.1	직접선택	연필	Pass
6.1.2	직접선택	브러시	Pass
6.1.3	직접선택	사각형	Pass
6.1.4	직접선택	원	Pass
6.2.1	스포이드	연필	Pass
6.2.2	스포이드	브러시	Pass
6.2.3	스포이드	사각형	Pass
6.2.4	스포이드	원	Pass
6.3.1	default	연필	Pass
6.3.2	default	브러시	Pass
6.3.3	default	사각형	Pass
6.3.4	default	원	Pass

Category-Partition Testing

| Test Case 7. 확대 축소

Test Case	확대	축소	결과
7.1	0	1	Pass
7.2	1	0	Pass
7.3	1	1	Pass
7.4	1	n	Pass
7.5	n	0	Pass
7.6	n	1	Pass
7.7	n	n	Pass

Pairwise Testing

Pairwise Testing

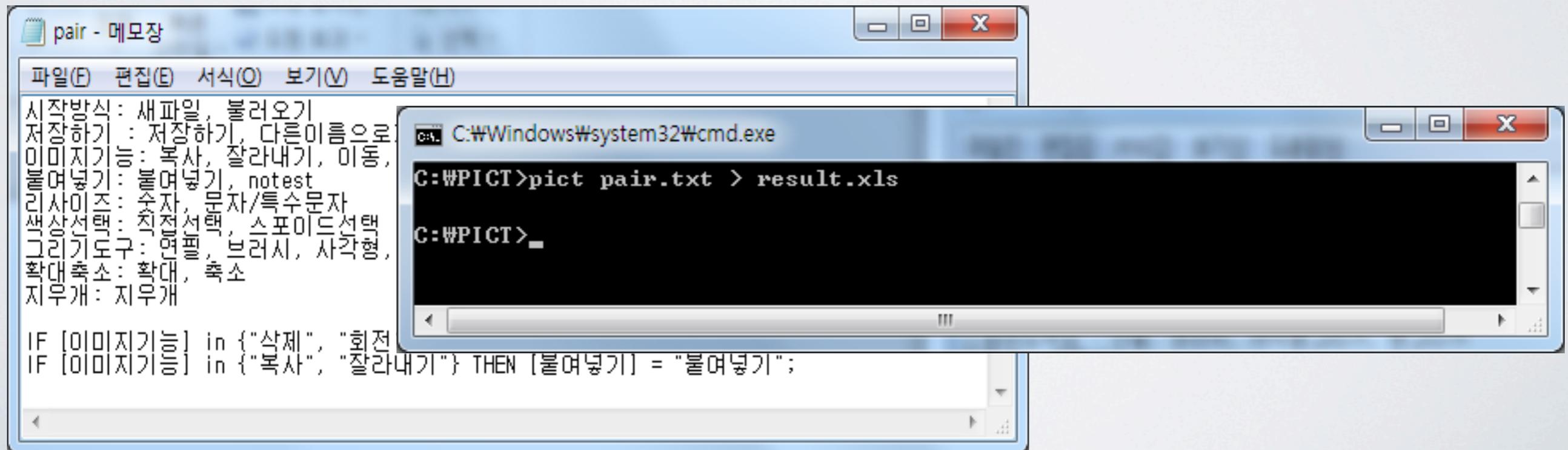
- | Category-Partition Testing을 이용해 뽑아낸 Category들을 이용하여 Pairwise Testing을 시행.
- | 이미지기능의 회전기능은 오류가 있는 기능으로 제외함

시작방식	저장하기	이미지기능	붙여넣기	리사이즈	색상선택	그리기도구	확대축소	지우개
새파일	저장하기	복사	붙여넣기	숫자	직접선택	연필	확대	지우개
불러오기	다른이름으로 저장	잘라내기		문자/특수문 자	스포이드	브러시	축소	
		이동				사각형		
		삭제				원		
		회전						
		반전						

Pairwise Testing

PICT

- 붙여넣기는 영역선택에서 복사와 잘라내기에서만 가능하므로 조건절에 표시해주어야 한다.
- 출력시엔 보기 편하게 엑셀파일로 출력



Pairwise Testing

PICT

20가지의 pair가 출력, Notest부분은 따로 test하지 않아도 된다는 의미로 사용

시작방식	저장하기	이미지기능	붙여넣기	리사이즈	색상선택	그리기도구	확대축소	지우개
새파일	저장하기	삭제	notest	숫자	스포이드선택	원	확대	지우개
불러오기	다른이름으로저장	복사	붙여넣기	문자/특수문자	직접선택	브러시	축소	지우개
불러오기	저장하기	이동	notest	문자/특수문자	스포이드선택	사각형	축소	지우개
새파일	다른이름으로저장	반전	notest	숫자	직접선택	원	축소	지우개
새파일	저장하기	복사	붙여넣기	숫자	직접선택	사각형	확대	지우개
불러오기	다른이름으로저장	잘라내기	붙여넣기	숫자	스포이드선택	브러시	확대	지우개
새파일	다른이름으로저장	잘라내기	붙여넣기	문자/특수문자	직접선택	사각형	축소	지우개
불러오기	다른이름으로저장	복사	붙여넣기	문자/특수문자	스포이드선택	원	확대	지우개
새파일	다른이름으로저장	이동	notest	숫자	직접선택	원	확대	지우개
불러오기	저장하기	반전	notest	문자/특수문자	스포이드선택	사각형	확대	지우개
불러오기	다른이름으로저장	삭제	notest	문자/특수문자	직접선택	사각형	축소	지우개
불러오기	다른이름으로저장	삭제	notest	문자/특수문자	스포이드선택	연필	확대	지우개
새파일	저장하기	이동	notest	숫자	직접선택	브러시	축소	지우개
새파일	저장하기	반전	notest	숫자	직접선택	연필	축소	지우개
새파일	저장하기	잘라내기	붙여넣기	숫자	스포이드선택	원	축소	지우개
불러오기	다른이름으로저장	삭제	notest	숫자	스포이드선택	브러시	확대	지우개
불러오기	저장하기	반전	notest	문자/특수문자	스포이드선택	브러시	확대	지우개
새파일	다른이름으로저장	복사	붙여넣기	문자/특수문자	스포이드선택	연필	축소	지우개
새파일	다른이름으로저장	이동	notest	숫자	스포이드선택	연필	축소	지우개
새파일	저장하기	잘라내기	붙여넣기	문자/특수문자	스포이드선택	연필	확대	지우개

Pairwise Testing

PICT

20가지의 Test모두 정상적으로 작동

시작방식	저장하기	이미지기능	붙여넣기	리사이즈	색상선택	그리기도구	확대축소	지우개	결과
새파일	저장하기	삭제	notest	숫자	스포이드선택	원	확대	지우개	○
불러오기	다른이름으로저장	복사	붙여넣기	문자/특수문자	직접선택	브러시	축소	지우개	○
불러오기	저장하기	이동	notest	문자/특수문자	스포이드선택	사각형	축소	지우개	○
새파일	다른이름으로저장	반전	notest	숫자	직접선택	원	축소	지우개	○
새파일	저장하기	복사	붙여넣기	숫자	직접선택	사각형	확대	지우개	○
불러오기	다른이름으로저장	잘라내기	붙여넣기	숫자	스포이드선택	브러시	확대	지우개	○
새파일	다른이름으로저장	잘라내기	붙여넣기	문자/특수문자	직접선택	사각형	축소	지우개	○
불러오기	다른이름으로저장	복사	붙여넣기	문자/특수문자	스포이드선택	원	확대	지우개	○
새파일	다른이름으로저장	이동	notest	숫자	직접선택	원	확대	지우개	○
불러오기	저장하기	반전	notest	문자/특수문자	스포이드선택	사각형	확대	지우개	○
불러오기	다른이름으로저장	삭제	notest	문자/특수문자	직접선택	사각형	축소	지우개	○
불러오기	다른이름으로저장	삭제	notest	문자/특수문자	스포이드선택	연필	확대	지우개	○
새파일	저장하기	이동	notest	숫자	직접선택	브러시	축소	지우개	○
새파일	저장하기	반전	notest	숫자	직접선택	연필	축소	지우개	○
새파일	저장하기	잘라내기	붙여넣기	숫자	스포이드선택	원	축소	지우개	○
불러오기	다른이름으로저장	삭제	notest	숫자	스포이드선택	브러시	확대	지우개	○
불러오기	저장하기	반전	notest	문자/특수문자	스포이드선택	브러시	확대	지우개	○
새파일	다른이름으로저장	복사	붙여넣기	문자/특수문자	스포이드선택	연필	축소	지우개	○
새파일	다른이름으로저장	이동	notest	숫자	스포이드선택	연필	축소	지우개	○
새파일	저장하기	잘라내기	붙여넣기	문자/특수문자	스포이드선택	연필	확대	지우개	○

Sonar

Sonar

| 설치법

/ Sonar

<http://www.sonarsource.org/downloads/>

sonar

[Download](#) [Features](#) [Screenshots](#) [Forge](#) [Get Support](#) [Get Involved](#) [Roadmap](#) [Blog](#)

Download

3.5.1	April 3, 2013	Fix two regressions when using Sonar Runner to perform analyses in specific cases. (release notes , screenshots)	Download (md5)
-------	---------------	--	--------------------------------

Sonar

| 설치법

^ <http://docs.codehaus.org/display/SONAR/Installing+and+Configuring+Sonar+Runner>

Installing and Configuring Sonar Runner



Added by [David RACODON](#), last edited by [David RACODON](#) on May 21, 2013 ([view change](#))

Name	Sonar Runner
Latest version	2.2.1 (21 May 2013)
Requires Sonar version	3.0 or higher (check version compatibility)
Download	http://repo1.maven.org/maven2/org/codehaus/sonar/runner/sonar-runner-dist/2.2.1/sonar-runner-dist-2.2.1.zip
License	GNU LGPL 3
Developers	Evgeny Mandrikov, Simon Brandhof, Fabrice Bellingard
Issue tracker	http://jira.codehaus.org/browse/SONARPLUGINS/component/14640
Sources	https://github.com/Sonarsource/sonar-runner

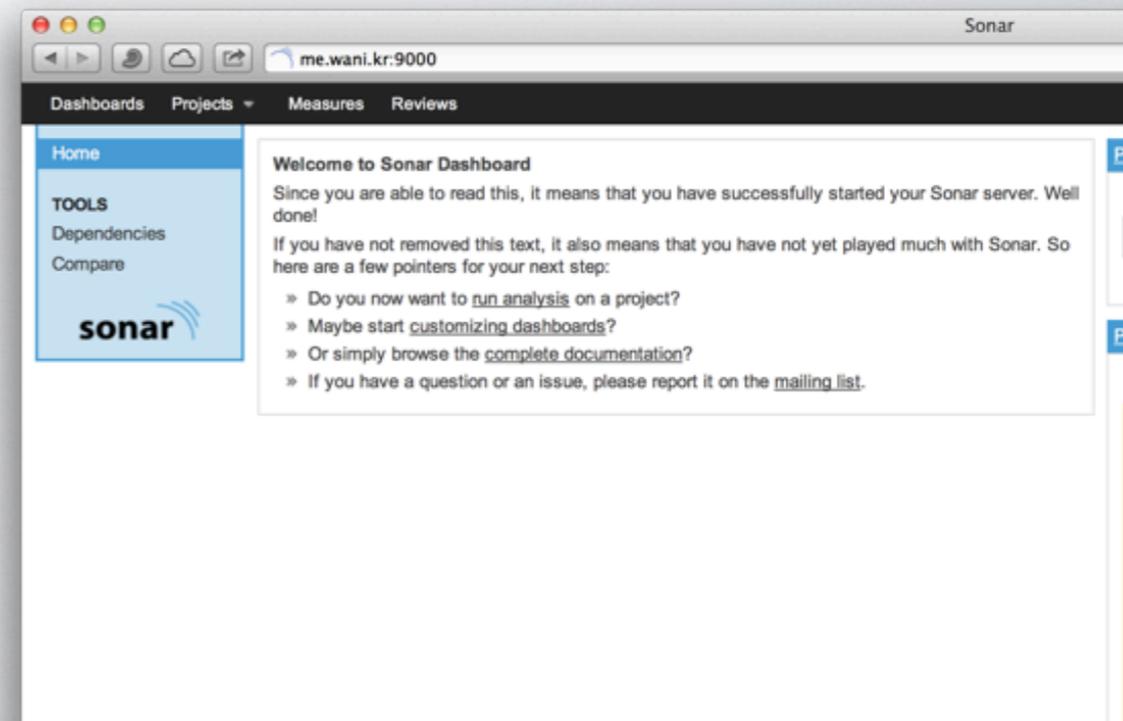
Sonar

| 설치법 (Sonar)

/ 원하는 디렉토리에 풀고, 다음 파일 실행하면 설치 완료

/ sonar-3.5.1/bin/[my os]/sonar.sh

/ localhost:9000 으로 접속가능



Sonar

| 설치법 (Sonar-runner)

/ 원하는 디렉토리에 풀고, 다음과 같이 환경변수 설정

```
4  
5 export SONAR_RUNNER_HOME='/Users/wani/sonar-runner-2.2.1'  
6  
7 export PATH="/Users/wani/cc/cruisecontrol-2.8.4/main/bin:$PATH"  
8 export PATH="$SONAR_RUNNER_HOME/bin:$PATH"
```

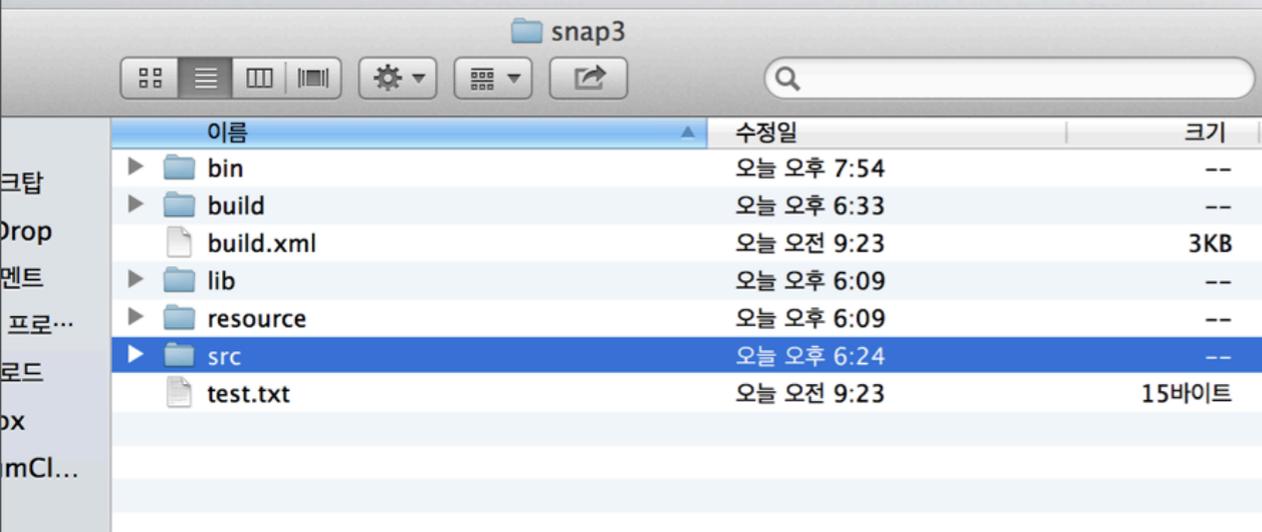
/ sonar-runner -h 실행시 다음과 같이 뜨면 성공

```
waniui-Mac-mini:bin wani$ sonar-runner -h  
INFO:  
INFO: usage: sonar-runner [options]  
INFO:  
INFO: Options:  
INFO:  -D,--define <arg>      Define property  
INFO:  -e,--errors              Produce execution error messages  
INFO:  -h,--help               Display help information
```

Sonar

사용법

sonar-project.properties 파일에
다음과 같이 작성



```
2 sonar.projectKey=my:project
3 sonar.projectName=My project
4 sonar.projectVersion=1.0
5
6 # optional description
7 #sonar.projectDescription=Fake description
8
9 # path to source directories (required)
10 sonar.sources=src
11
12 # path to test source directories (optional)
13 #sonar.tests=testDir1,testDir2
14
15 # path to project binaries (optional), for
16 #sonar.binaries=bin
17
18 # optional comma-separated list of paths to
19 sonar.libraries=lib/*.jar
20
21 # The value of the property must be the key
22 sonar.language=java
23
24 # Additional parameters
25 #sonar.my.property=value
```

Sonar

| 사용법

/ sonar-runner 실행하면

성공적으로 sonar 안에 프로젝트가 추가됨

Sonar

me.wani.kr:9000

Dashboards Projects Measures Reviews Settings Log in Search

Home

TOOLS
Dependencies
Compare

sonar

Welcome to Sonar Dashboard
Since you are able to read this, it means that you have successfully started your Sonar server. Well done!
If you have not removed this text, it also means that you have not yet played much with Sonar. So here are a few pointers for your next step:

- » Do you now want to [run analysis](#) on a project?
- » Maybe start [customizing dashboards](#)?
- » Or simply browse the [complete documentation](#)?
- » If you have a question or an issue, please report it on the [mailing list](#).

Projects

Name	Version	LOCs	RCI	Date
My project	1.0	1,704	47.2%	18:38

1 results

Projects

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

My project

Sonar

Version 1.0 - 2013/06/03 18:38:51

Time changes...

Lines of code

1,704

2,116 lines
761 statements
24 files

Classes

23

4 packages
143 methods
4 accessors

Violations

425

Rules compliance

47.2%

 <u>Blocker</u>	0	
 <u>Critical</u>	17	<div style="width: 10%;"></div>
 <u>Major</u>	212	<div style="width: 50%;"></div>
 <u>Minor</u>	178	<div style="width: 40%;"></div>
 <u>Info</u>	18	<div style="width: 5%;"></div>

Comments

4.1%

73 lines
0.0% docu. API
139 undocu. API

Duplications

18.1%

382 lines
22 blocks
8 files

Package tangle index

0.0%

> 0 cycles

Dependencies to cut

0 between packages
0 between files

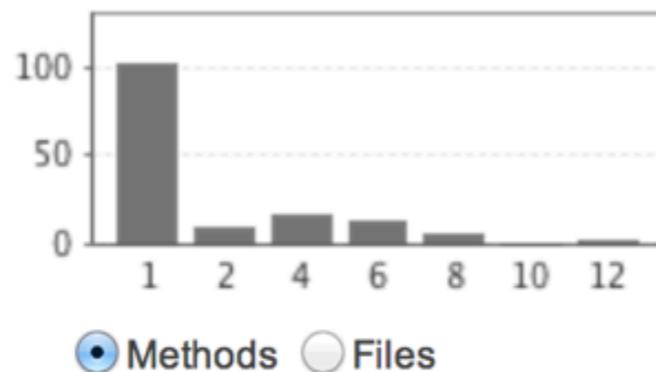
Complexity

2.3 /method

14.0 /class

13.4 /file

Total: 322



Unit tests coverage

-

Unit test success

0 tests

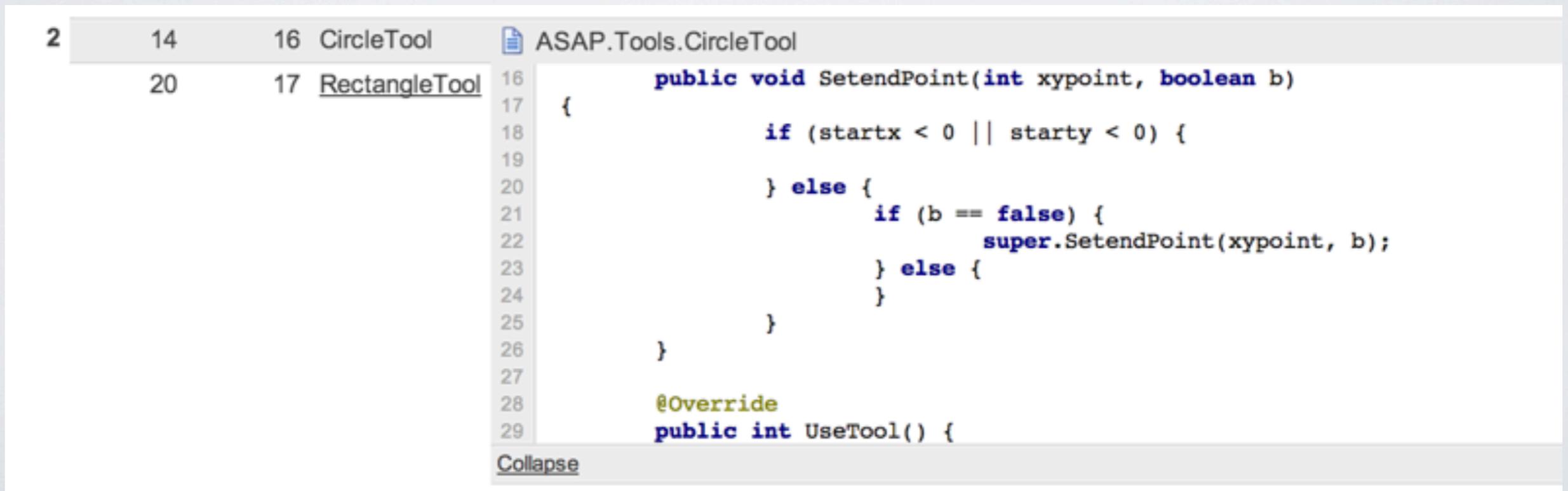
Events

All

Sonar

| Duplications (코드 중복)

/ 전체 2116Line 중 382Line 중복, 18.1%



```
2 14 16 CircleTool ASAP.Tools.CircleTool
20 17 RectangleTool 16 public void SetEndPoint(int xypoint, boolean b)
17 {
18     if (startx < 0 || starty < 0) {
19
20     } else {
21         if (b == false) {
22             super.SetEndPoint(xypoint, b);
23         } else {
24         }
25     }
26 }
27
28 @Override
29 public int UseTool() {
```

Collapse

Sonar

! Duplications (코드 중복)

/ 비슷한 기능들 중에 중복이 많이 발생

/ 하나의 도구에 하나의 클래스를 사용하기 보단 기능단위로 나누는 과정이 필요

2	14	16	CircleTool
	20	17	<u>RectangleTool</u>
4	18	41	CircleTool
	18	70	<u>CircleTool</u>
	17	97	<u>CircleTool</u>
	17	125	<u>CircleTool</u>

3	35	24	BrushTool
	35	31	<u>PenTool</u>
	35	46	<u>EraseTool</u>
2	21	62	BrushTool
	21	23	<u>EraseTool</u>
2	20	85	BrushTool
	20	69	<u>PenTool</u>
3	10	85	BrushTool
	10	69	<u>PenTool</u>
	10	84	<u>EraseTool</u>

Sonar

I Duplications (코드 중복)

Package	Class	Line(Nb Lines)	Other
ASAP.Tools	CircleTool	16(+14)	RectangleTool, 17
		41(+18)	CircleTool 70,97,125
	BrushTool	24(+35)	PenTool 31 Erase Tool 46
		62(+21)	EraseTool 23
		85(+20)	PenTool 69
		85(+10)	PenTool 69 EraseTool 84

Sonar

I Duplications (코드 중복)

Package	Class	Line(Nb Lines)	Other
ASAP.Tools	EraseTool	23(+21)	BrushTool 62
		43(+51)	PenTool 28
		46(+35)	BrushTool 24 PenTool 31
		84(+10)	PenTool 69 BrushTool 85
	PenTool	28(+51)	EraseTool 43
		31(+35)	BrushTool 24 EraseTool 46
		69(+20)	PenTool 69 BrushTool 85
		69(+10)	EraseTool 84 BrushTool 85

Sonar

| Duplications (코드 중복)

Package	Class	Line(Nb Lines)	Other
ASAP.Tools	MoveTool	13(+22)	SelectTool 13
	SelectTool	13(+22)	MoveTool 13
	RectagnleTool	16(+20)	CircleTool 16
ASAP.Canvas	MYCanvas	206(+13)	MYCanvas 243

Sonar

| Severity

^ Violations 425개 중, Critical 17, Major 212, 즉 47.2%

Sonar

| Severity - Critical

/' Empty If Stmt

/' If statement에서 공백인 부분 총 15개 발견

Package	Class	Line
ASAP.Tools	CircleTool	18, 23
	MoveTool	20, 23
	PipetteTool	22, 26
	RectangleTool	20, 30
	SelectTool	20, 23
	BrushTool	42
	EraseTool	64
	PenTool	49
ASAP.Canvas	MYCanvas	132, 136

Sonar

| Severity - Critical

/' Empty Finally Block

/' Finally 선언 후 공백인 부분 1군대 발견

Package	Class	Line
ASAP.Canvas	MYCanvas	114

Sonar

| Severity - Critical

/' Security - Array is stored directly

/' Array 전체를 직접 접근한 경우, 1군대 발견

Package	Class	Line
ASAP.Canvas	Image	264

Sonar

Severity - Major

- / 잠재적인 위험을 담고 있는 부분으로서 100% 수정하지 않아도 됨
- / Method Name, Package Name, Local Variable Name
- / 위 3가지의 경우 이름 규칙이 잘못된 경우 발생하는데 이부분은 제외하기로 함.(각 97개, 23개, 17개 발견됨)

Sonar

Severity - Major

Parameter Assignment

method에서 선언한 변수에 직접 접근 한 경우, 총 20개 발견

Package	Class	Line
ASAP.Tools	BrushTool	93, 94, 102, 103
	EraseTool	92, 93, 101, 102
	PenTool	77, 78, 86, 87
ASAP.Canvas	MYCanvas	129, 130, 147, 148, 331, 332, 404, 405

Sonar

Severity - Major

Simplify Boolean Expression

if (a == true)같이 사용한 경우, 총 10개 발견

Package	Class	Line
ASAP.Tools	MoveTool	19, 36
	BrushTool	17
	CircleTool	21
	EraseTool	19
	PenTool	34
	PipetteTool	21
	RectangleTool	25
	SelectTool	19
ASAP.Canvas	MYCanvas	268

Sonar

| Severity - Major

/' If Else Stmts Must Use Braces

/' {}로 묶이지 않는 조건절인 경우, 총 9개 발견

Package	Class	Line
ASAP.Tools	FileManager	8
ASAP.Canvas	SelectedImage	67

Sonar

Severity - Major

Visibility Modifier

가급적 protected보단 private를 사용, 총 8개 발견

Package	Class	Line
ASAP.Tools	VirtualTool	7,9,10,12,13,15,16
ASAP.Canvas	VirtuallImage	6

Sonar

| Severity - Major

/' Final Class

/' 더이상 상속되지 않는 클래스의 경우 Final 선언, 총 6개 발견

Package	Class	Line
ASAP.Tools	CircleTool	19, 36
	EraseTool	17
	MoveTool	21
	PenTool	19
	PipetteTool	34
	RectangleTool	21

Sonar

| Severity - Major

/' Unused local variable

/' 사용되지 않는 변수, 총 4개 발견

Package	Class	Line
ASAP.Canvas	MyCanvas	72,73,74,76

Sonar

| Severity - Major

/' Avoid Duplicate Literals

/' 가급적 중복된 String값은 변수로 사용하도록, 총 3개 발견

Package	Class	Line
ASAP	FileManager	53, 55, 57
		jpg, gif, png가 각 4번씩 더 등장

Sonar

| Severity - Major

/' Avoid Print Stack Trace(use a logger call instead)

/' printStackTrace를 가급적 사용 자제, 총 3개 발견

Package	Class	Line
ASAP	FileManager	69, 97
		33

Sonar

| Severity - Major

/ Avoid Commented-out lines of code

/ 코드라인에 주석을 처리한 경우, 총 3개 발견

Package	Class	Line
ASAP.Canvas	Image	121
	MYCanvas	118
	SelectedImage	71

Sonar

| Severity - Major

/ Unnecessary Local Before Return

/ return하는 변수가 바로 이전에 선언된 경우 불필요함, 총 2개 발견

Package	Class	Line
ASAP.Canvas	MYCanvas	159, 165

Sonar

| Severity - Major

/' If Stmt's Must Use Braces

/' if 구문에 {}이 사용되지 않은 경우, 총 2개 발견

Package	Class	Line
ASAP.Canvas	MYCanvas	268
	SelectedImage	34

Sonar

Severity - Major

Cyclomatic Complexity

최대 10으로 허용되었는데 13임, 1군대 발견.

Package	Class	Line
ASAP.Tools	CircleTool	28 (아래에 있는 함수)

Sonar

Severity - Major

Ncss Method Count

NCSS 84줄이 나왔음, 1군대 발견.

Package	Class	Line
ASAP.Tools	CircleTool	29 (아래로 쪽)

Sonar

| Severity - Major

/' Avoiding Catching NPE

/' , 1군대 발견.

Package	Class	Line
ASAP.Canvas	MYCanvas	109

Sonar

| Severity - Major

/' Simplify Boolean Return

/' Boolean을 리턴할 경우 if stmt를 없앨수 있음, 1군대 발견.

Package	Class	Line
ASAP.Canvas	MYCanvas	360

Sonar

| Severity - Major

/' Constructor Calls Overridable Method

/' , 1군대 발견.

Package	Class	Line
ASAP.Canvas	Image	19

Sonar

| Complexity

/ Complexity를 Method, Class, File 각각에 대한 평균값을 제공

/ Complexity/Class = Complexity/File 이나, 다른경우도 있음

Sonar

Complexity / Method

Package	Class	Complexity / Method
ASAP.Tools (2.7)	CircleTool	4.8
	BrushTool	4.2
	EraseTool	4.2
	PenTool	3.6
	MoveTool	3.3
	RectangleTool	3.3
	PipetteTool	1.8
	SelectTool	1.8
	VirtualTool	1.1
	BrushToolTest	1.0
	CircleToolTest	1.0
	EraseToolTest	1.0
	PipetteToolTest	1.0

Sonar

Complexity / Method

Package	Class	Complexity / Method
ASAP (2.5)	FileManager	6.0
	GUIMain	1.5
	MousePointer	1.0

Sonar

Complexity / Method

Package	Class	Complexity / Method
ASAP.Canvas (2.0)	Image	2.6
	SelectedImage	2.4
	MYCanvas	2.3
	ImageTransterable	1.8
	VirtuallImage	1.6
	MYCanvasTest	1.0

Sonar

Complexity / Method

Package	Class	Complexity / Method
ASAP.test (2.5)	ASAPTEST1	1.2

Sonar

Complexity / class

Package	Class	Complexity / Method
ASAP.Canvas (25.5)	MYCanvas	64.0
	Image	31.0
	SelectedImage	24.0
	MYCanvasTest	14.0
	VirtuallImage	13.0
	ImageTransterable	7.0

Sonar

Complexity / class

Package	Class	Complexity / Method
ASAP.Tools (10.5)	EraseTool	21.0
	BrushTool	21.0
	CircleTool	19.0
	PenTool	18.0
	MoveTool	13.0
	RectangleTool	13.0
	VirtualTool	9.0
	PipetteTool	7.0
	SelectTool	7.0
	BrushToolTest	2.0
	CircleToolTest	2.0
	EraseToolTest	2.0
	PipetteToolTest	2.0

Sonar

Complexity / class

Package	Class	Complexity / Method
ASAP (9.0)	FileManager	18.0
	MousePointer	6.0
	GUIMain	3.0

Sonar

Complexity / class

Package	Class	Complexity / Method
ASAP.test (6.0)	ASAPTESTI	6.0

Sonar

Complexity / file

Complexity / file 나머진 같고, ASAP 패키지만 다름

Package	Class	Complexity / Method
ASAP (6.8)	FileManager	18.0
	MousePointer	6.0
	GUIMain	3.0
	Window	0.0

Sonar

Complexity

ASAP.Canvas

- /class 는 높은 복잡도를 가지지만, /method는 낮은 복잡도를 가짐
- 상당히 많은 method가 구현되어 있음, 다양한 클래스로 세분화 하여야 함

ASAP.Tools

- /method 가 높은 복잡도를 가짐
- class에 비해 너무 거대한 method를 다루고 있거나, class에 비해 method의 갯수가 너무 적음. 메서드 자체를 나누어서 저장을 하거나, 클래스를 묶어야 함

ASAP

- 평균값은 적당하나, /method에서 FileManager가 혼자서 너무 큰 Complexity를 갖고 있음
- FileManager를 좀더 분리해서 적절히 나눠갖게 하던가, 전체 패키지에 골고루 기능을 분배해야 하는 필요성이 있음

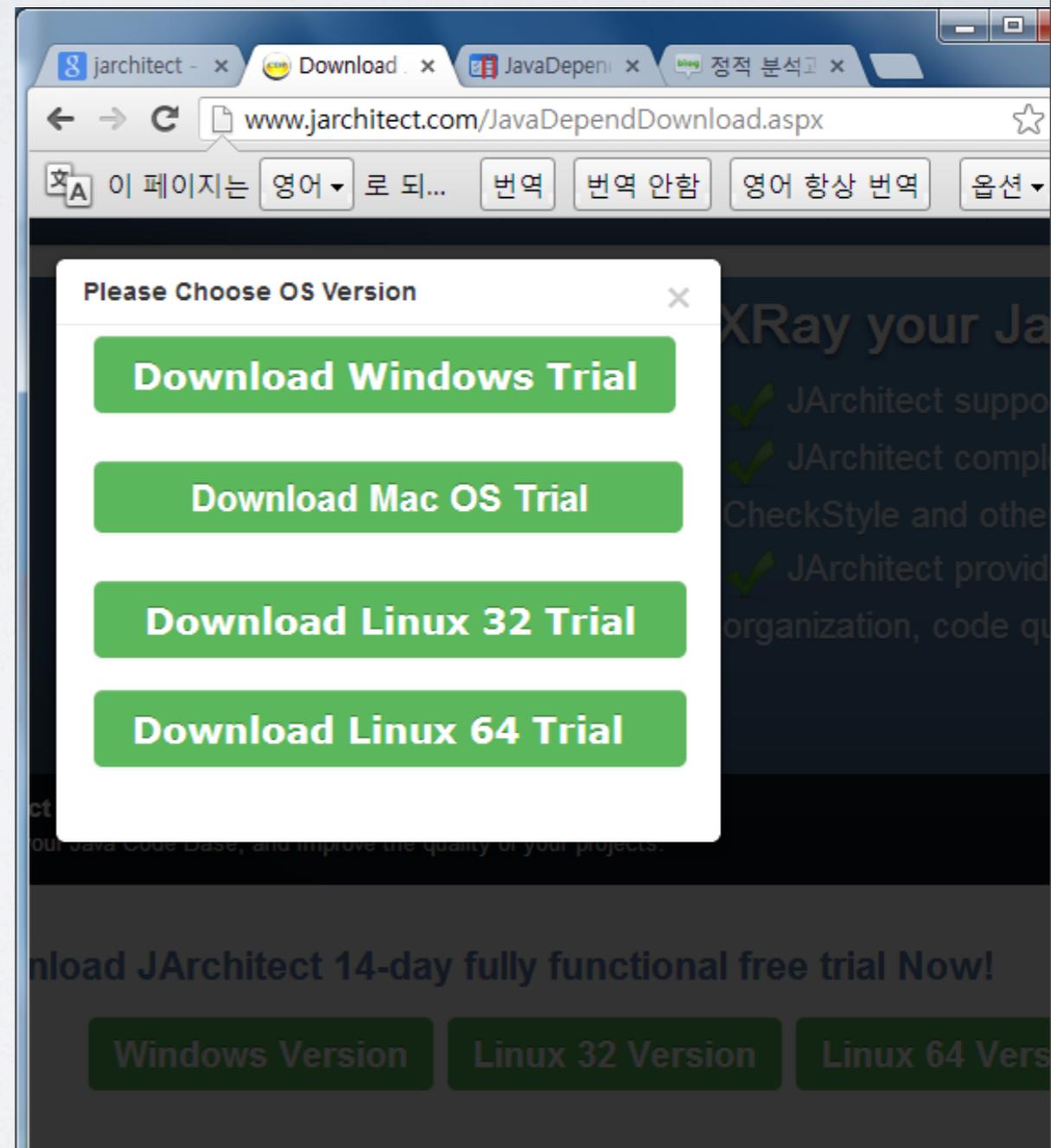
JArchitect

JArchitect

| 설치

/ <http://www.jarchitect.com/JavaDependDownload.aspx>

/ 다운로드 후, 파일 실행

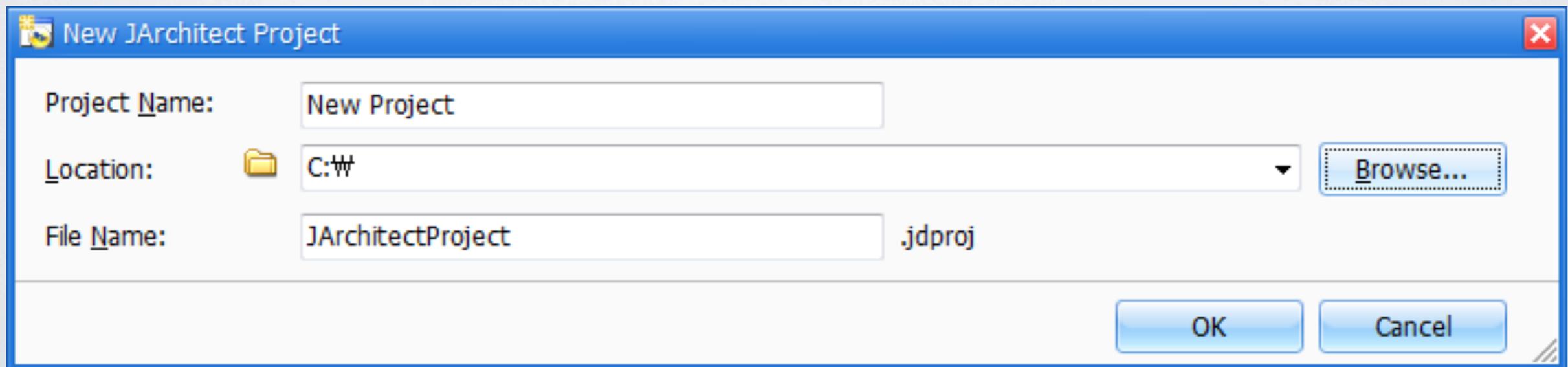


JArchitect

| 설치

^ File -> New Project

^ Browse, 내 프로젝트를 올림

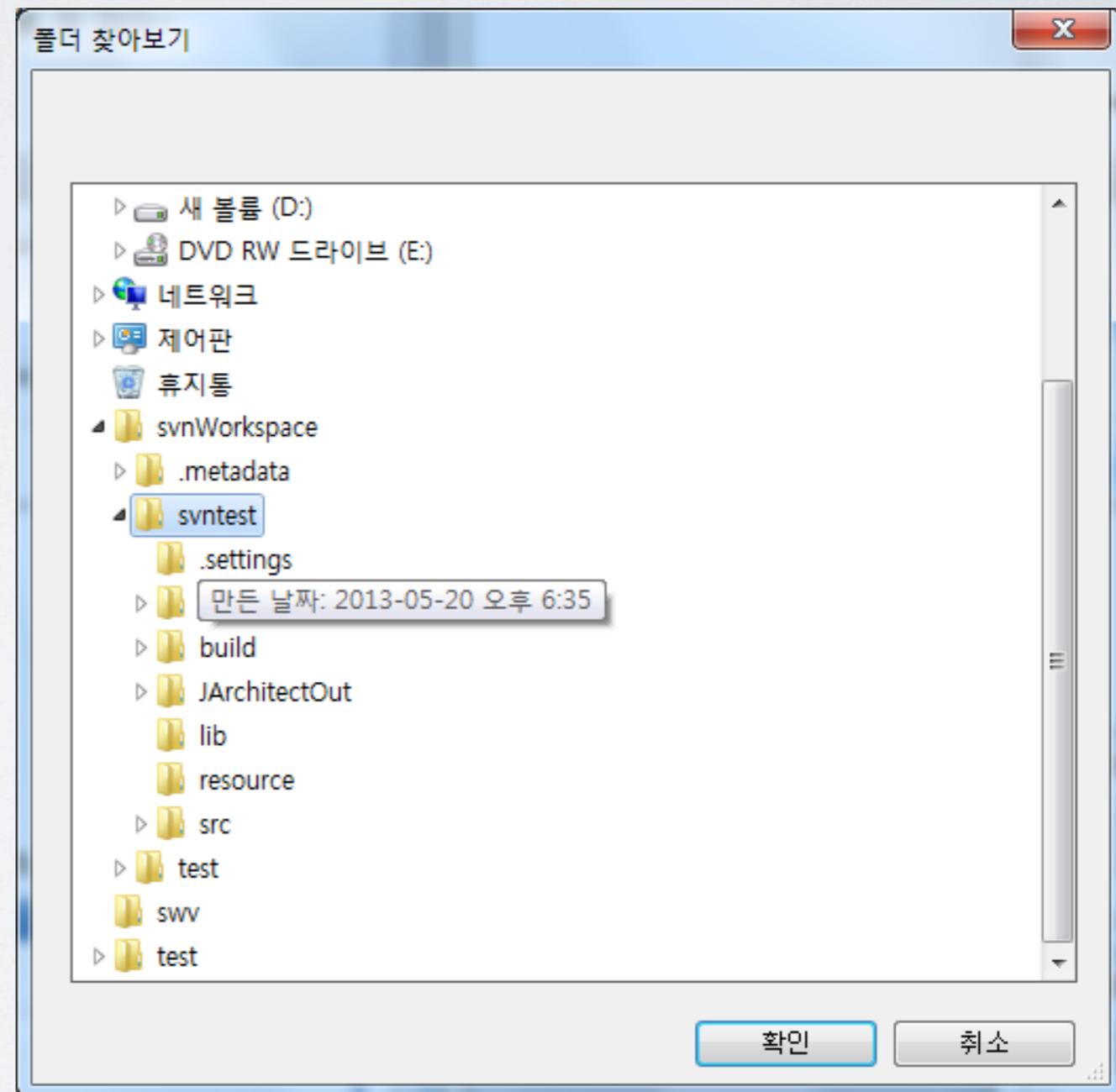


JArchitect

| 설치

/ File -> New Project

/ Browse, 내 프로젝트를 올림

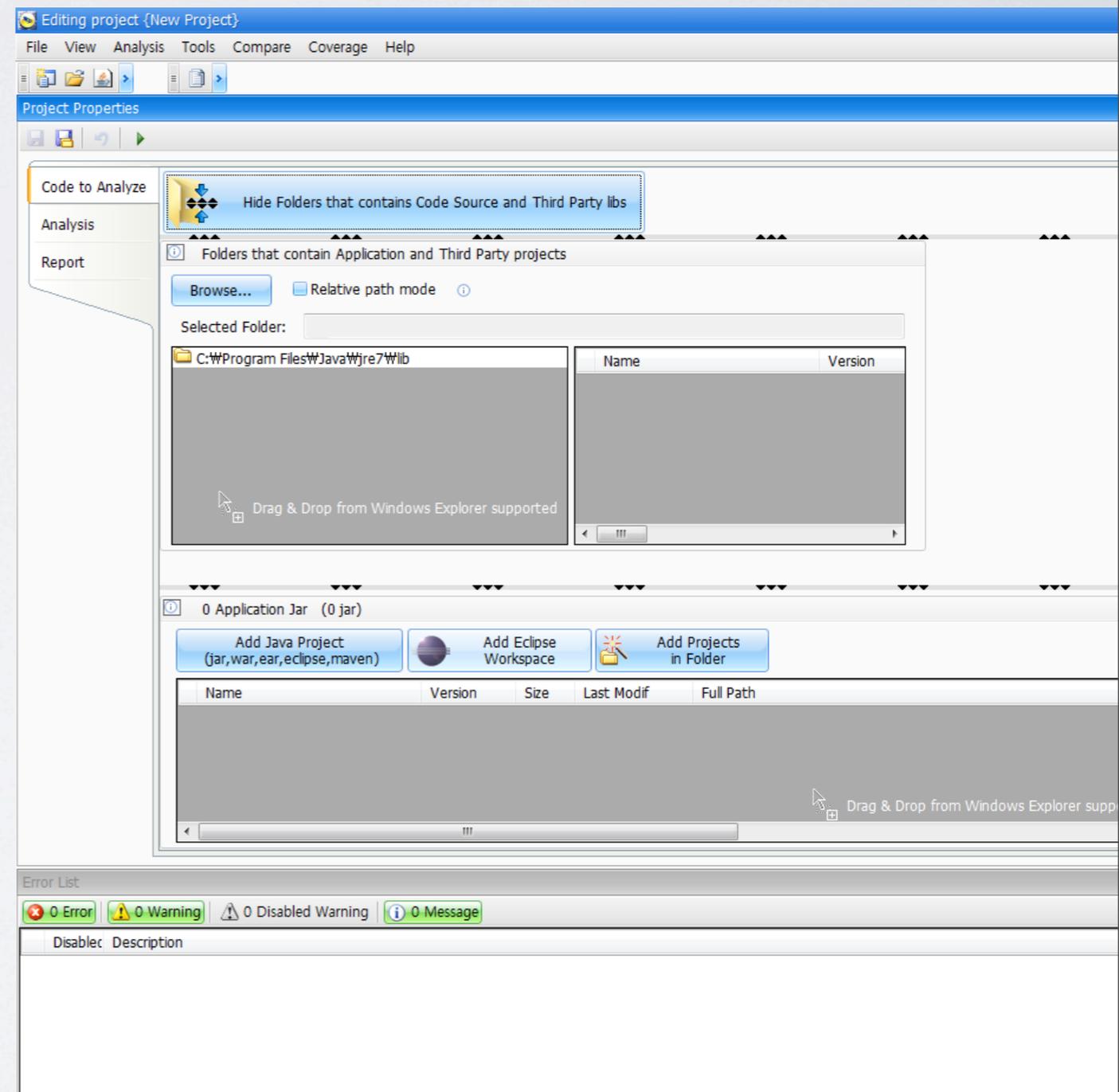


JArchitect

| 설치

/ File -> New Project

/ Browse, 내 프로젝트를 올림

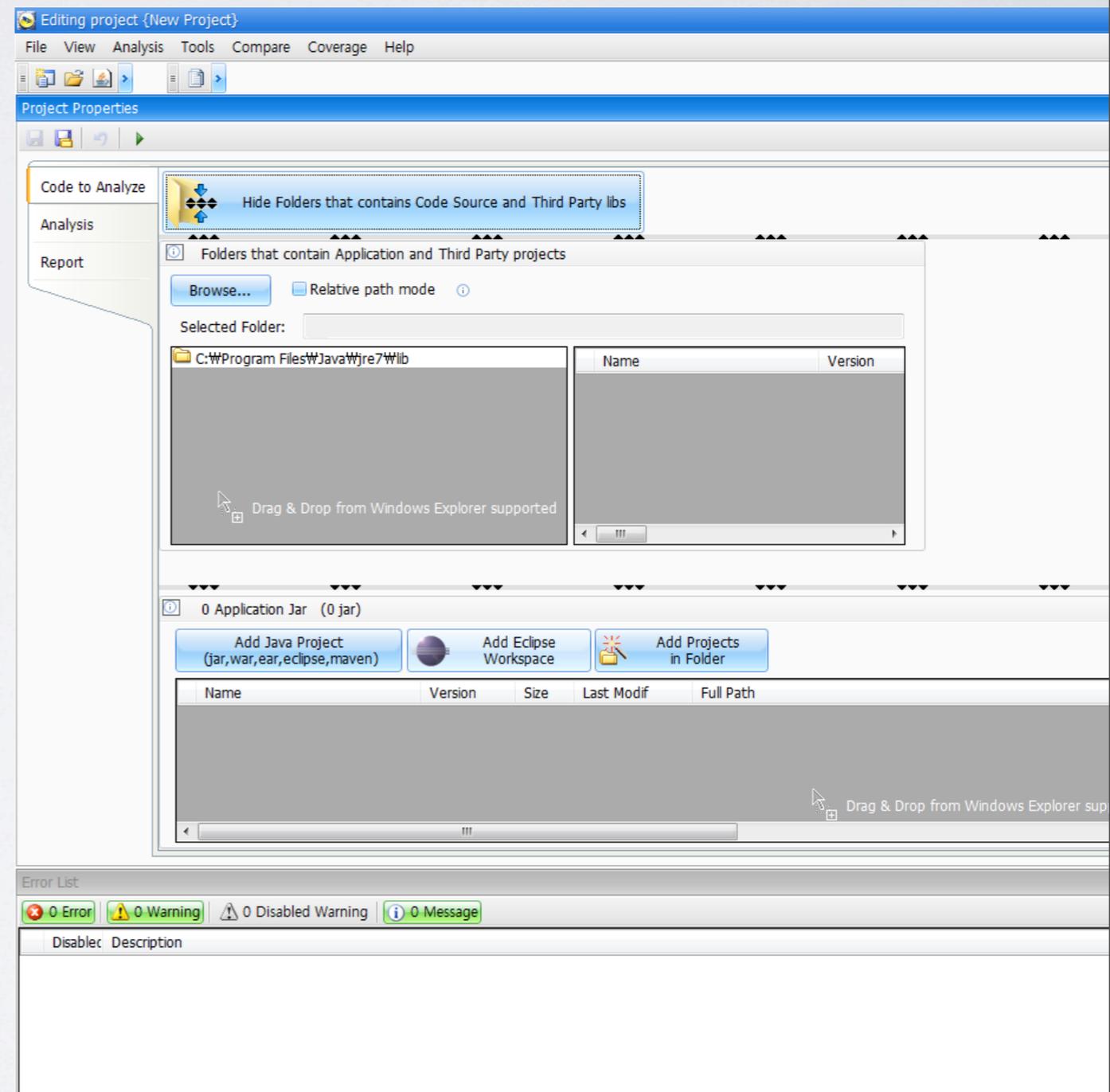


JArchitect

| 설치

/ Add Eclipse Workspace

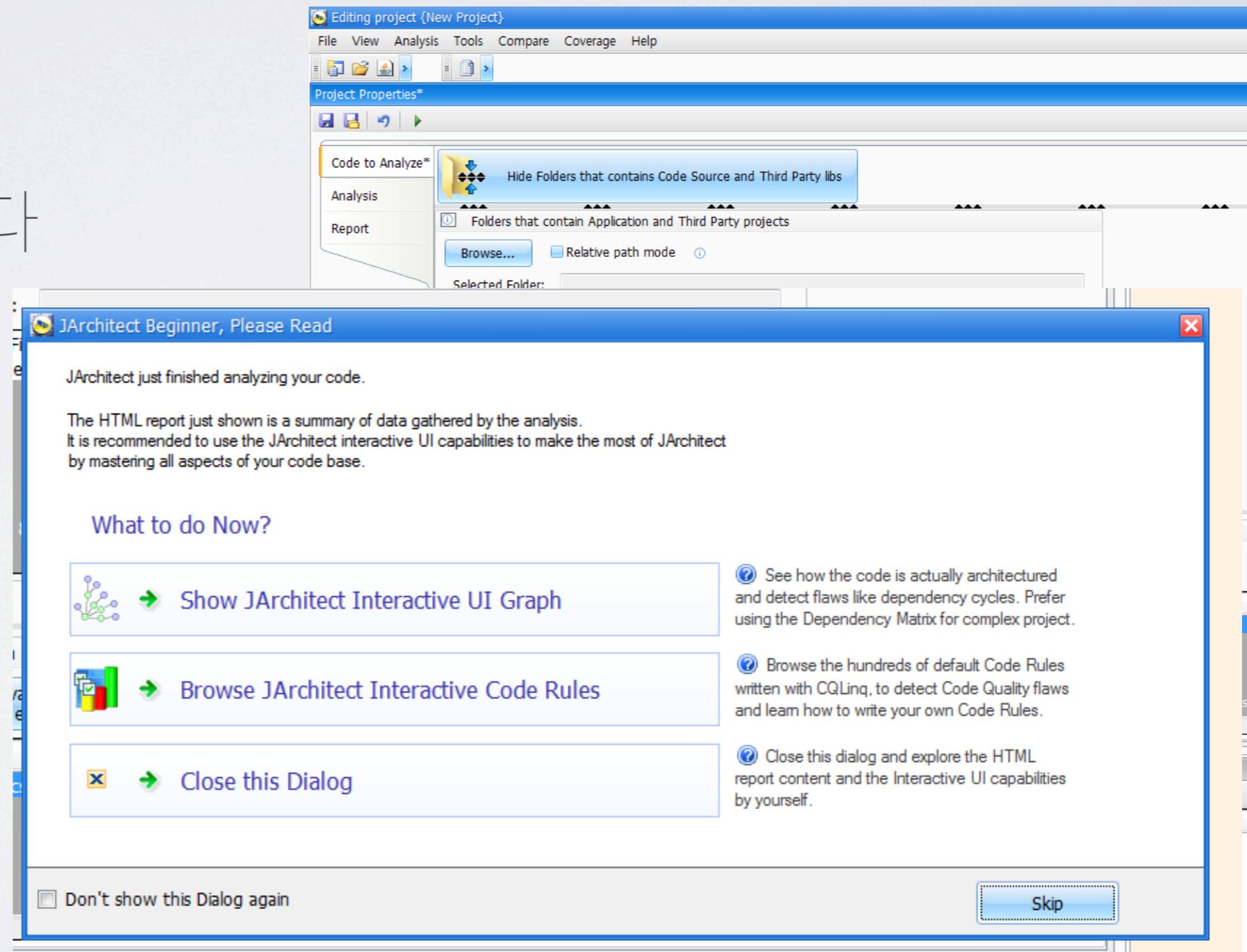
/ 내 이클립스 workspace 선택



JArchitect

| 설치

/ 그리고 F5를 누른다



JArchitect

The screenshot displays the JArchitect application interface. The main window is titled "New Project (analysis done Mon 03 Jun 10:13 most recent)". The interface is divided into several panes:

- Class Browser:** Shows a tree view of the project structure with nodes for "New Project", "200711007", "20130409", "XML_DOM", "rt", and "xercesImpl".
- Dependency Graph:** A central pane showing a dependency graph with 5 nodes and 2 edges. The nodes are "XML_DOM", "xercesImpl", "200711007", "20130409", and "rt". Edges connect "XML_DOM" to "xercesImpl" and "XML_DOM" to "rt".
- Info:** A pane on the right displaying a "Global Summary" for the project. It includes project details and application statistics.
- Queries and Rules Explorer:** A pane at the bottom showing a list of queries and rules, such as "Code Quality", "Object Oriented Design", and "Dead Code".

The "Global Summary" pane contains the following information:

Project Name: New Project
Project File: C:\Users\cse502-01\workspace\XML_DOM\JArchitectProject\jproj
Analysis Date: Mon 03 Jun 10:13 most recent
Analyzed by: JArchitect v3.1.0.6435

Application New Project

- # ByteCode instructions : 1 018
- # lines of code (LOC) : 222
- # lines of comment : 17
- Percentage Comment : 7%
- # Projects : 3
- # Packages : 1
- # Types : 7
- # Methods : 33
- # Fields : 18

Coverage:

- Percentage Coverage : N/A because no coverage data specified
- # Lines of Code Covered : N/A because no coverage data specified
- # Lines of Code Not Covered : N/A because no coverage data specified

Third party code used by the application:

- # Third party Projects used : 2
- # Packages used : 7

The "Queries and Rules Explorer" pane shows a list of queries and rules with checkboxes for activation and a table of active items.

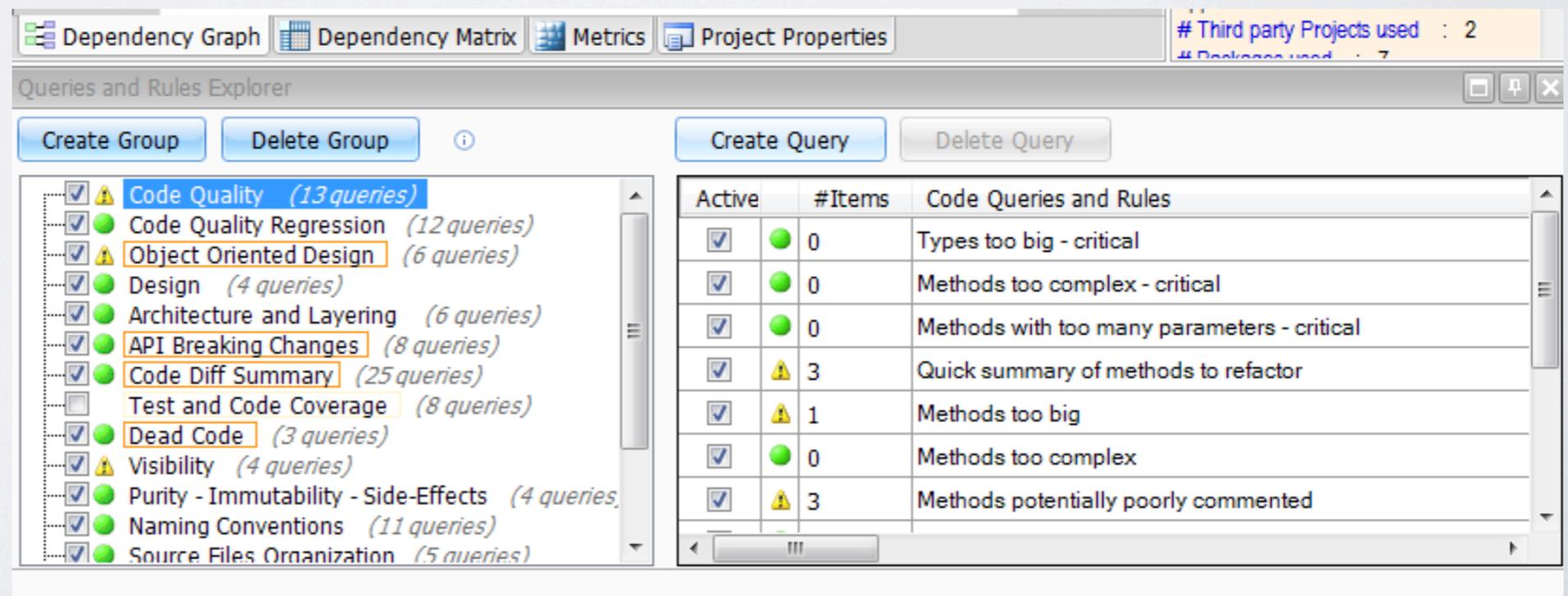
Active	#Items	Code Queries and Rules
<input checked="" type="checkbox"/>	0	Types too big - critical
<input checked="" type="checkbox"/>	0	Methods too complex - critical
<input checked="" type="checkbox"/>	0	Methods with too many parameters - critical
<input checked="" type="checkbox"/>	3	Quick summary of methods to refactor
<input checked="" type="checkbox"/>	1	Methods too big
<input checked="" type="checkbox"/>	0	Methods too complex
<input checked="" type="checkbox"/>	3	Methods potentially poorly commented

사용법

JArchitect

| 사용법

Dependency Graph, Dependency Metrics, Metrics, Project Property 각 탭이 결과를 제공



Queries and Rules Explorer

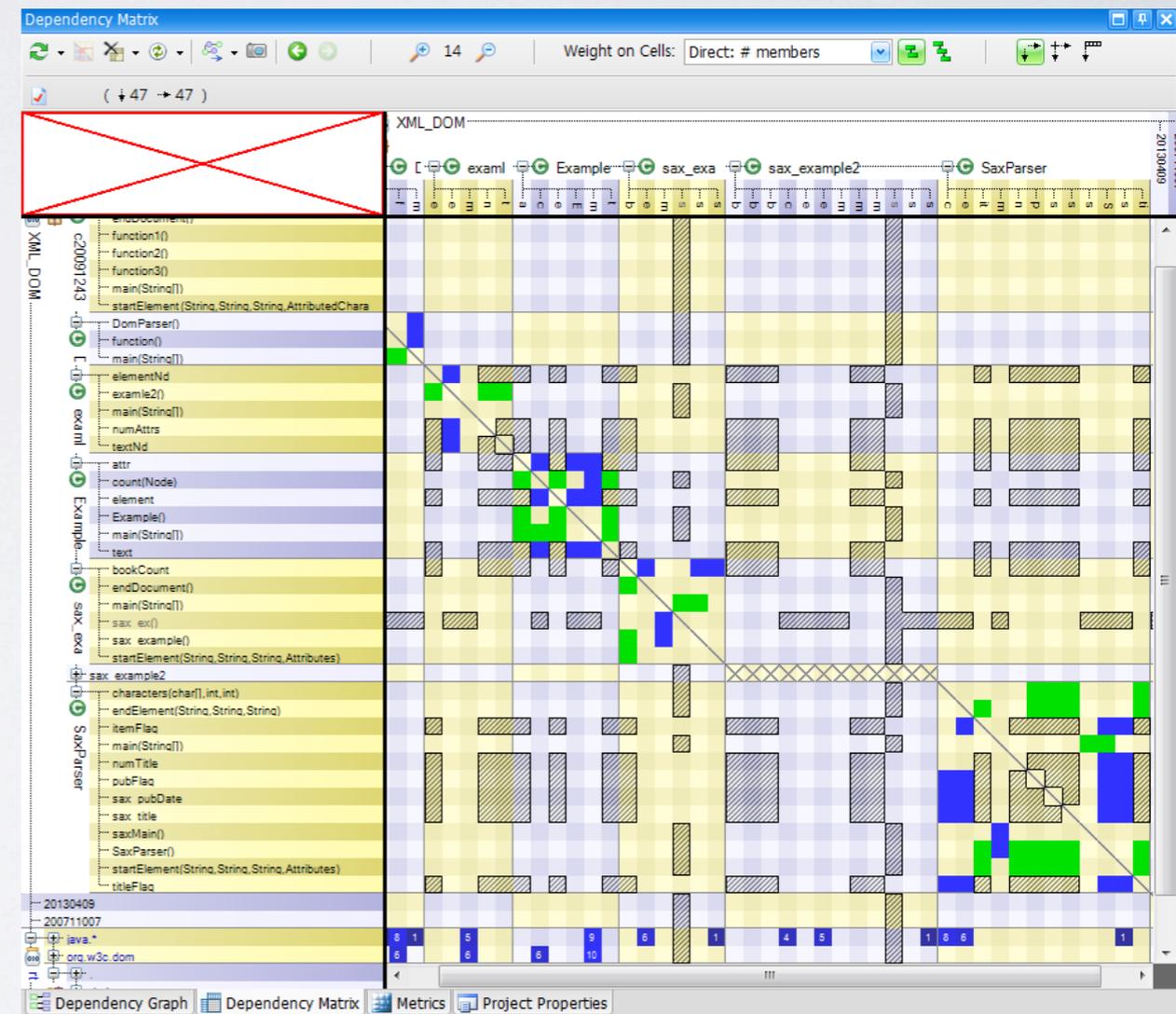
Active #Items Code Queries and Rules

<input checked="" type="checkbox"/>	0	Types too big - critical
<input checked="" type="checkbox"/>	0	Methods too complex - critical
<input checked="" type="checkbox"/>	0	Methods with too many parameters - critical
<input checked="" type="checkbox"/>	3	Quick summary of methods to refactor
<input checked="" type="checkbox"/>	1	Methods too big
<input checked="" type="checkbox"/>	0	Methods too complex
<input checked="" type="checkbox"/>	3	Methods potentially poorly commented

JArchitect

Dependency Matrix

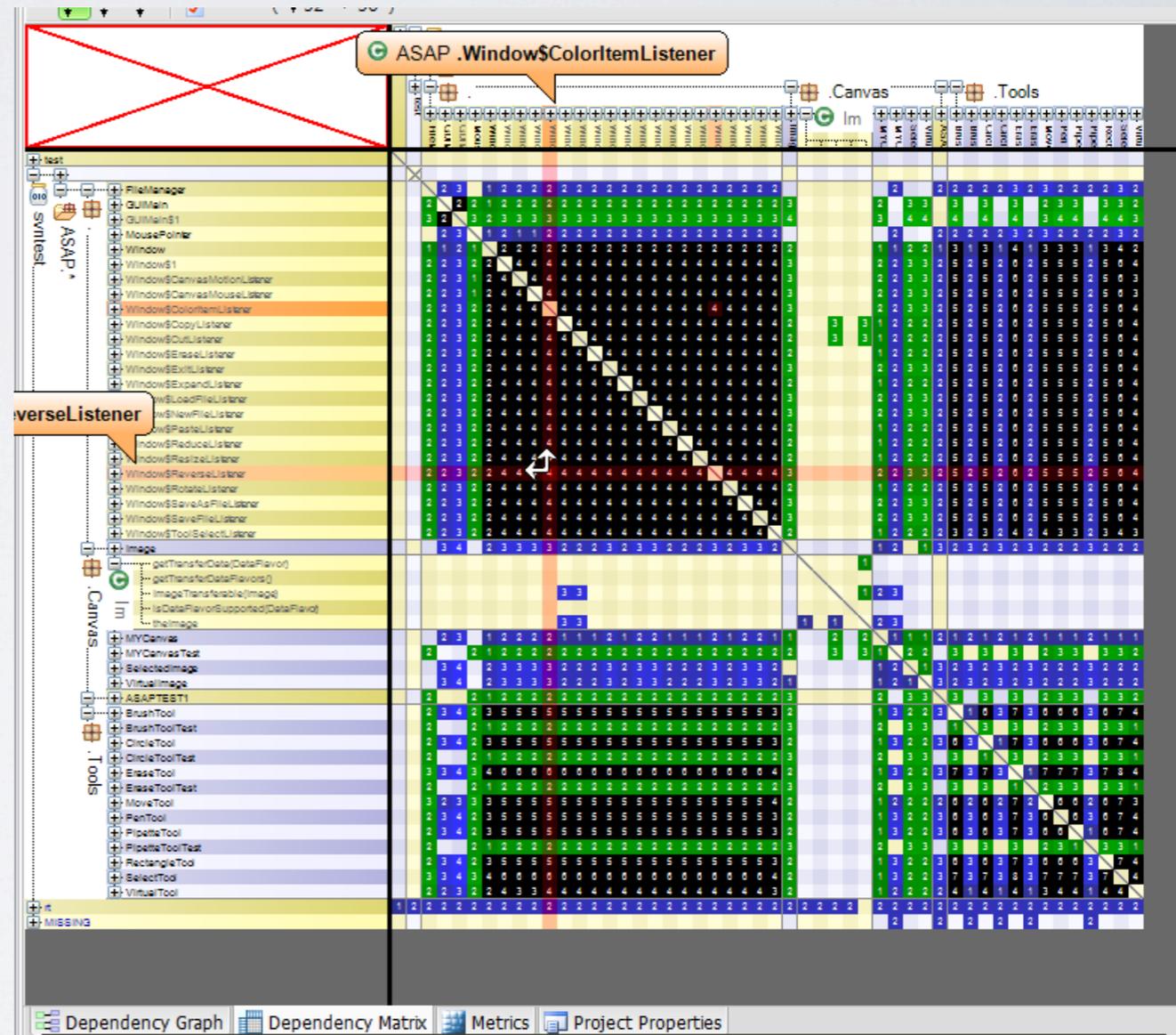
- / 이 매트릭스는 중앙의 45도 각도로 있는 검정색 실선을 중심으로 대칭이다
- / 초록색은 세로축의 메소드가 가로축의 메소드에게 사용된다는 뜻
- / 파란색은 세로축의 메소드가 가로축의 메소드를 사용한다는 뜻
- / 패키지, 클래스, 메소드의 단위를 설정 가능



JArchitect

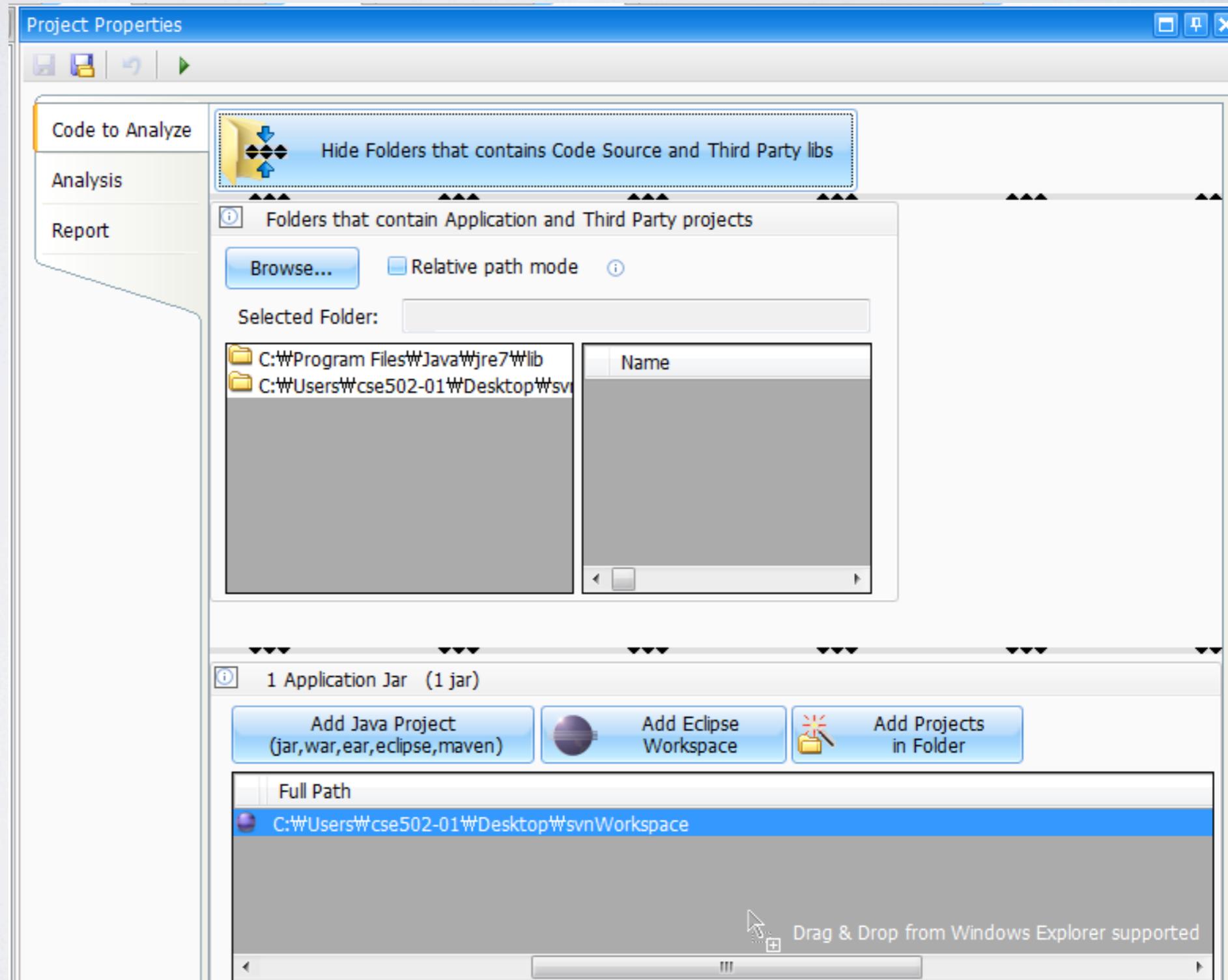
Dependency Matrix

검정색은 간접적으로 서로를 사용하고 있다는 뜻



JArchitect

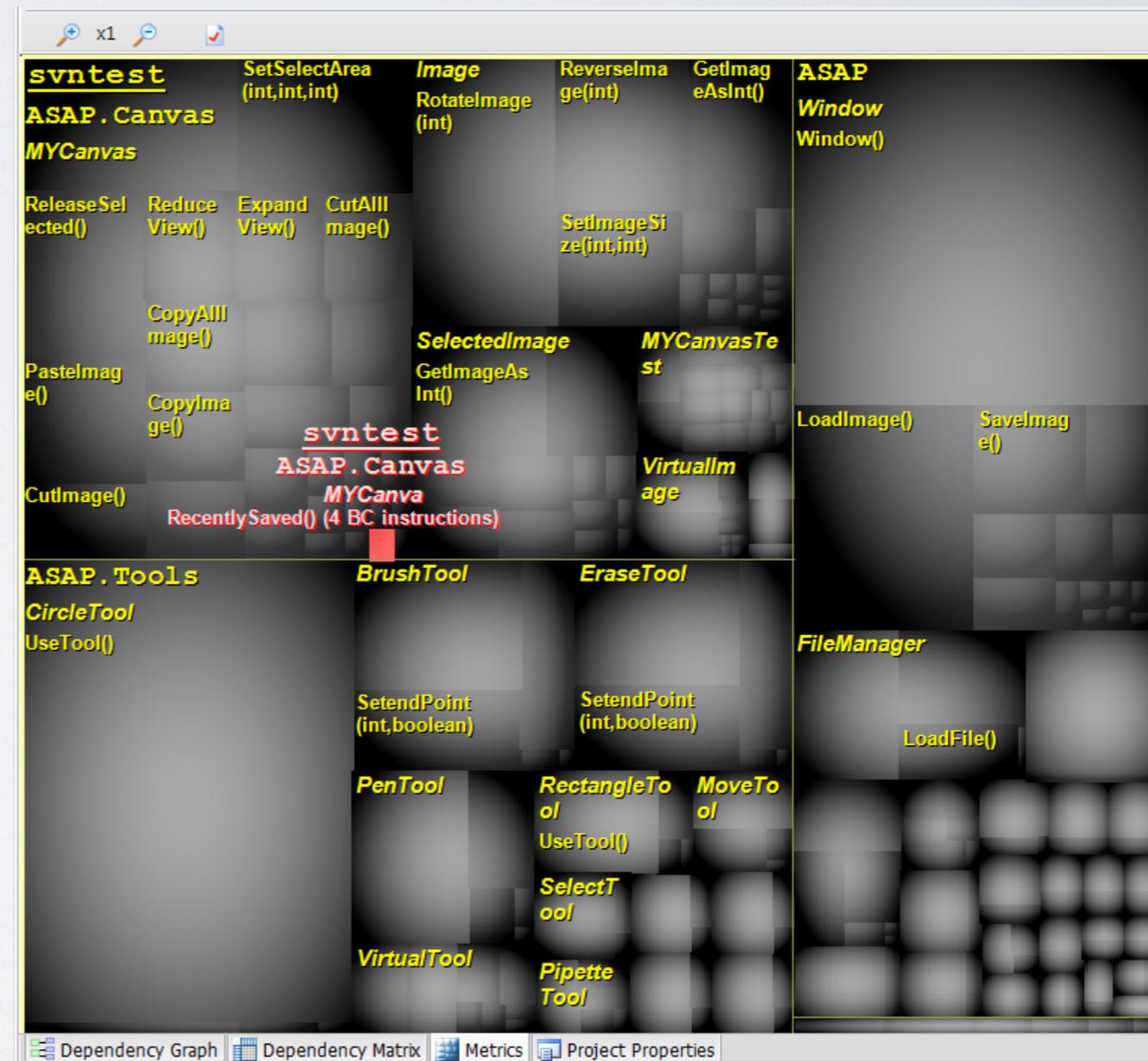
Project Property



JArchitect

Metrics

- / 트리맵의 모습은 코드의 계층구조를 의미
- / 큰 사각형은 클래스, 그 안의 사각형은 해당 클래스의 메서드
- / 마우스를 올리면 선명하게 볼 수 있고, 그 크기와 위치를 통해 클래스 내의 계층 구조를 알 수 있음
- / 영역이 크면 코드라인의 수가 많음
(Byte Code Instructions)



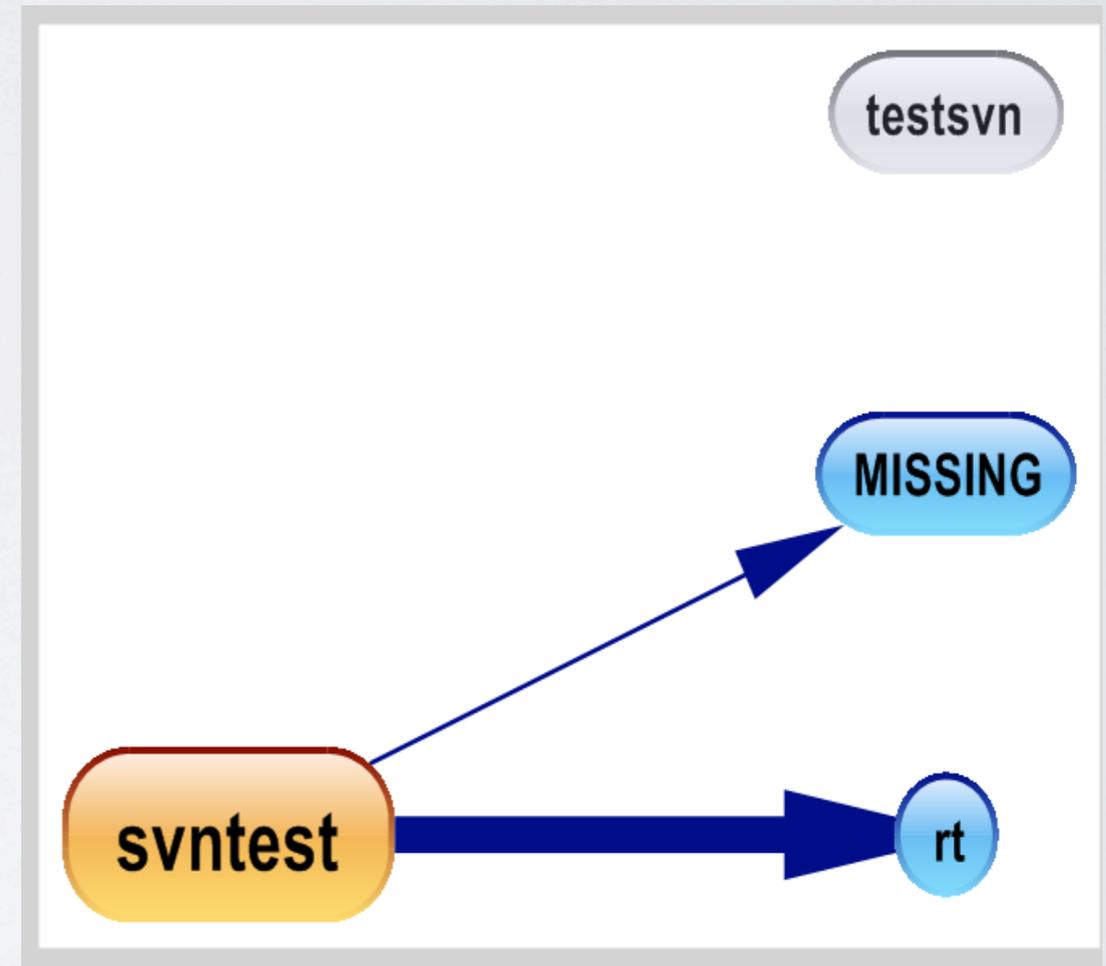
JArchitect

Dependency Graph

/ ASAP는 svntest(저장소)에 들어있으며 missing, rt를 사용 (의존한다)

/ rt의 82개의 메서드 사용

/ missing의 6개의 메서드 사용

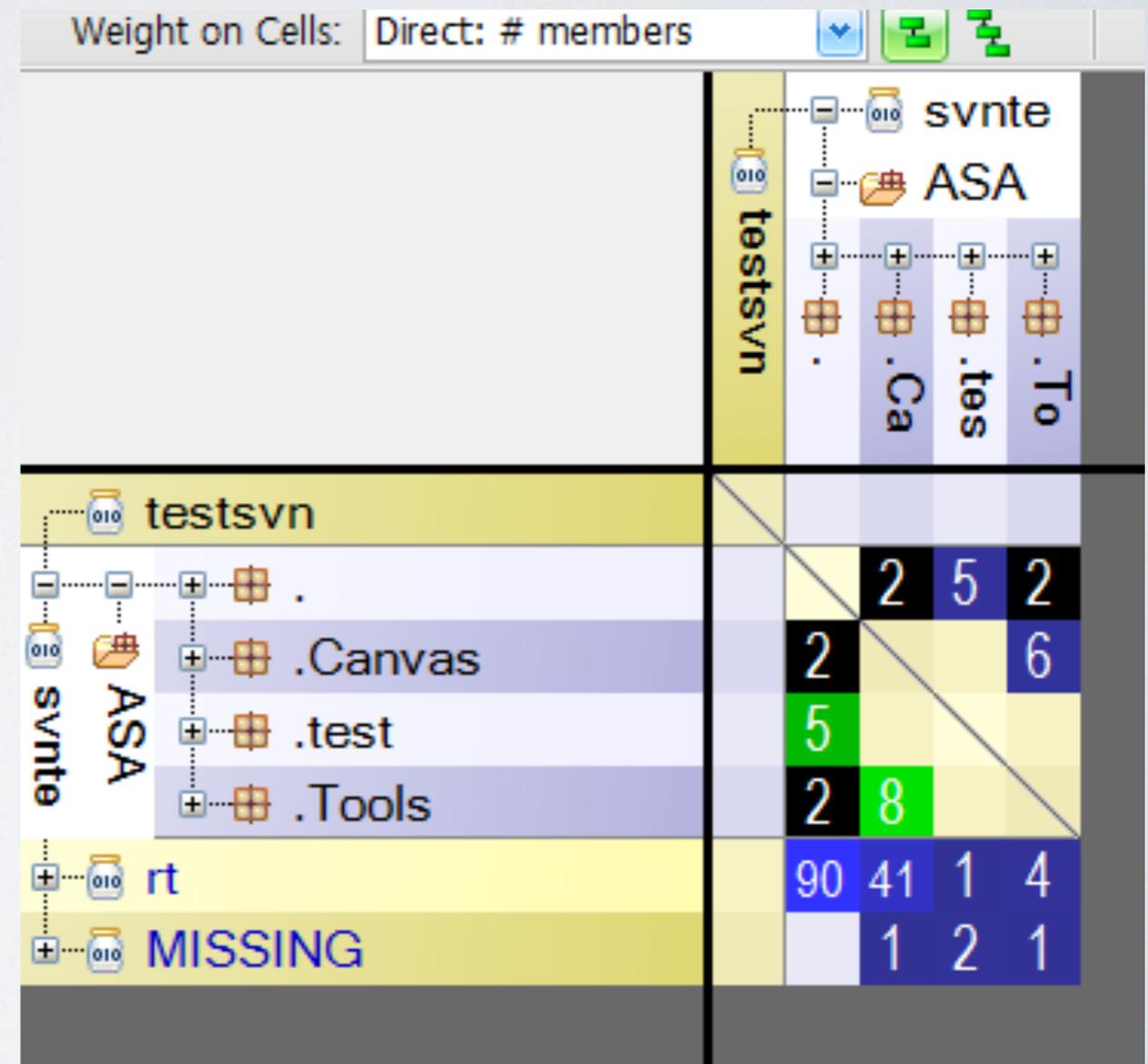


JArchitect

Dependency Matrix

ASAP.Canvas와 ASAP.Tools 클래스는 서로 직접적으로 사용하고 있으며 각각 서로 1번 이용, 즉 Dependency Cycle의 길이가 2

A패키지가 변하면 B패키지도 변하고, 그로 인해 다시 A패키지가 영향 받을 수 있어, 코드를 예측하지 않은 방향으로 변할 가능성이 있음

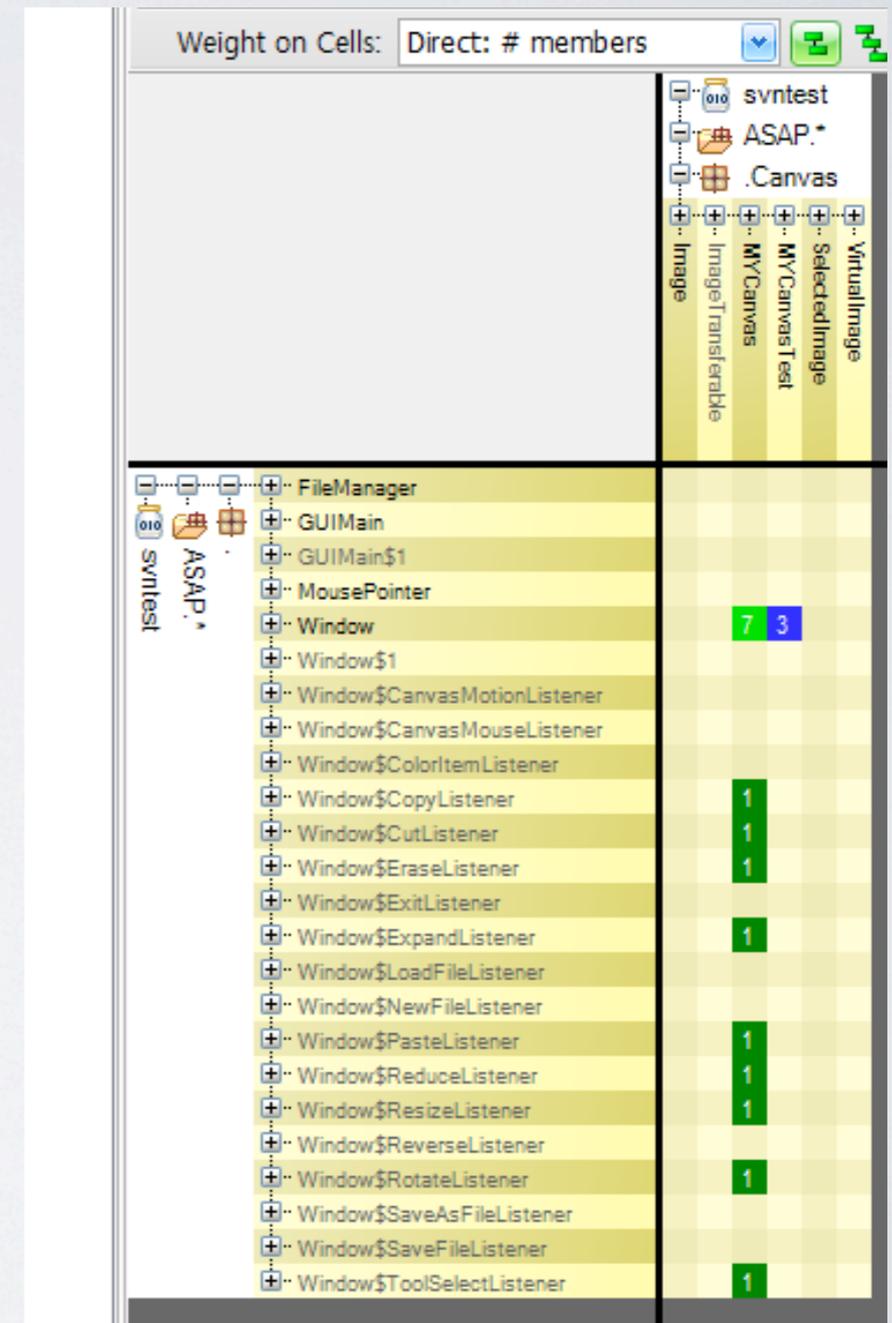


JArchitect

Dependency Matrix

ASAP x ASAP.Canvas

ASAP.Canvas.MYCanvas와 Window, ASAP.Canvas.MYCanvasTest와 Window에서 서로 사용하고 있어서 위험 할 수 있음



JArchitect

Dependency Matrix

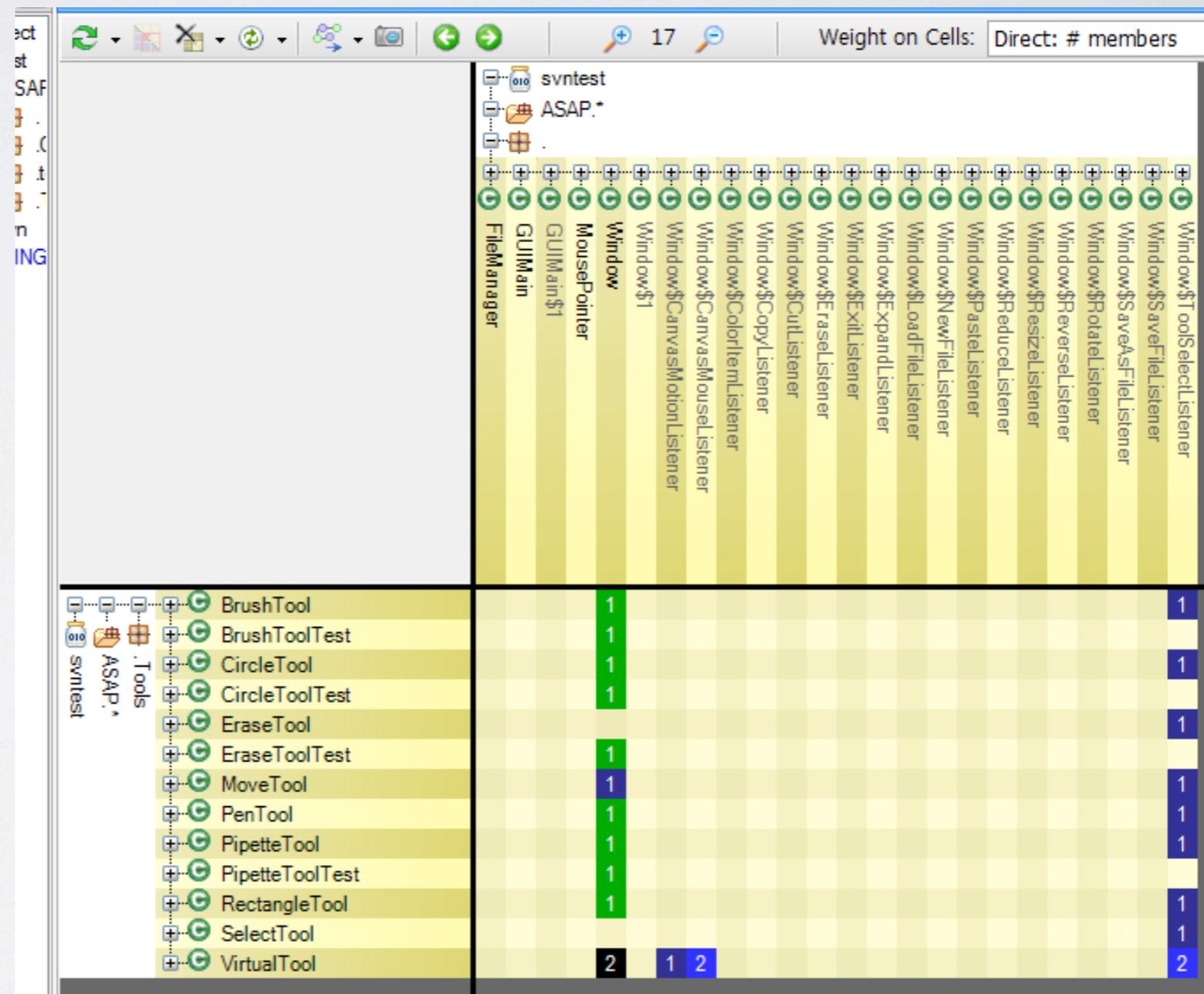
ASAP x ASAP.Tools

직접 사용에서는 크게
문제는 없으나,

ASAP.Tools.VirtualTool

과 Window가 문제가

생길 수 있음



JArchitect

Metrics (Code Quality)

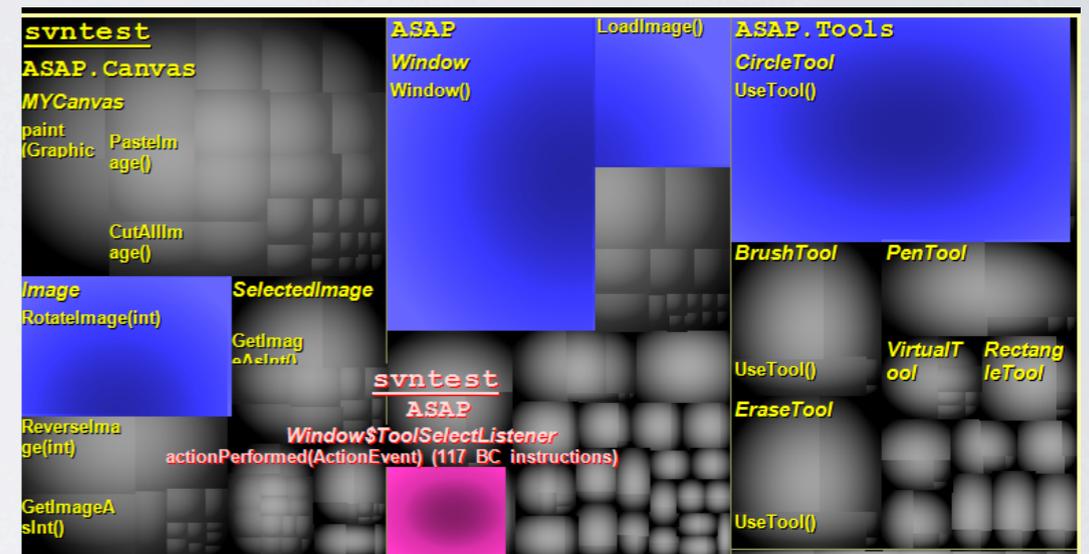
Code Queries and Rules	Items	비고
Methods too big	5	
Methods potentially poorly commented	6	
Methods with too many local variables	5	
Methods too complex	1	
Types too big	0	
Methods too complex - critical	0	
Methods with too many parameter	0	
Methods with too many field	0	
Methods with too many cohesion	0	

JArchitect

Metrics (Code Quality)

Method too big

너무 코드라인이 긴 메서드



Package	Class	Method
ASAP	Window	Window
ASAP	Window	LoadImage
ASAP.Canvas	Image	RotateImage
ASAP.Tools	CircleTool	UseTool

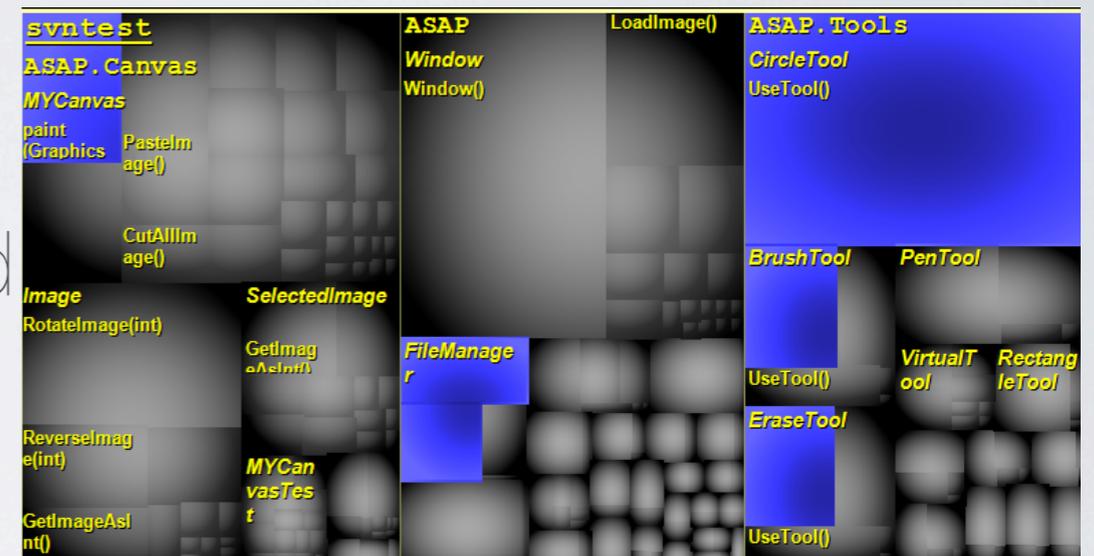
JArchitect

Metrics (Code Quality)

Methods potentially poorly commented

주석이 없는 메서드

전체적으로 주석 정리 잘 되어있음

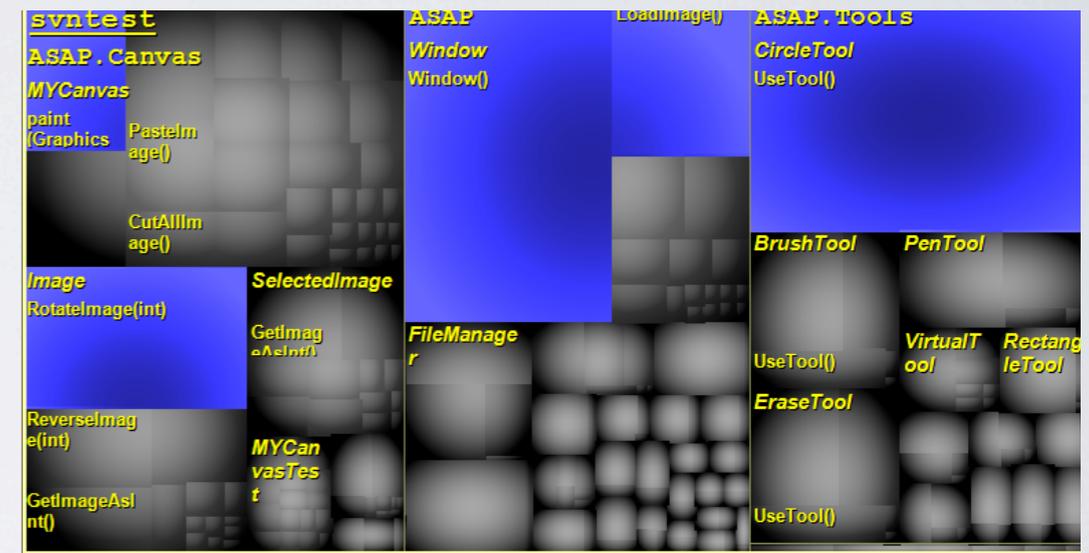


JArchitect

Metrics (Code Quality)

Methods with too many local variables

지역변수를 너무 많이 쓰는 메서드



Package	Class	Method
ASAP	Window	Window
ASAP	Window	LoadImage
ASAP.Canvas	MYCanvas	paint
ASAP.Canvas	Image	RotateImage
ASAP.Tools	CircleTool	UseTool

JArchitect

Metrics (Object Oriented Design)

전체적으로 객체지향률을 50개를 어겨 객체지향을 잘 따른 코드라 보긴 어렵다. 다만 한군대에서 침해한 경우가 많기 때문에 쉽게 수정 될 수 있다.

Code Queries and Rules	Items	비고
Class with no descendant should be final if possible	42	
Avoid the Singleton pattern	8	
Base class should not use derivatives	0	
Class shouldn't be too deep in inheritance tree	0	
Don't assign static fields from instance methods	0	
Avoid empty interfaces	0	

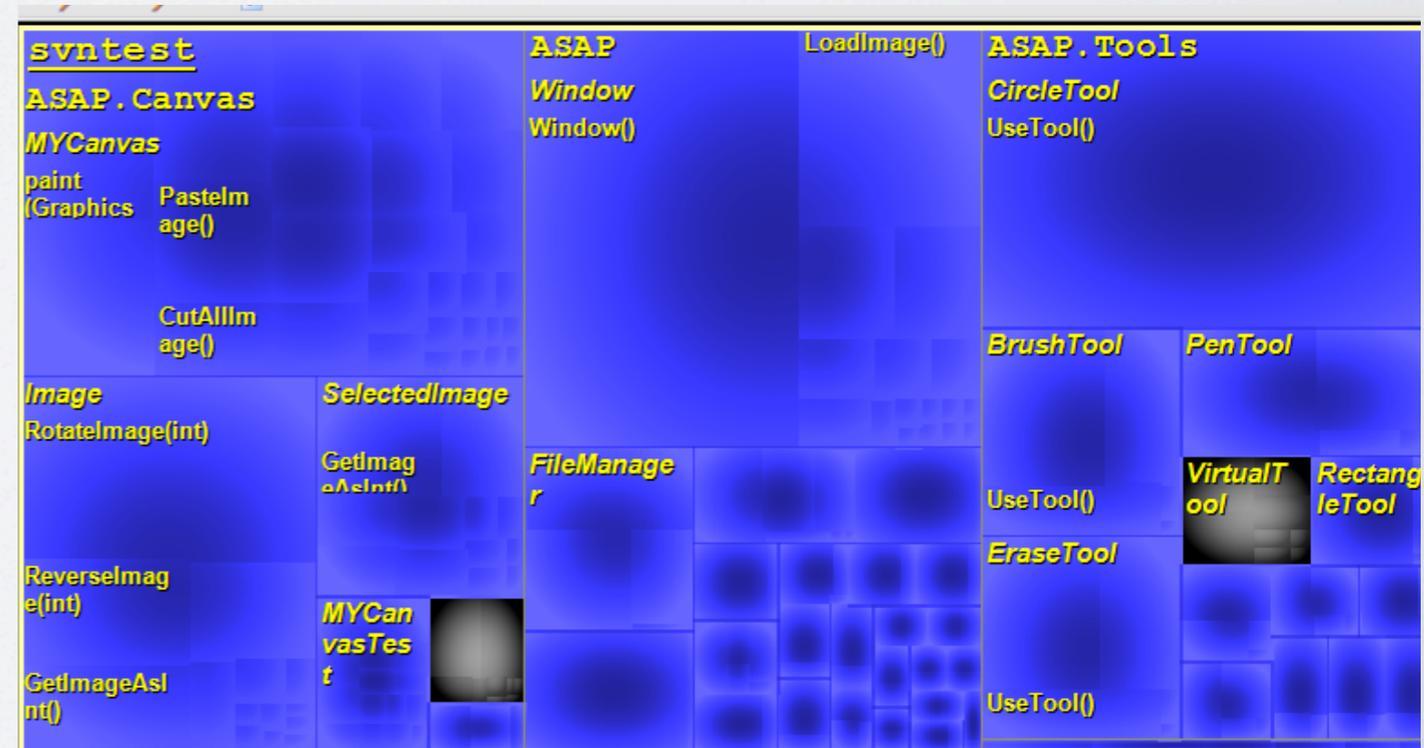
JArchitect

Metrics (Object Oriented Design)

Class with no descendant should be final if possible

상속받은 클래스 마지막에 final keyword를 붙이지 않은 경우

거의 대부분 Class...



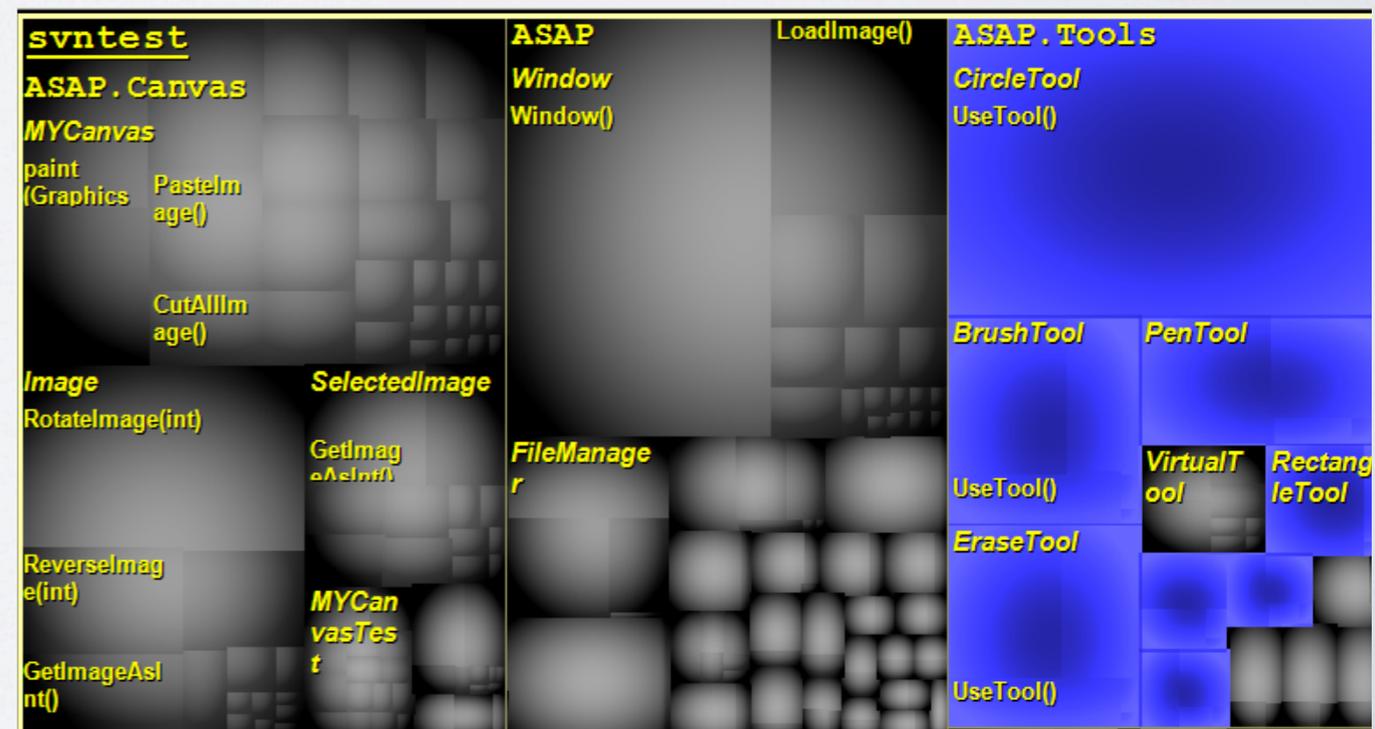
JArchitect

Metrics (Object Oriented Design)

Avoid the Singleton pattern

싱글턴패턴은 객체지향에서 사용하면 위험할 수 있음

ASAP.Tools의 대부분 메서드



JArchitect

| Metrics (Design)

/ 전체적인 디자인은 잘 짜여져 있음

Code Queries and Rules	Items	비고
Avoid packages with few types	1	
Instances size shouldn't be too big	1	
Nested types should not be visible	0	
Don't use obsolete types, methods of fields	0	

JArchitect

Metrics (Architecture and Layering)

Code Queries and Rules	Items	비고
Avoid packages mutually dependent	2	
Avoid packages dependency cycles	1	
Higher cohesion - lower coupling	4	
Projects with poor cohesion(Relational Cohesion)	0	
Projects that don't satisfy the Abstractness / Instability principle	0	
Example of custom rule to check for dependency	0	

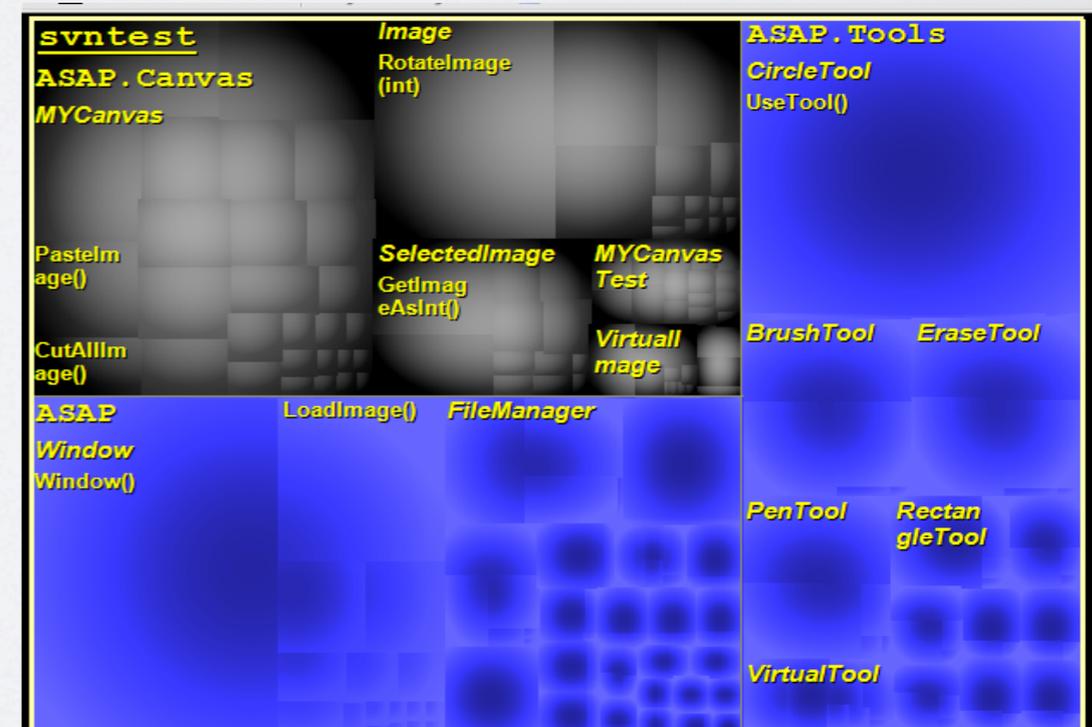
JArchitect

Metrics (Architecture and Layering)

Avoid package mutually dependency

패키지가 상호 의존해선 안되는데, 서로 상호의존 하고 있음

ASAP, ASAP.Tools



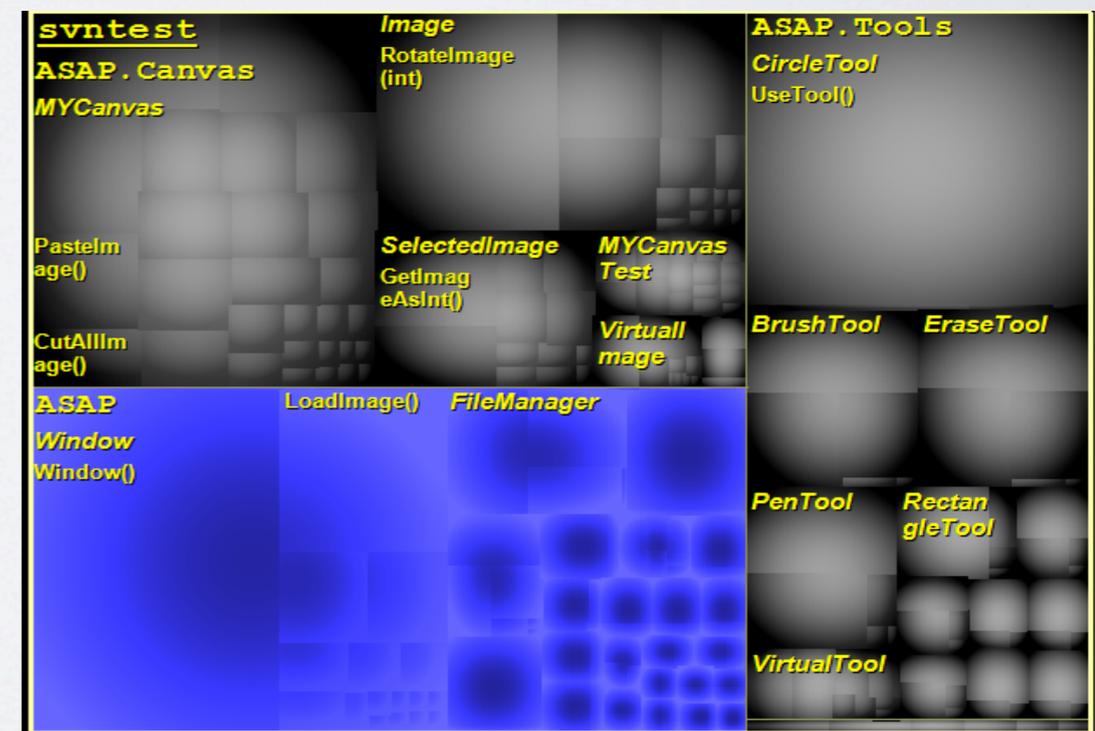
JArchitect

Metrics (Architecture and Layering)

Avoid package dependency cycles

패키지는 dependency cycle을 가져선 안됨

ASAP Package



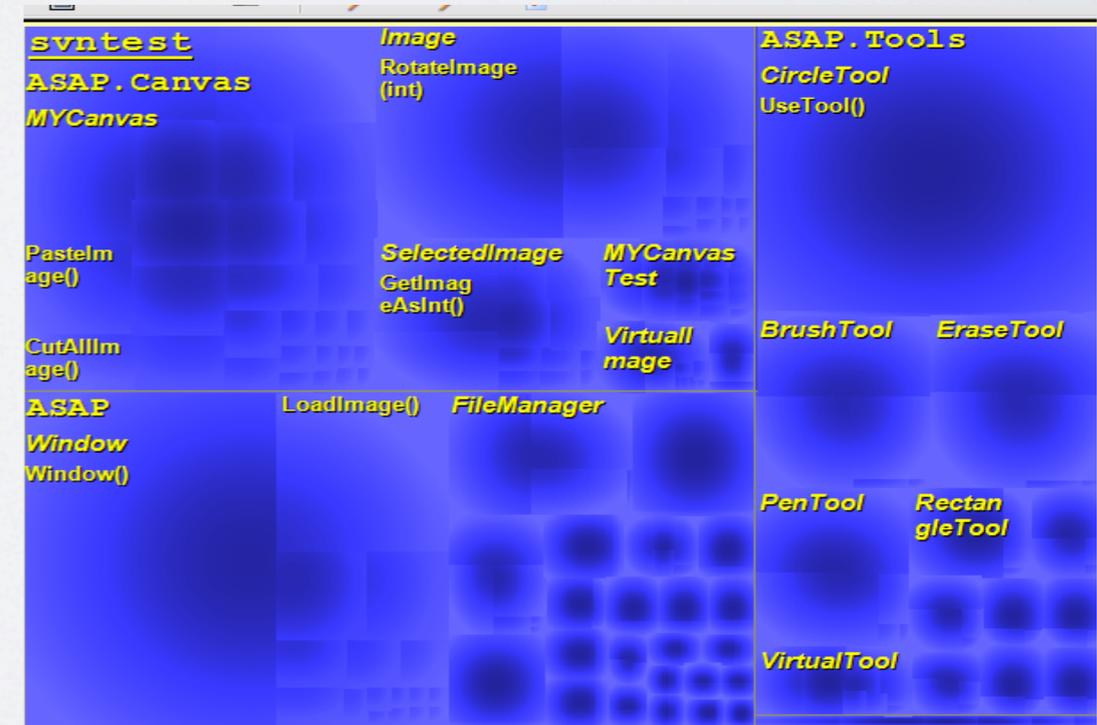
JArchitect

Metrics (Architecture and Layering)

Higher cohesion - lower coupling

응집력이 높고, 낮은 결합력

All Package



JArchitect

Metrics (Visibility)

/ 대체적으로 우수.

Code Queries and Rules	Items	비고
Methods the could have a lower visibility	0	
Fields that could have a lower visibility	2	
Fields should be declared as private	10	
Constructors of abstract classes should be declared as protected on private	0	

JArchitect

Metrics (Purity-Immutability-Side Effects)

Code Queries and Rules	Items	비고
Fields should be marked as ReadOnly when possible	42	
Property Getters should be immutable	0	
Avoid static fields with a mutable field type	8	
A field must not be assigned from outside its parent hierarchy types	1	

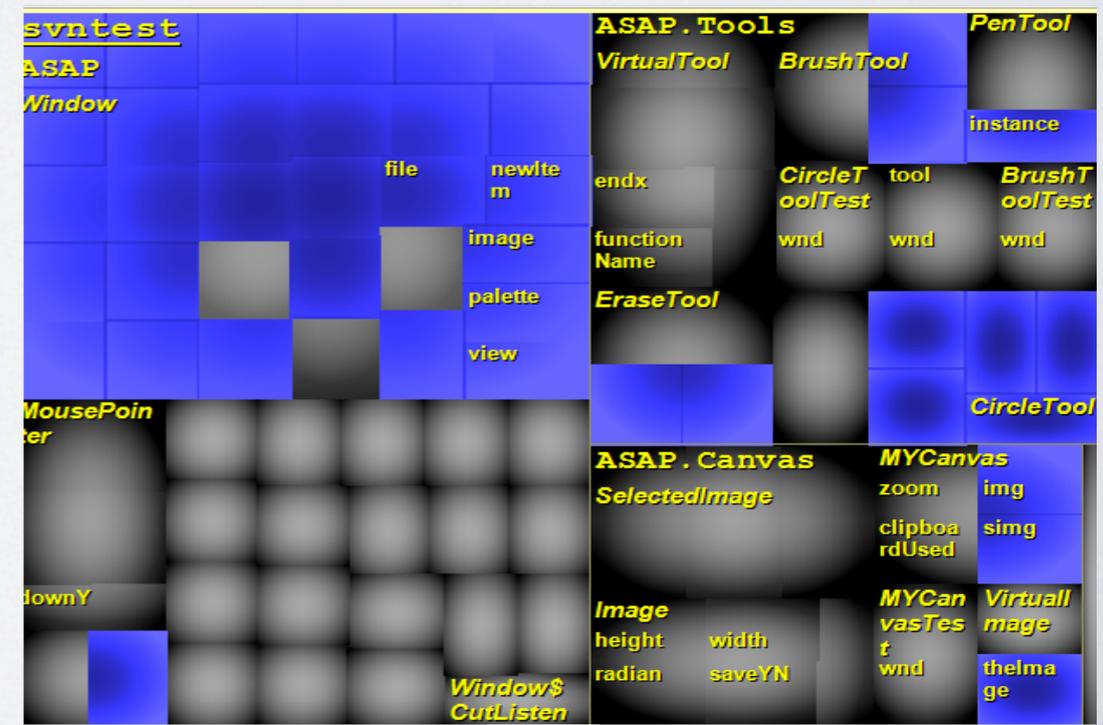
JArchitect

Metrics (Purity-Immutability-Side Effects)

Field should be marked as ReadOnly when as possible

field는 가능한한 ReadOnly로 되어야 함

대부분 Window 관련 필드



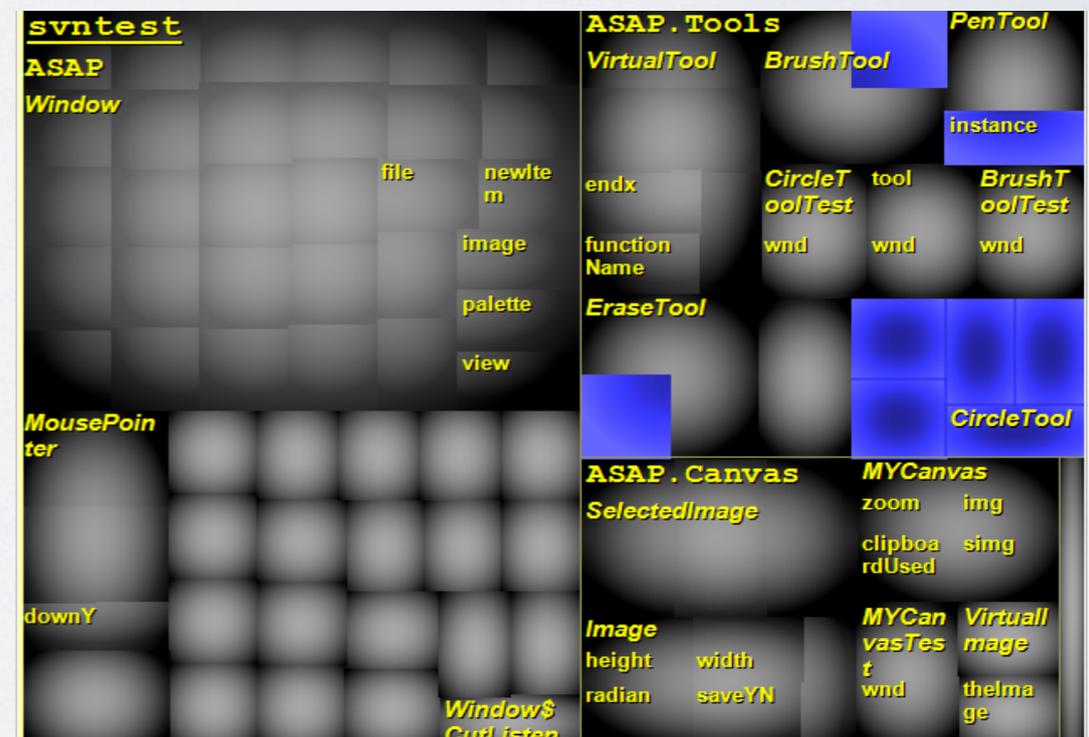
JArchitect

Metrics (Purity-Immutability-Side Effects)

Avoid static field with a mutable field type

static field는 변하기 쉬우면 안된다

ASAP.Tools 패키지에서만 8개



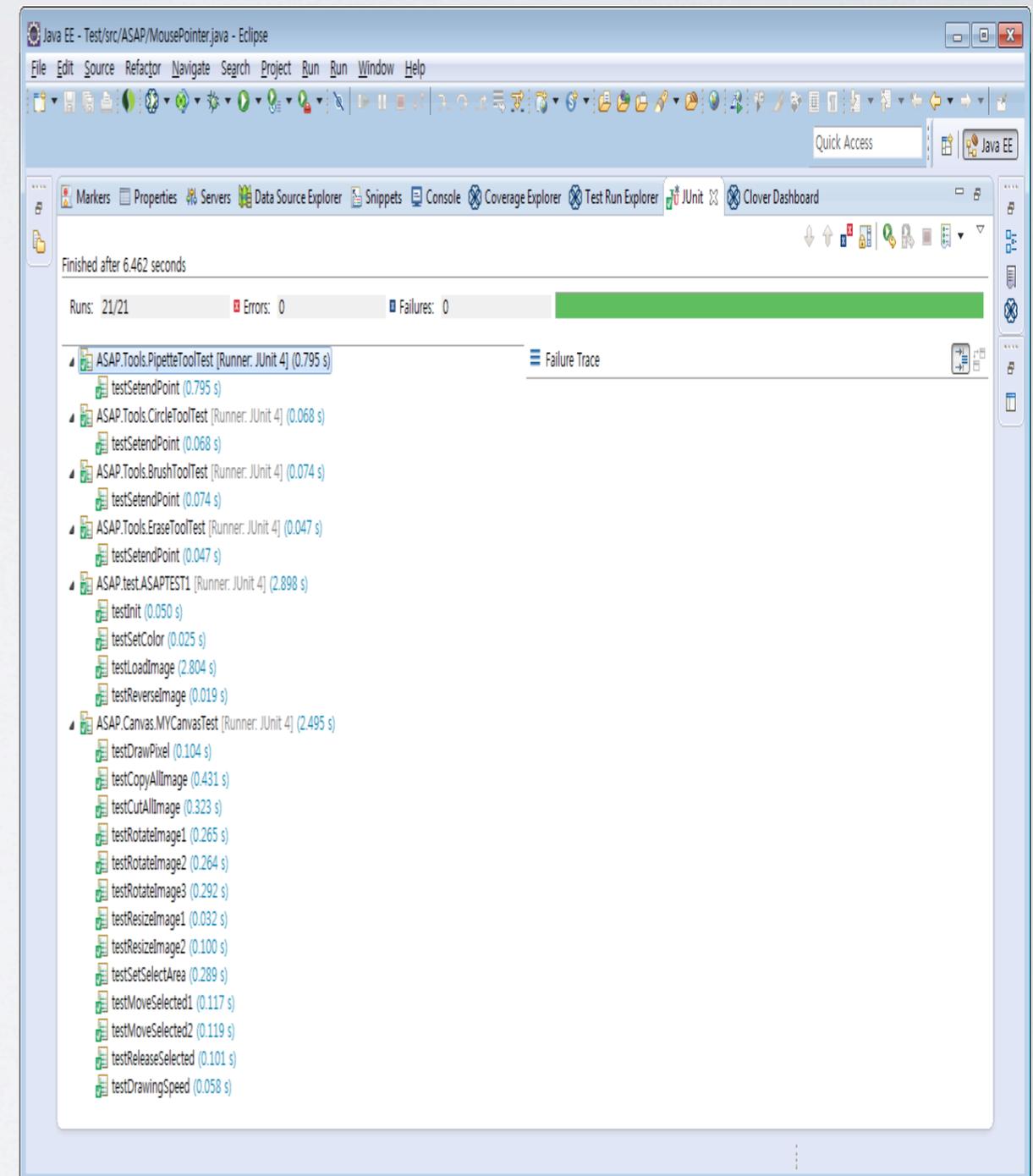
Clover

Clover

Dependency Metrics

JUnit Test Case 21개

전부 정상작동



Clover

Coverage explorer

Show: Application classes Se

Element	Cov%	Av Me Cpx	Cpx	% Covered Branches	% Covered Methods	% Covered Statements
Team4	50.4%	2.3	363.0	36.8%	50.9%	54.5%
ASAP.Tools	52.6%	2.9	125.0	38.1%	60.5%	58.2%
VirtualTool.java	81.8%	1.1	9.0	50.0%	75.0%	87.0%
SelectTool.java	0.0%	1.8	7.0	0.0%	0.0%	0.0%
RectangleTool.java	0.0%	3.2	13.0	0.0%	0.0%	0.0%
PipetteTool.java	88.9%	1.8	7.0	50.0%	100.0%	100.0%
PenTool.java	0.0%	3.6	18.0	0.0%	0.0%	0.0%
MoveTool.java	16.7%	2.5	10.0	0.0%	50.0%	16.7%
EraseTool.java	51.9%	4.2	21.0	42.3%	100.0%	52.2%
CircleTool.java	98.3%	4.8	19.0	92.9%	100.0%	100.0%
BrushTool.java	52.6%	4.2	21.0	42.3%	100.0%	53.2%
ASAP.test	92.3%	1.2	6.0	0.0%	100.0%	87.5%
ASAPTEST1.java	92.3%	1.2	6.0	0.0%	100.0%	87.5%
ASAP.Canvas	66.6%	2.0	131.0	63.7%	56.1%	69.4%
VirtualImage.java	63.6%	1.5	12.0	83.3%	37.5%	75.0%
SelectedImage.java	83.6%	1.8	22.0	78.6%	75.0%	87.2%
MYCanvas.java	53.6%	2.1	59.0	43.8%	50.0%	56.5%
ImageTransferable.java	66.7%	1.2	5.0	50.0%	75.0%	66.7%
Image.java	82.6%	2.4	33.0	84.4%	57.1%	85.7%
ASAP	32.1%	2.1	101.0	4.4%	29.8%	38.4%
Window.java	34.7%	2.2	78.0	2.8%	28.6%	42.1%
MousePointer.java	0.0%	1.0	6.0	0.0%	0.0%	0.0%
GUIMain.java	100.0%	1.0	3.0	0.0%	100.0%	100.0%
FileManager.java	15.6%	4.7	14.0	11.1%	33.3%	16.1%

Clover

| Coverage explorer

/ Statement level에서 Coverage 측정

/ 전체 Code Coverage : 50.4%

/ Covered Branches : 36.8%

Covered Methods : 50.9%

Covered Statements : 54.5%

/ Branch Coverage 비율이 유난히 낮음

Test가 Branches에 대해서는 제대로 행해지지 않음

Clover

Test Run Explorer

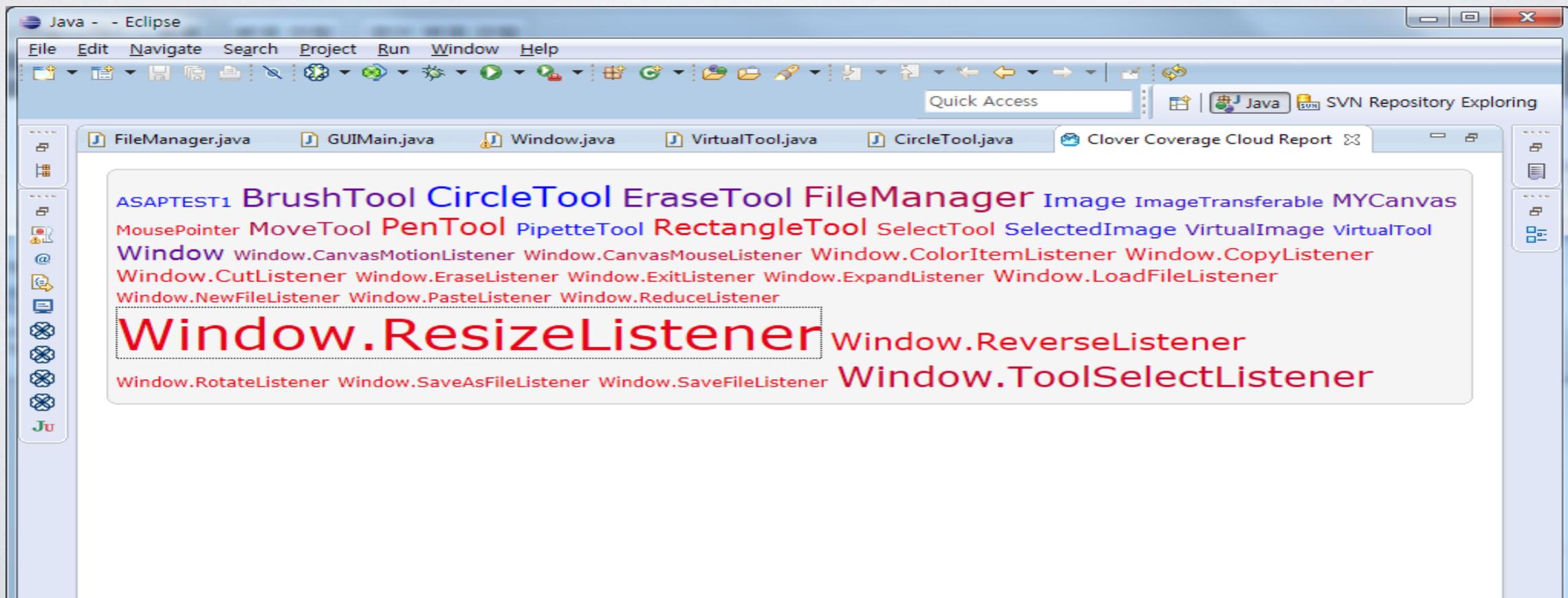
Junit Test Case 정상작동

Tests run:					Coverage Contribution:		
Test	Start	Rslt	Time	Msg	Class	Contrib%	Uniq%
Test					BrushTool	42.3%	42.3%
BrushToolTest.testSetendPoint()	13. 6. 3 오후 8:21	PASS	0.0s		ImprovedLine	22.6%	22.6%
BrushToolTest.testSetendPoint()	13. 6. 3 오후 8:20	PASS	0.0s		SetendPoint	43.5%	43.5%
CircleToolTest.testSetendPoint()	13. 6. 3 오후 8:21	PASS	0.006s		UseTool	100.0%	100.0%
CircleToolTest.testSetendPoint()	13. 6. 3 오후 8:20	PASS	0.014s		CanvasMotionLister	25.0%	25.0%
EraseToolTest.testSetendPoint()	13. 6. 3 오후 8:20	PASS	0.005s		mouseMoved	100.0%	100.0%
EraseToolTest.testSetendPoint()	13. 6. 3 오후 8:21	PASS	0.0s		CanvasMouseListener	12.5%	12.5%
MYCanvasTest.testCopyAllImage()	13. 6. 3 오후 8:21	PASS	0.286s		mouseEntered	100.0%	100.0%
MYCanvasTest.testCopyAllImage()	13. 6. 3 오후 8:20	PASS	0.208s		mouseExited	100.0%	100.0%
MYCanvasTest.testCutAllImage()	13. 6. 3 오후 8:21	PASS	0.282s		CircleTool	94.9%	94.9%
MYCanvasTest.testCutAllImage()	13. 6. 3 오후 8:20	PASS	0.28s		SetendPoint	71.4%	71.4%
MYCanvasTest.testDrawingSpeed()	13. 6. 3 오후 8:20	PASS	0.0s		UseTool	100.0%	100.0%
MYCanvasTest.testDrawingSpeed()	13. 6. 3 오후 8:21	PASS	0.0s		EraseTool	42.9%	42.9%
MYCanvasTest.testDrawingSpeed()	13. 6. 3 오후 8:21	PASS	0.0s		ImprovedLine	22.6%	22.6%
MYCanvasTest.testDrawPixel()	13. 6. 3 오후 8:21	PASS	0.0s		SetendPoint	43.5%	43.5%
MYCanvasTest.testDrawPixel()	13. 6. 3 오후 8:20	PASS	0.0s		UseTool	100.0%	100.0%
MYCanvasTest.testMoveSelected1()	13. 6. 3 오후 8:20	PASS	0.005s		Image	67.4%	67.4%
MYCanvasTest.testMoveSelected1()	13. 6. 3 오후 8:21	PASS	0.049s		getImageAsInt	100.0%	100.0%
MYCanvasTest.testMoveSelected2()	13. 6. 3 오후 8:21	PASS	0.006s		getImageSize	100.0%	100.0%
MYCanvasTest.testMoveSelected2()	13. 6. 3 오후 8:20	PASS	0.042s		rotateImage	100.0%	100.0%
MYCanvasTest.testReleaseSelected()	13. 6. 3 오후 8:20	PASS	0.006s		setImageSize	100.0%	100.0%
MYCanvasTest.testReleaseSelected()	13. 6. 3 오후 8:21	PASS	0.008s		setNotSaved	100.0%	100.0%
MYCanvasTest.testResizeImage1()	13. 6. 3 오후 8:20	PASS	0.001s		ImageTransferable	66.7%	66.7%
MYCanvasTest.testResizeImage1()	13. 6. 3 오후 8:21	PASS	0.0s		ImageTransferat	100.0%	100.0%
MYCanvasTest.testResizeImage2()	13. 6. 3 오후 8:21	PASS	0.046s		getTransferData	60.0%	60.0%
MYCanvasTest.testResizeImage2()	13. 6. 3 오후 8:20	PASS	0.104s		getTransferData	100.0%	100.0%
MYCanvasTest.testRotateImage1()	13. 6. 3 오후 8:20	PASS	0.222s		MYCanvas	45.3%	45.3%
MYCanvasTest.testRotateImage1()	13. 6. 3 오후 8:21	PASS	0.232s		CopyAllImage	100.0%	100.0%
MYCanvasTest.testRotateImage2()	13. 6. 3 오후 8:20	PASS	0.23s		CutAllImage	100.0%	100.0%
MYCanvasTest.testRotateImage2()	13. 6. 3 오후 8:21	PASS	0.219s		DrawPixel	80.0%	80.0%
MYCanvasTest.testRotateImage3()	13. 6. 3 오후 8:21	PASS	0.256s		GetClipboard	100.0%	100.0%
MYCanvasTest.testRotateImage3()	13. 6. 3 오후 8:20	PASS	0.238s		GetPixel	100.0%	100.0%
MYCanvasTest.testSetSelectArea()	13. 6. 3 오후 8:20	PASS	0.002s		MoveSelected	100.0%	100.0%
MYCanvasTest.testSetSelectArea()	13. 6. 3 오후 8:21	PASS	0.001s		ReleaseSelected	89.5%	89.5%
PipetteToolTest.testSetendPoint()	13. 6. 3 오후 8:20	PASS	0.017s		ResizeImage	100.0%	100.0%
PipetteToolTest.testSetendPoint()	13. 6. 3 오후 8:21	PASS	0.011s		RotateImage	100.0%	100.0%

Clover

Cloud Report

가장 복잡도가 높으면서 Test를 하지 않은 곳을 알 수 있다



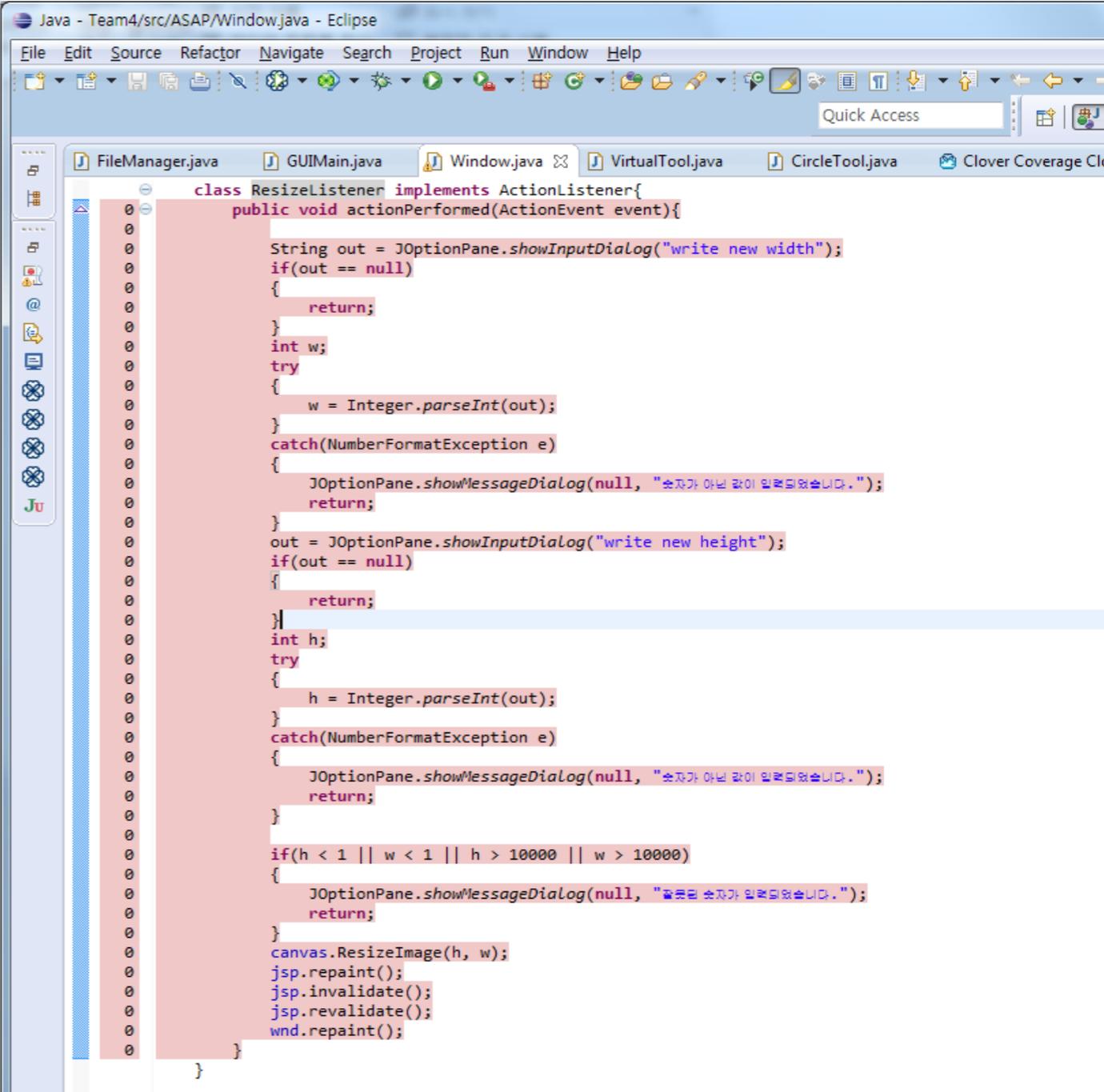
Clover

Cloud Report

Window.ResizeListener의 Code

같은 조건절이 반복되고

Test Case가 전혀 작성되지 않음



```
Java - Team4/src/ASAP/Window.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
FileManager.java GUIMain.java Window.java VirtualTool.java CircleTool.java Clover Coverage Cl...

class ResizeListener implements ActionListener{
    public void actionPerformed(ActionEvent event){
        String out = JOptionPane.showInputDialog("write new width");
        if(out == null)
        {
            return;
        }
        int w;
        try
        {
            w = Integer.parseInt(out);
        }
        catch(NumberFormatException e)
        {
            JOptionPane.showMessageDialog(null, "숫자가 아닌 값이 입력되었습니다.");
            return;
        }
        out = JOptionPane.showInputDialog("write new height");
        if(out == null)
        {
            return;
        }
        int h;
        try
        {
            h = Integer.parseInt(out);
        }
        catch(NumberFormatException e)
        {
            JOptionPane.showMessageDialog(null, "숫자가 아닌 값이 입력되었습니다.");
            return;
        }
        if(h < 1 || w < 1 || h > 10000 || w > 10000)
        {
            JOptionPane.showMessageDialog(null, "불가능한 숫자가 입력되었습니다.");
            return;
        }
        canvas.ResizeImage(h, w);
        jsp.repaint();
        jsp.invalidate();
        jsp.revalidate();
        wnd.repaint();
    }
}
```

Clover

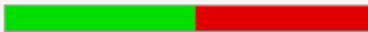
Dashboard

Most Complex Packages :
ASAP.Canvas

Most Complex Classes : MYCanvas

ASAP.Canvas Package는 Test의 비율을 더 높이고 기능을 세분화 해서 Class를 나누는 것이 나을 것이라고 분석됨.

Coverage 37 classes, 778 / 1,543 elements

50.4% 

Test Results 34 / 34 tests 2.78 secs

100%

Most Complex Packages

1. 66.6%  ASAP.Canvas (131)
2. 52.6%  ASAP.Tools (125)
3. 32.3%  ASAP (101)
4. 92.3%  ASAP.test (6)

Most Complex Classes

1. 53.6%  MYCanvas (59)
2. 82.6%  Image (33)
3. 54.3%  Window (31)
4. 83.6%  SelectedImage (22)
5. 52.6%  BrushTool (21)

Top 20 Project Risks

Window.ResizeListener Window.ReverseListener PenTool RectangleTool Window.ColorPicker
Window.LoadFileListener FileManager SelectTool EraseTool BrushTool MoveTool MousePointer Window.ExpandListe

Least Tested Methods

1. 0% Window.ToolSelectListener.actionPerformed(ActionEvent) : void (9)
2. 0% FileManager.SaveAsFile(int[],int,int) : String (6)
3. 0% Window.ResizeListener.actionPerformed(ActionEvent) : void (9)
4. 0% PenTool.ImprovedLine(int,int,int,int) : void (7)
5. 0% MYCanvas.PasteImage() : void (7)
6. 0% FileManager.SaveFile(int[],String,int,int) : int (5)
7. 0% Window.NewImage() : void (4)
8. 0% PenTool.SetendPoint(int,boolean) : void (8)
9. 0% RectangleTool.UseTool() : int (7)
10. 0% MYCanvas.ClearImage() : void (4)
11. 0% Window.SaveImage() : void (4)
12. 0% Window.ExitProgram() : void (4)

JDepend

JDepend

Analysis

- 3개의 Packages에서 Cycle Dependency 발생
- AC = 0, 추상화가 전혀 안되어 있음
- 3개의 Packages 모두 D가 낮아 완전 추상적이면서 안정적이거나, 완전 구체적이면서 불안정한 패키지 두가지 모두에 속하지 못한다고 볼 수 있음

Packages with cycle								
Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
ASAP	28	0	3	3	0.00	0.50	0.50	⚠
ASAP.Canvas	12	0	2	4	0.00	0.66	0.33	⚠
ASAP.Tools	26	0	1	5	0.00	0.83	0.16	⚠

JDepend

Analysis

- 3개의 패키지는 추상화가 전혀 안 되어 있음
- ASAP, ASAP.Canvas는 Main Sequence에서 많이 떨어져있음
- 구체화 되어있는 Package는 불안정성이 높아야 하는데 두개의 경우 그렇지 못함



JDepend

| 결과

- ′ Cycle Dependency가 3개의 Package모두에게서 발생했다는 것은 좋지 않은 현상, 패키지 내의 Class구조가 좋지 않음
- ′ Java의 객체지향에 맞지 않는 설계, 추상화가 전혀 되어있지 않고 클래스간에 계층적 관계가 전혀 없음

Eclipse TPTP

Eclipse TPTP

| 개요

- ′ Test and Performance Tools Platform
- ′ Eclipse에서 프로그래밍 언어에 상관없이 Test와Profiling을 할 수 있는 기반을 만들어주는 Eclipse의 project.
- ′ 3가지 Mode(Execution Time,Memory,Eclipse)가 지원되지만 동시에 수행되지는 않는다

Eclipse TPTP

| 설치

^ 설치 전에 필요로 하는 플러그인 존재

^ WTP(Web Tools Platform) //J2EE기본설치

EMF(Eclipse Modeling Framework SDK)

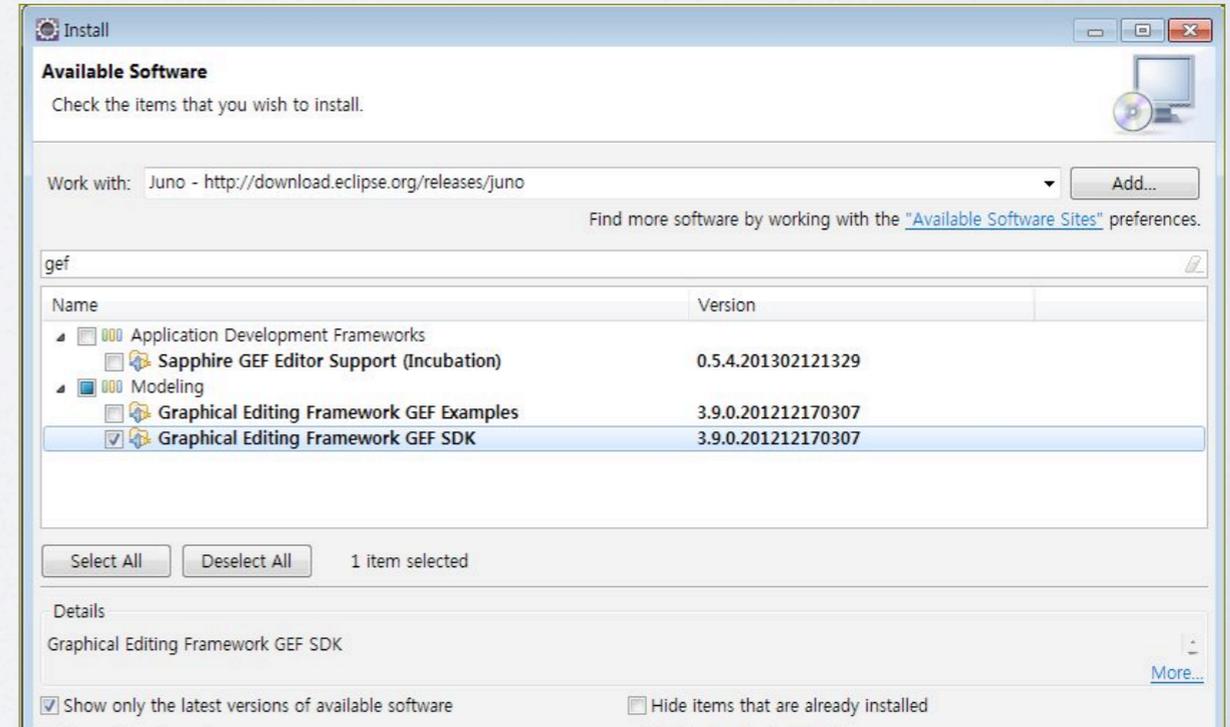
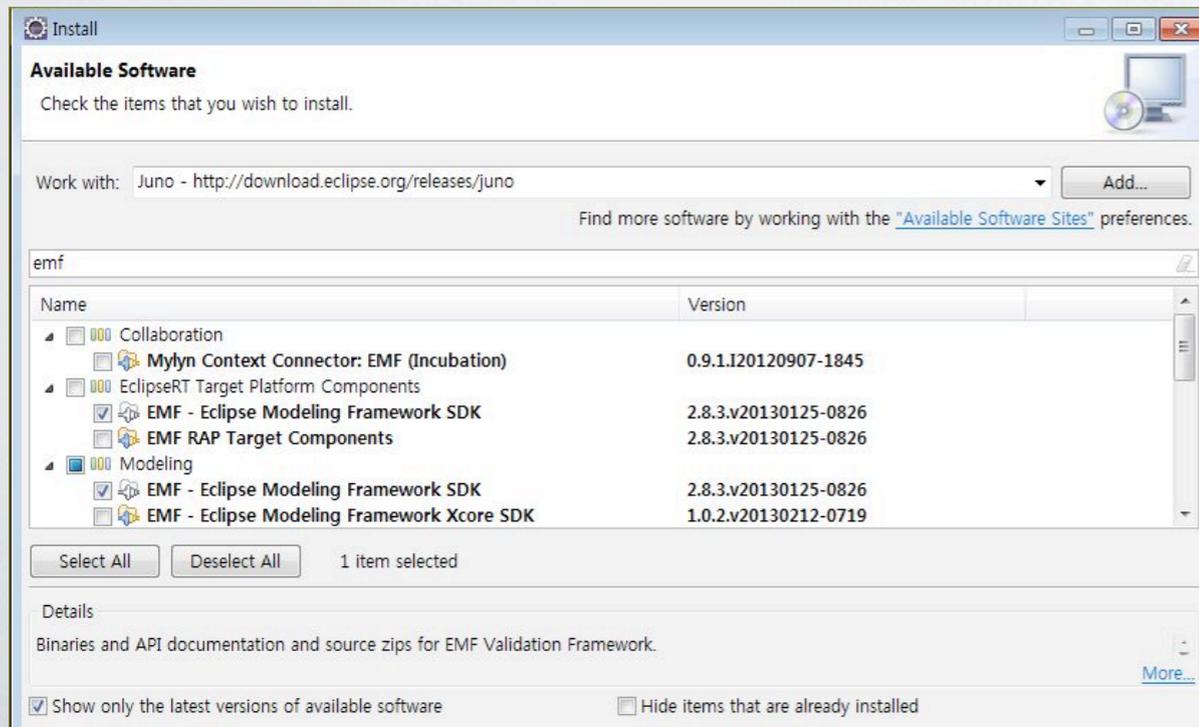
GEF(Graphical Editing Framework SDK)

Eclipse TPTP

| 설치

^ Help -> Install New Software

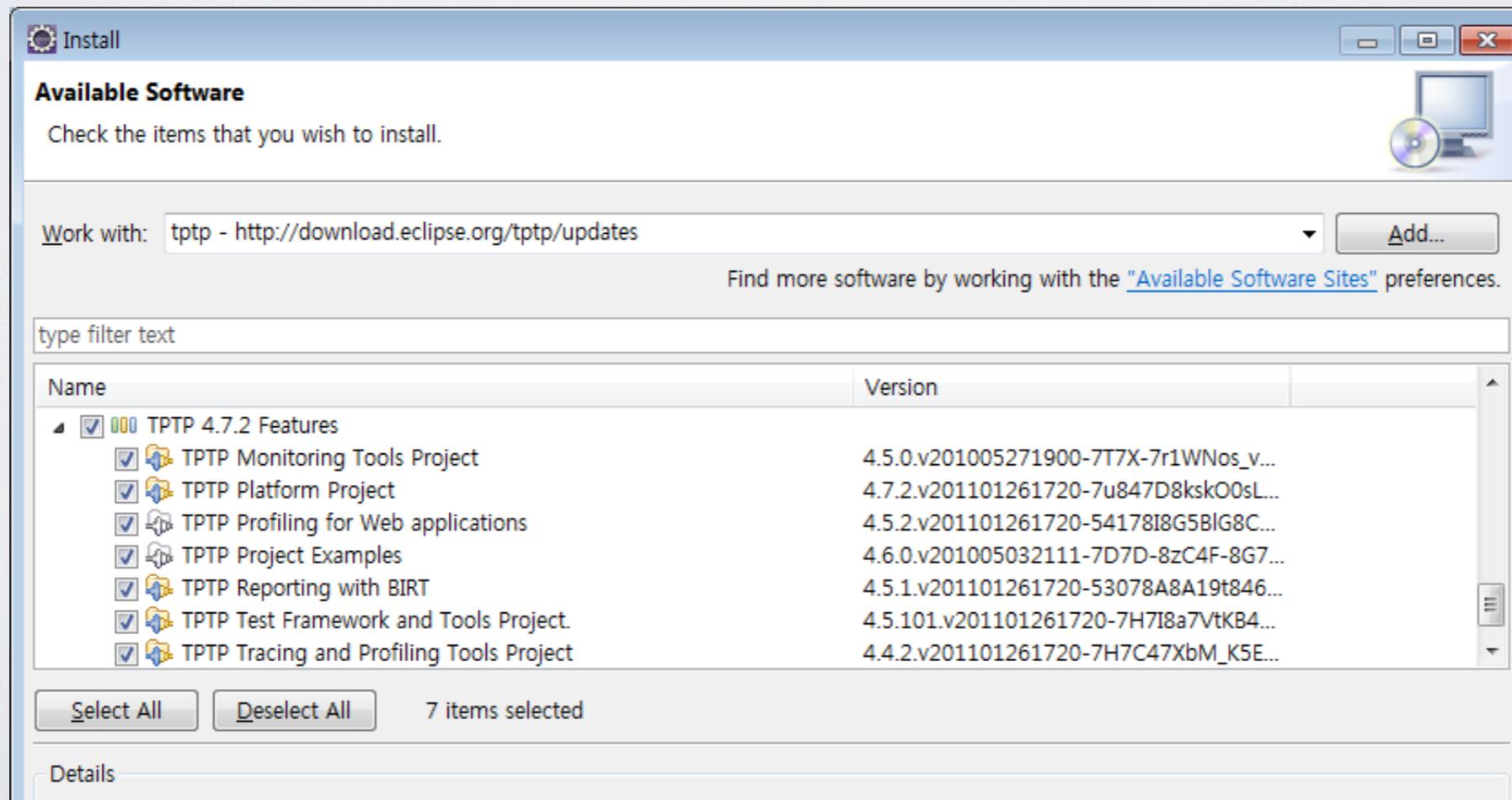
^ WTP, EMF, GEF (<http://download.eclipse.org/releases/juno>)



Eclipse TPTP

| 설치

^ Eclipse TPTP (<http://download.eclipse.org/tptp/updates>)



Eclipse TPTP

| Agent Server

/ 원격지 HOST에 AGENT SERVER를 설치 하므로서 원격지 JVM에 대해서 프로파일 할 수 있는 기능 제공

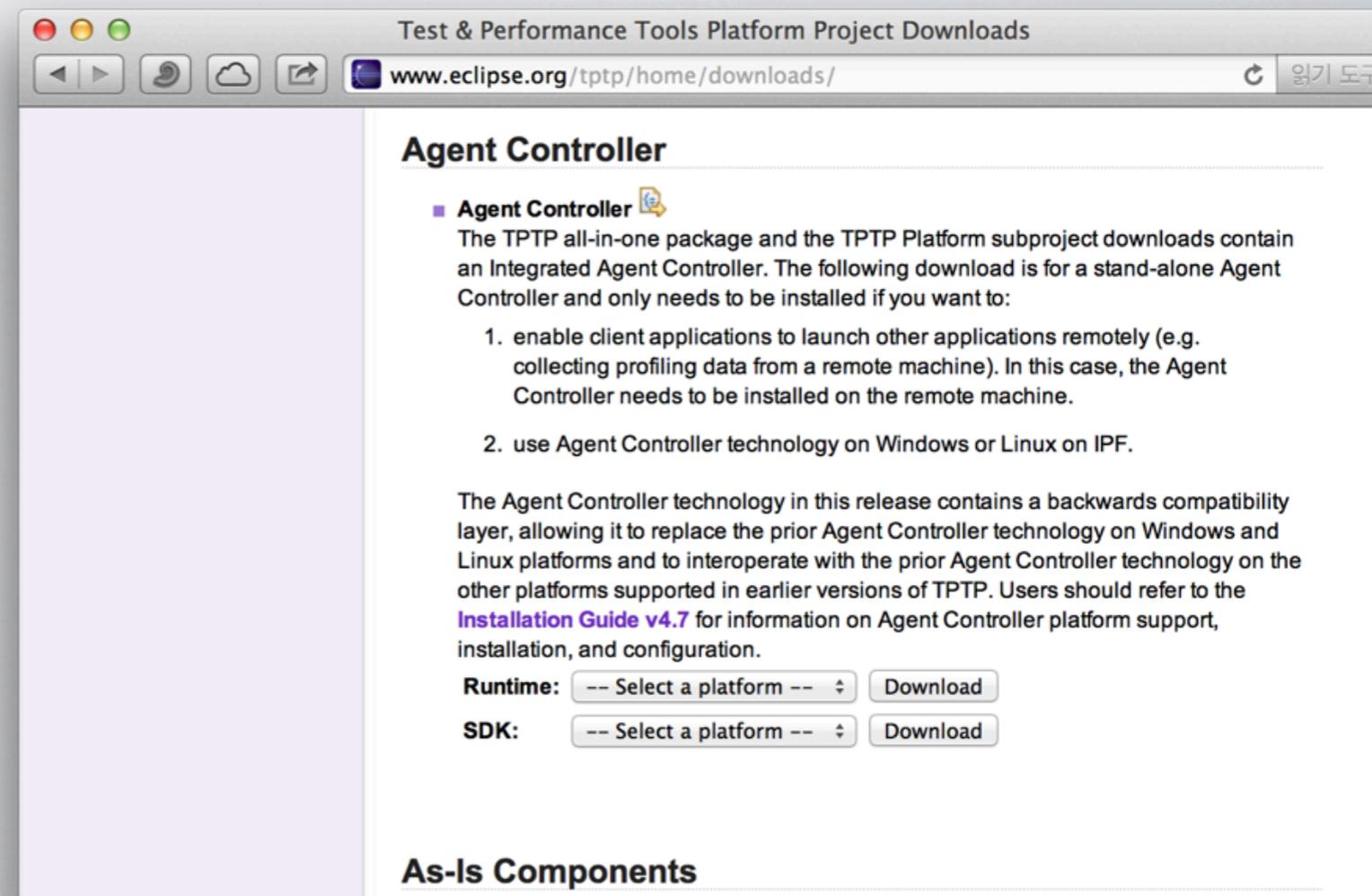
Eclipse TPTP

Agent Server

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

Agent Controller

Runtime : WINDOWS-IA32
Download



The screenshot shows a web browser window titled "Test & Performance Tools Platform Project Downloads" with the URL "www.eclipse.org/tptp/home/downloads/". The page content includes:

Agent Controller

- Agent Controller**
The TPTP all-in-one package and the TPTP Platform subproject downloads contain an Integrated Agent Controller. The following download is for a stand-alone Agent Controller and only needs to be installed if you want to:
 - enable client applications to launch other applications remotely (e.g. collecting profiling data from a remote machine). In this case, the Agent Controller needs to be installed on the remote machine.
 - use Agent Controller technology on Windows or Linux on IPF.

The Agent Controller technology in this release contains a backwards compatibility layer, allowing it to replace the prior Agent Controller technology on Windows and Linux platforms and to interoperate with the prior Agent Controller technology on the other platforms supported in earlier versions of TPTP. Users should refer to the [Installation Guide v4.7](#) for information on Agent Controller platform support, installation, and configuration.

Runtime: -- Select a platform -- [Download]

SDK: -- Select a platform -- [Download]

As-Is Components

Eclipse TPTP

Agent Server

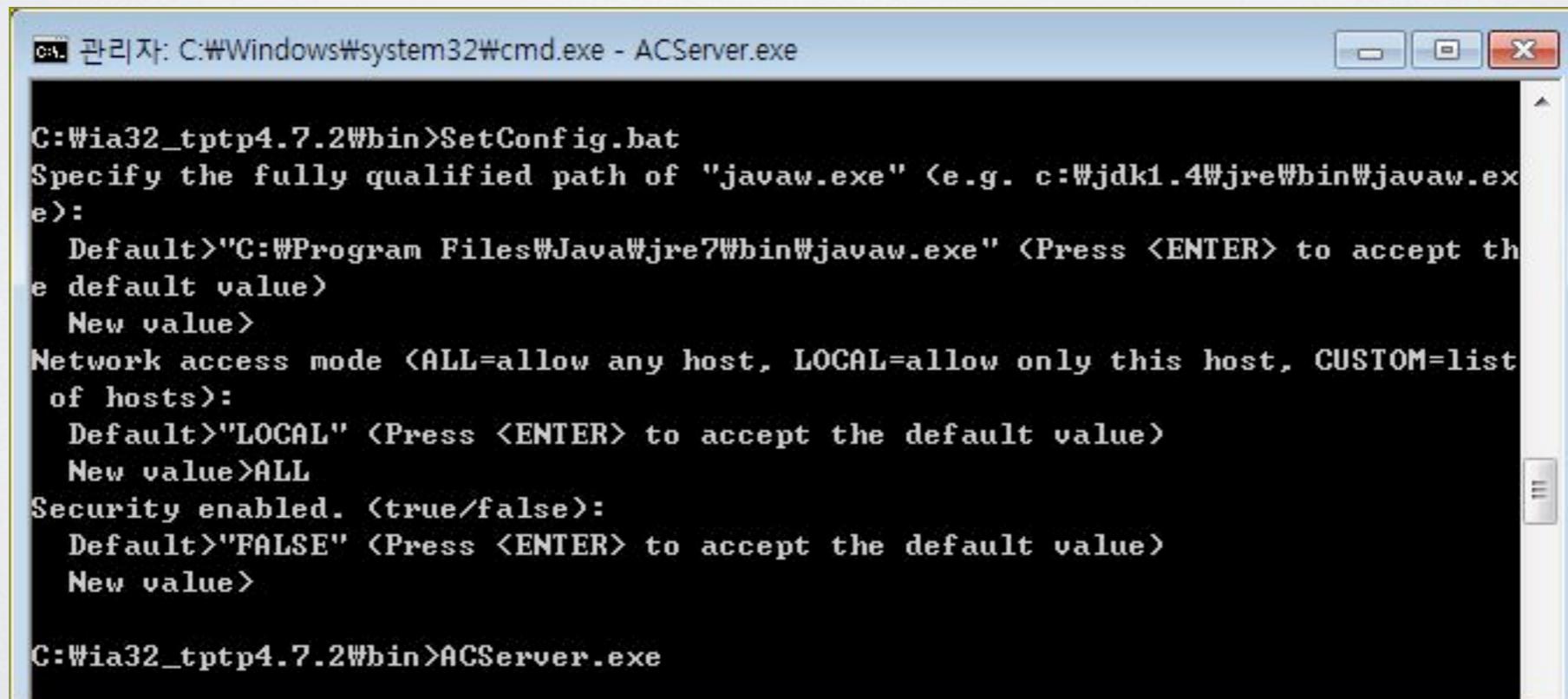
- / 설치폴더\bin\setConfig.bat 실행
- / 사용할 java.exe 실행 파일의 전체 경로 입력
- / 네트워크 Access mode 입력(ALL 입력)
- / Security enabled 입력(기본값 사용)
- / 나머지 옵션은 기본값 사용

```
관리자: C:\Windows\system32\cmd.exe
C:\wia32_tptp4.7.2\bin>SetConfig.bat
Specify the fully qualified path of "javaw.exe" (e.g. c:\jdk1.4\jre\bin\javaw.exe):
Default>"C:\Program Files\Java\jre7\bin\javaw.exe" <Press <ENTER> to accept the default value>
New value>
Network access mode <ALL=allow any host, LOCAL=allow only this host, CUSTOM=list of hosts>:
Default>"LOCAL" <Press <ENTER> to accept the default value>
New value>
Security enabled. <true/false>:
Default>"FALSE" <Press <ENTER> to accept the default value>
New value>
C:\wia32_tptp4.7.2\bin>
```

Eclipse TPTP

Agent Server

원격지에 있는 EclipseTPTP 에서 로컬 호스트 실행중인 JVM을 프로파일링 하기 위해서 Agent server 를 실행해야 한다



```
C:\Windows\system32\cmd.exe - ACServer.exe

C:\wia32_tptp4.7.2\bin>SetConfig.bat
Specify the fully qualified path of "javaw.exe" (e.g. c:\jdk1.4\jre\bin\javaw.exe):
Default>"C:\Program Files\Java\jre7\bin\javaw.exe" <Press <ENTER> to accept the default value>
New value>
Network access mode <ALL=allow any host, LOCAL=allow only this host, CUSTOM=list of hosts>:
Default>"LOCAL" <Press <ENTER> to accept the default value>
New value>ALL
Security enabled. <true/false>:
Default>"FALSE" <Press <ENTER> to accept the default value>
New value>

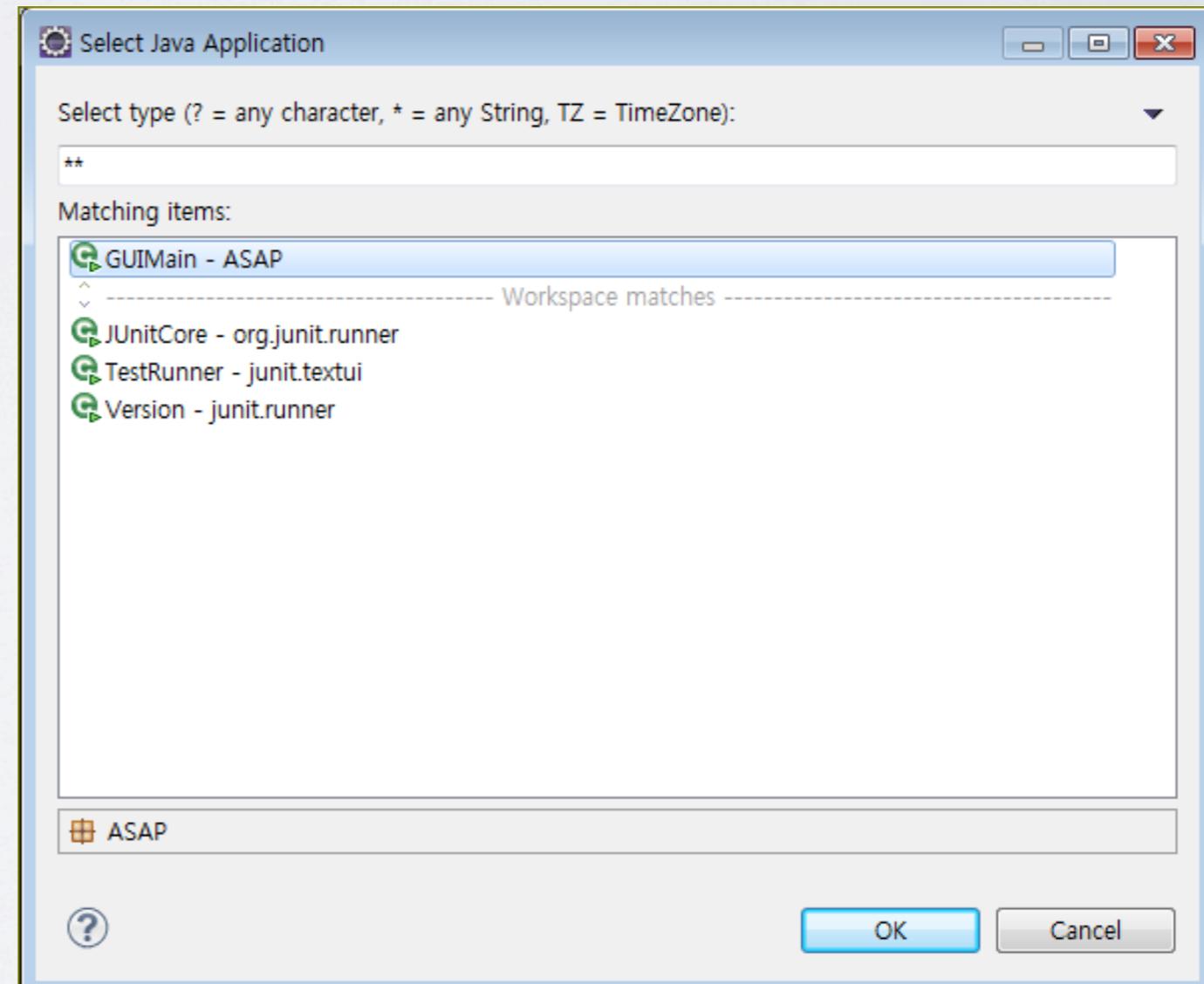
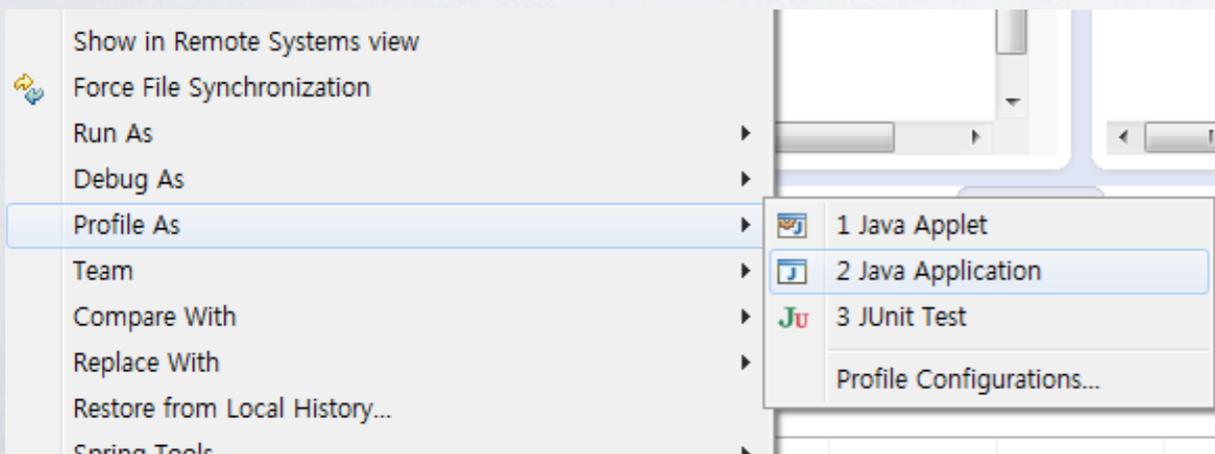
C:\wia32_tptp4.7.2\bin>ACServer.exe
```

Eclipse TPTP

실행

Profile As -> 2. Java Application

Select Matching Item -> ok



Eclipse TPTP

- | Select Profile

 - / Execution Time Analysis

 - / 실행시간 분석

 - / Memory Analysis

 - / 메모리 분석

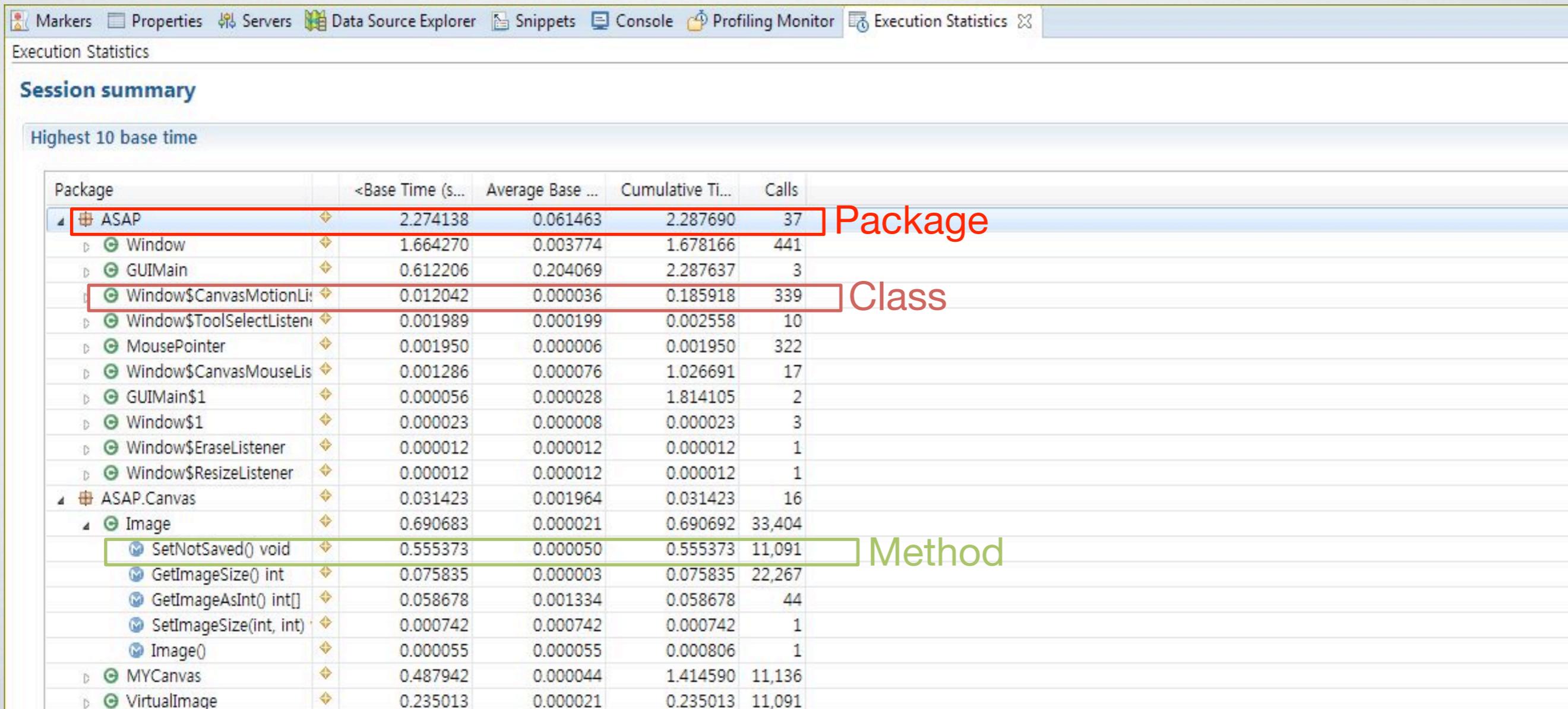
 - / Thread Analysis

 - / 쓰레드 분석

 - / Probe Insertion

Eclipse TPTP

Execution Statistics



Execution Statistics

Session summary

Highest 10 base time

Package	<Base Time (s...	Average Base ...	Cumulative Ti...	Calls
ASAP	2.274138	0.061463	2.287690	37
Window	1.664270	0.003774	1.678166	441
GUIMain	0.612206	0.204069	2.287637	3
Window\$CanvasMotionLi	0.012042	0.000036	0.185918	339
Window\$ToolSelectListen	0.001989	0.000199	0.002558	10
MousePointer	0.001950	0.000006	0.001950	322
Window\$CanvasMouseLis	0.001286	0.000076	1.026691	17
GUIMain\$1	0.000056	0.000028	1.814105	2
Window\$1	0.000023	0.000008	0.000023	3
Window\$EraseListener	0.000012	0.000012	0.000012	1
Window\$ResizeListener	0.000012	0.000012	0.000012	1
ASAP.Canvas	0.031423	0.001964	0.031423	16
Image	0.690683	0.000021	0.690692	33,404
SetNotSaved() void	0.555373	0.000050	0.555373	11,091
GetImageSize() int	0.075835	0.000003	0.075835	22,267
GetImageAsInt() int[]	0.058678	0.001334	0.058678	44
SetImageSize(int, int)	0.000742	0.000742	0.000742	1
Image()	0.000055	0.000055	0.000806	1
MYCanvas	0.487942	0.000044	1.414590	11,136
VirtualImage	0.235013	0.000021	0.235013	11,091

Eclipse TPTP

Execution Statistics

Execution Statistics

Session summary

Highest 10 base time

Package	<Base Time (s...	Average Base ...	Cumulative Ti...	Calls
ASAP	2.274138	0.061463	2.287690	37
Window	1.664270	0.003774	1.678166	441
GUIMain	0.612206	0.204069	2.287637	3
Window\$CanvasMotionLi	0.012042	0.000036	0.185918	339
Window\$ToolSelectListeni	0.001989	0.000199	0.002558	10
MousePointer	0.001950	0.000006	0.001950	322
Window\$CanvasMouseLis	0.001286	0.000076	1.026691	17
GUIMain\$1	0.000056	0.000028	1.814105	2
Window\$1	0.000023	0.000008	0.000023	3
Window\$EraseListener	0.000012	0.000012	0.000012	1
Window\$ResizeListener	0.000012	0.000012	0.000012	1
ASAP.Canvas	0.031423	0.001964	0.031423	16
Image	0.690683	0.000021	0.690692	33,404
SetNotSaved() void	0.555373	0.000050	0.555373	11,091
GetImageSize() int	0.075835	0.000003	0.075835	22,267
GetImageAsInt() int[]	0.058678	0.001334	0.058678	44
SetImageSize(int, int)	0.000742	0.000742	0.000742	1
Image()	0.000055	0.000055	0.000806	1
MYCanvas	0.487942	0.000044	1.414590	11,136
VirtualImage	0.235013	0.000021	0.235013	11,091

상대적으로 많은 시간이 걸리는 Class

Eclipse TPTP

Execution Statistics

Window Class, 프로그램에 사용되는 이미지 불러오기 및 프로그램에 사용되는 객체의 대부분이 생성됨

```
////////////////////////////////////
JMenuBar bar = new JMenuBar() ;
bar.setFont( new Font( "굵글", 0 , 13 ) ) ;

////////////////////////////////////file menu////////////////////////////////////
file = new JMenu("파일") ;

newItem = new JMenuItem( "새 파일" ) ;
newItem.addActionListener( new NewFileListener() ) ;
open = new JMenuItem( "열러오기" ) ;
open.addActionListener( new LoadFileListener() ) ;
save = new JMenuItem( "저장" ) ;
save.addActionListener( new SaveFileListener() ) ;
saveAs = new JMenuItem( "다른이름으로 저장" ) ;
saveAs.addActionListener( new SaveAsFileListener() ) ;
exit = new JMenuItem( "종료" ) ;
exit.addActionListener( new ExitListener() ) ;

file.add( newItem ) ;
file.add( open ) ;
file.add( save ) ;
file.add( saveAs ) ;
file.add( exit ) ;
bar.add( file ) ;

// ///////////////////////////////////edit menu////////////////////////////////////
edit = new JMenu("편집");
cut = new JMenuItem("잘라내기");
cut.addActionListener(new CutListener());
copy = new JMenuItem("복사");
copy.addActionListener(new CopyListener());
paste = new JMenuItem("붙여넣기");
paste.addActionListener(new PasteListener());
// move = new JMenuItem( " ***y***... " ) ;
// select = new JMenuItem( " ... " ) ;
delete = new JMenuItem("지우기");
delete.addActionListener(new EraseListener());
move = new JMenuItem("이동");
move.addActionListener(new ToolSelectListener("move"));

ImageIcon select = new ImageIcon(getClass().getResource("select.png"));
JButton bselect = new JButton(select);
bselect.addActionListener(new ToolSelectListener("select"));
vertical.add(bselect);

ImageIcon eraser = new ImageIcon(getClass().getResource("eraser.png"));
JButton beraser = new JButton(eraser);
beraser.addActionListener(new ToolSelectListener("eraser"));
vertical.add(beraser);

ImageIcon pip = new ImageIcon(getClass().getResource("pip.png"));
JButton bpip = new JButton(pip);
bpip.addActionListener(new ToolSelectListener("pip"));
vertical.add(bpip);

ImageIcon pencil = new ImageIcon(getClass().getResource("pencil.png"));
JButton bpencil= new JButton(pencil);
bpencil.addActionListener(new ToolSelectListener("pencil"));
vertical.add(bpencil);

ImageIcon brush = new ImageIcon(getClass().getResource("brush.png"));
JButton bbrush= new JButton(brush);
bbrush.addActionListener(new ToolSelectListener("brush"));
vertical.add(bbrush);

ImageIcon rectangle = new ImageIcon(getClass().getResource("rectangle.png"));
JButton brectangle= new JButton(rectangle);
brectangle.addActionListener(new ToolSelectListener("rectangle"));
vertical.add(brectangle);

ImageIcon circle = new ImageIcon(getClass().getResource("circle.png"));
JButton bcircle= new JButton(circle);
bcircle.addActionListener(new ToolSelectListener("circle"));
vertical.add(bcircle);

vertical.setAlignmentY(Component.CENTER_ALIGNMENT);

c.add(vertical, BorderLayout.WEST);

colorpanel = new JLabel("color");
Color col = new Color(color);
```

Eclipse TPTP

Execution Statistics

Markers Properties Servers Data Source Explorer Snippets Console Profiling Monitor Execution Statistics

Execution Statistics - ASAP.GUIMain at cse8-PC [PID: 1824]

Method Invocation Details

Selected method 선택된 method

Calls	Method	Class	Package	<Base Time (s...	Average Base ...	Cumulative Ti...	Cumulative C...
11,091	SetNotSaved() void	Image	ASAP.Canvas	0.555373	0.000050	0.555373	0.000000

Selected method is invoked by:

Invoked by	Method	Class	Package	<Base Time (seconds)	Average Base Time...	Cumulativ...	Calls	Cumulative C.
11089	DrawPixel(int, int, int) void	MYCanvas	ASAP.Canvas	0.261743	0.000024	1.127562	11,089	0.000000
2	ReleaseSelected() void	MYCanvas	ASAP.Canvas	0.000115	0.000058	0.000151	2	0.000000

수행회수 호출한 Method 포함된 class 포함된 package 수행시간 평균수행시간 누적 수행시간

Selected method invokes:

Eclipse TPTP

Execution Statistics

실행시간이 가장 많이 걸렸던 window() method를 열어보니, 다음과 같이 많은 method들을 호출하는 걸 확인할 수 있었음.

Invokes	Method	Class	Package	<Base Time (s...	Average Base ...	Cumulative Ti...	Calls	Cumulative C...
1	Window\$ColorItemListener(ASAP....	Window\$Col...	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$EraseListener(ASAP.Win...	Window\$Era...	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$CanvasMouseListener(A...	Window\$Ca...	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$1(ASAP.Window)	Window\$1	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$NewFileListener(ASAP.W...	Window\$Ne...	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$CopyListener(ASAP.Win...	Window\$Co...	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$CutListener(ASAP.Windo...	Window\$Cu...	ASAP	0.000010	0.000010	0.000010	1	0.000000
1	Window\$SaveFileListener(ASAP.W...	Window\$Sa...	ASAP	0.000009	0.000009	0.000009	1	0.000000
1	Window\$LoadFileListener(ASAP....	Window\$Lo...	ASAP	0.000009	0.000009	0.000009	1	0.000000
1	FileManager()	FileManager	ASAP	0.000008	0.000008	0.000008	1	0.000000

Eclipse TPTP

Memory Statistics

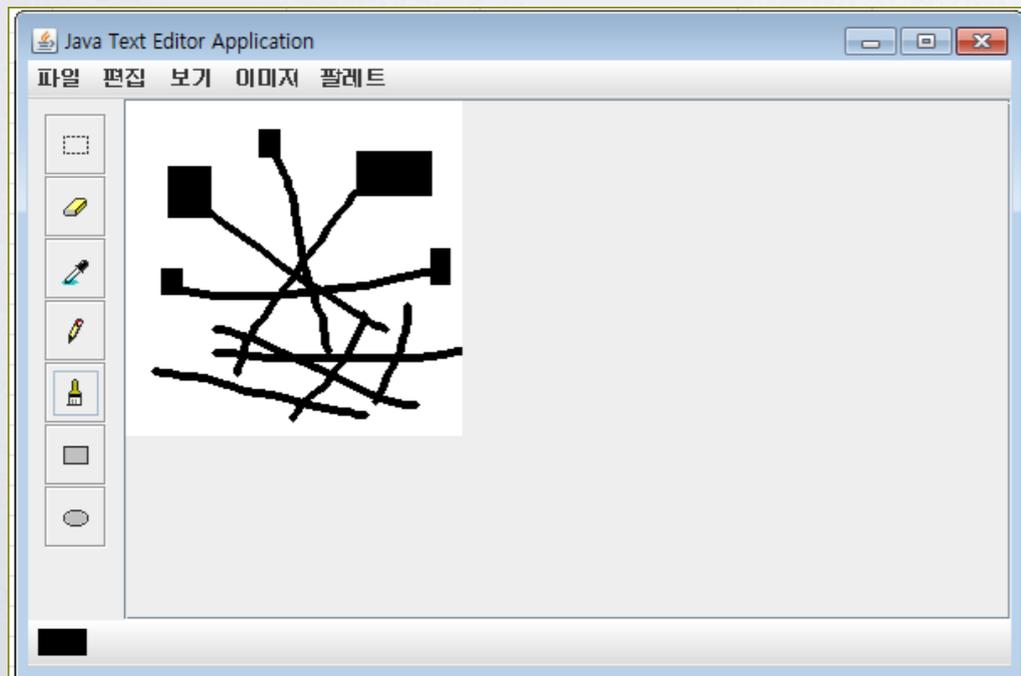
Live Instances : Garbage collection이 되지 않은 instances수로 Memory에 살아 있는 instances 수

Avg. AGE : 한번의 Garbage Collect로부터 살아남으면 Age 하나 증가

Active Size(bytes) : Live Instances의 Memory Size

Total Instances : 전체 Instances의 수

Total Size(bytes) : Total Instances의 수



Memory Analysis - ASAP.GUIMain at cse8-PC [PID: 2020]

Memory Statistics

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

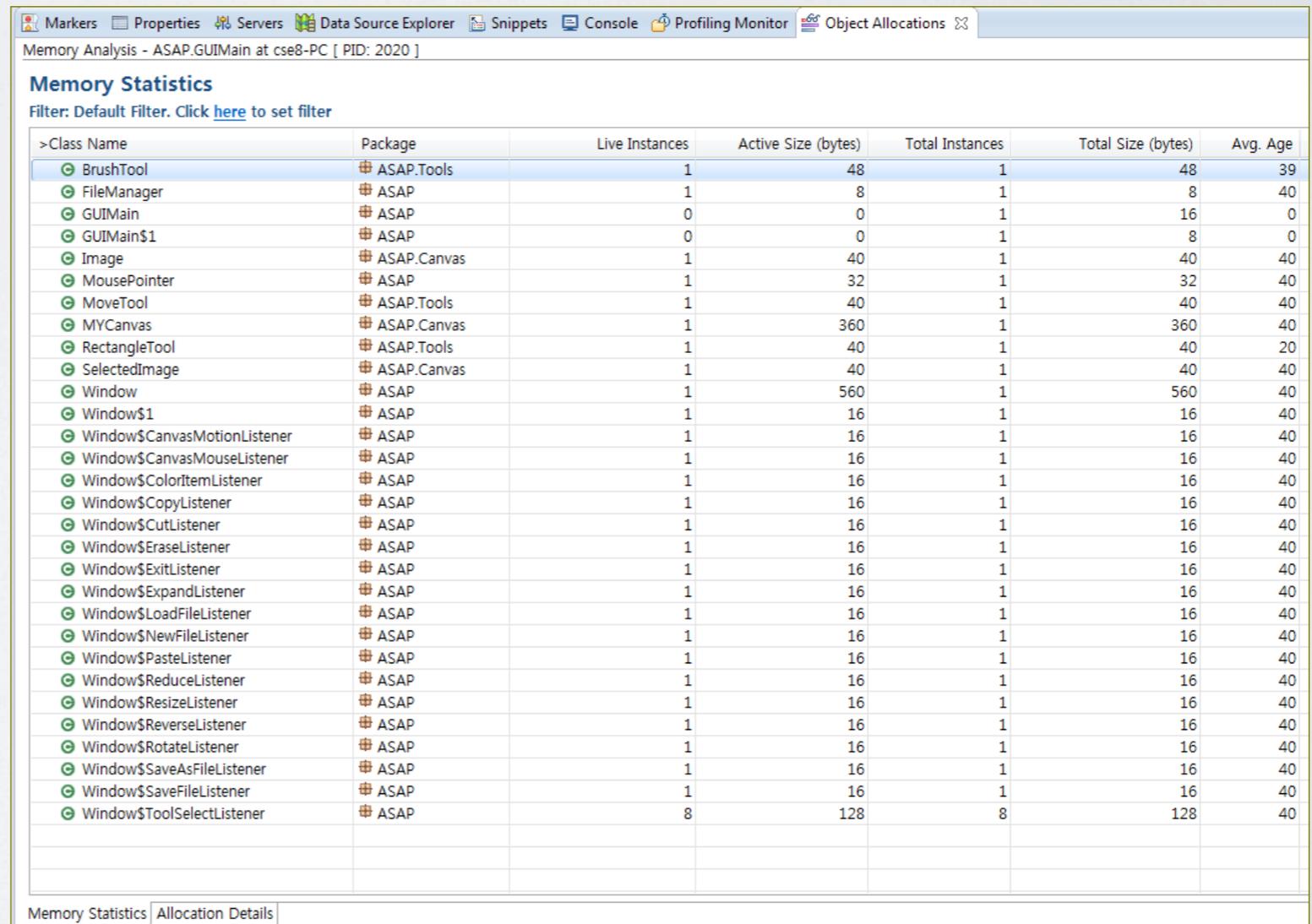
>Class Name	Package	Live Instances	Active Size (bytes)	Total Instances	Total Size (bytes)	Avg. Age
BrushTool	ASAP.Tools	1	48	1	48	39
FileManager	ASAP	1	8	1	8	40
GUIMain	ASAP	0	0	1	16	0
GUIMain\$1	ASAP	0	0	1	8	0
Image	ASAP.Canvas	1	40	1	40	40
MousePointer	ASAP	1	32	1	32	40
MoveTool	ASAP.Tools	1	40	1	40	40
MYCanvas	ASAP.Canvas	1	360	1	360	40
RectangleTool	ASAP.Tools	1	40	1	40	20
SelectedImage	ASAP.Canvas	1	40	1	40	40
Window	ASAP	1	560	1	560	40
Window\$1	ASAP	1	16	1	16	40
Window\$CanvasMotionListener	ASAP	1	16	1	16	40
Window\$CanvasMouseListener	ASAP	1	16	1	16	40
Window\$ColorItemListener	ASAP	1	16	1	16	40
Window\$CopyListener	ASAP	1	16	1	16	40
Window\$CutListener	ASAP	1	16	1	16	40
Window\$EraseListener	ASAP	1	16	1	16	40
Window\$ExitListener	ASAP	1	16	1	16	40
Window\$ExpandListener	ASAP	1	16	1	16	40
Window\$LoadFileListener	ASAP	1	16	1	16	40
Window\$NewFileListener	ASAP	1	16	1	16	40
Window\$PasteListener	ASAP	1	16	1	16	40
Window\$ReduceListener	ASAP	1	16	1	16	40
Window\$ResizeListener	ASAP	1	16	1	16	40
Window\$ReverseListener	ASAP	1	16	1	16	40
Window\$RotateListener	ASAP	1	16	1	16	40
Window\$SaveAsFileListener	ASAP	1	16	1	16	40
Window\$SaveFileListener	ASAP	1	16	1	16	40
Window\$ToolSelectListener	ASAP	8	128	8	128	40

Memory Statistics | Allocation Details

Eclipse TPTP

Memory Statistics

여기서도 window, canvas의 메모리 점유율이 다른 클래스보다 다소 많이 사용되는 것을 확인



Memory Analysis - ASAP.GUIMain at cse8-PC [PID: 2020]

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

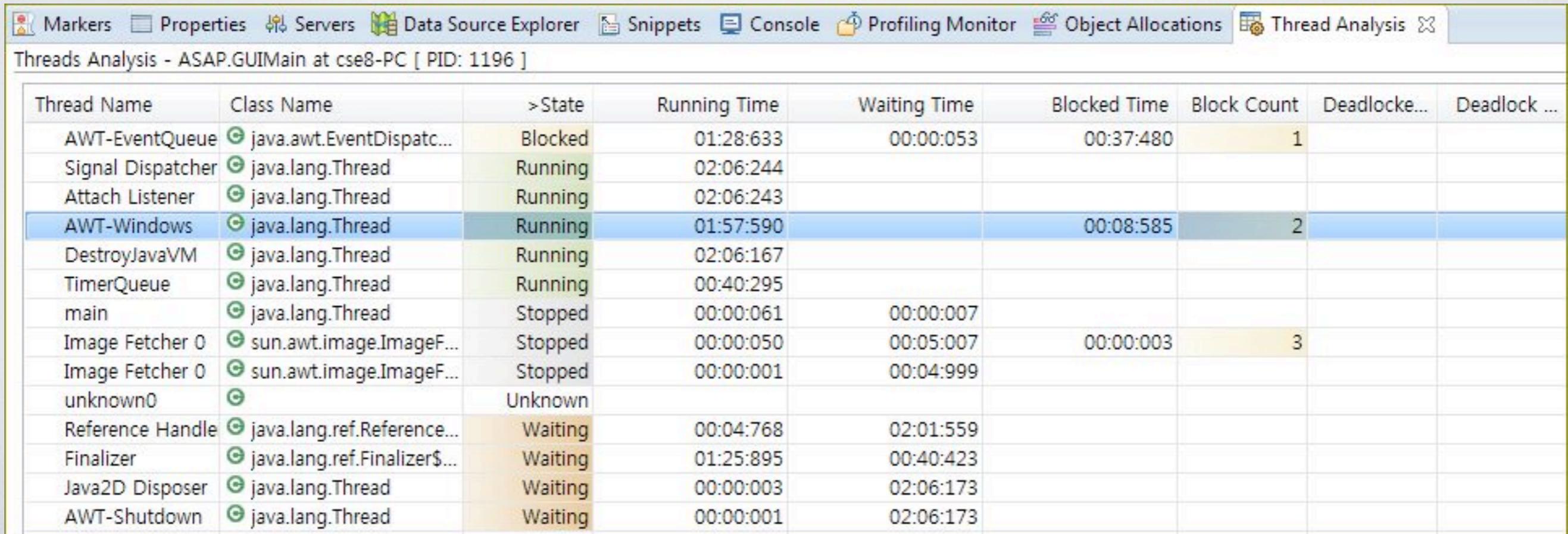
>Class Name	Package	Live Instances	Active Size (bytes)	Total Instances	Total Size (bytes)	Avg. Age
BrushTool	ASAP.Tools	1	48	1	48	39
FileManager	ASAP	1	8	1	8	40
GUIMain	ASAP	0	0	1	16	0
GUIMain\$1	ASAP	0	0	1	8	0
Image	ASAP.Canvas	1	40	1	40	40
MousePointer	ASAP	1	32	1	32	40
MoveTool	ASAP.Tools	1	40	1	40	40
MYCanvas	ASAP.Canvas	1	360	1	360	40
RectangleTool	ASAP.Tools	1	40	1	40	20
SelectedImage	ASAP.Canvas	1	40	1	40	40
Window	ASAP	1	560	1	560	40
Window\$1	ASAP	1	16	1	16	40
Window\$CanvasMotionListener	ASAP	1	16	1	16	40
Window\$CanvasMouseListener	ASAP	1	16	1	16	40
Window\$ColorItemListener	ASAP	1	16	1	16	40
Window\$CopyListener	ASAP	1	16	1	16	40
Window\$CutListener	ASAP	1	16	1	16	40
Window\$EraseListener	ASAP	1	16	1	16	40
Window\$ExitListener	ASAP	1	16	1	16	40
Window\$ExpandListener	ASAP	1	16	1	16	40
Window\$LoadFileListener	ASAP	1	16	1	16	40
Window\$NewFileListener	ASAP	1	16	1	16	40
Window\$PasteListener	ASAP	1	16	1	16	40
Window\$ReduceListener	ASAP	1	16	1	16	40
Window\$ResizeListener	ASAP	1	16	1	16	40
Window\$ReverseListener	ASAP	1	16	1	16	40
Window\$RotatetListener	ASAP	1	16	1	16	40
Window\$SaveAsFileListener	ASAP	1	16	1	16	40
Window\$SaveFileListener	ASAP	1	16	1	16	40
Window\$ToolSelectListener	ASAP	8	128	8	128	40

Memory Statistics | Allocation Details

Eclipse TPTP

Thread Analysis

몇 가지의 Test Case 수행에도 deadlock 현상은 발견되지 않음



The screenshot shows the Eclipse IDE's Thread Analysis window. The title bar includes various toolbars like Markers, Properties, Servers, Data Source Explorer, Snippets, Console, Profiling Monitor, Object Allocations, and Thread Analysis. The main window title is "Threads Analysis - ASAP.GUIMain at cse8-PC [PID: 1196]". Below the title bar is a table with the following columns: Thread Name, Class Name, >State, Running Time, Waiting Time, Blocked Time, Block Count, Deadlock..., and Deadlock ...

Thread Name	Class Name	>State	Running Time	Waiting Time	Blocked Time	Block Count	Deadlock...	Deadlock ...
AWT-EventQueue	java.awt.EventDispatc...	Blocked	01:28:633	00:00:053	00:37:480	1		
Signal Dispatcher	java.lang.Thread	Running	02:06:244					
Attach Listener	java.lang.Thread	Running	02:06:243					
AWT-Windows	java.lang.Thread	Running	01:57:590		00:08:585	2		
DestroyJavaVM	java.lang.Thread	Running	02:06:167					
TimerQueue	java.lang.Thread	Running	00:40:295					
main	java.lang.Thread	Stopped	00:00:061	00:00:007				
Image Fetcher 0	sun.awt.image.ImageF...	Stopped	00:00:050	00:05:007	00:00:003	3		
Image Fetcher 0	sun.awt.image.ImageF...	Stopped	00:00:001	00:04:999				
unknown0		Unknown						
Reference Handle	java.lang.ref.Reference...	Waiting	00:04:768	02:01:559				
Finalizer	java.lang.ref.Finalizer\$...	Waiting	01:25:895	00:40:423				
Java2D Disposer	java.lang.Thread	Waiting	00:00:003	02:06:173				
AWT-Shutdown	java.lang.Thread	Waiting	00:00:001	02:06:173				

Eclipse TPTP

Thread Analysis

Thread Visualizer

직접 추가한 thread는 없고, 시스템에서 임의로 생성한 것들로만 확인 됨

The screenshot displays the Eclipse IDE's Thread Analysis tool. The main window shows a timeline of thread activity for the process 'ASAP.GUIMain at MSDN-SPECIAL [PID: 5900]'. The timeline is divided into three thread groups: 'system', '<undefined>', and 'main'. The 'main' group contains threads such as 'main', 'AWT-Shutdown', 'AWT-Window', 'AWT-EventQueue-0', 'DestroyJavaVM', and two 'Image Fetcher' threads. The 'system' group includes threads like 'Reference Handler', 'Finalizer', 'Signal Dispatcher', 'Attach Listener', 'Java2D Disposer', 'D3D Screen Updater', and 'TimerQueue'. The '<undefined>' group contains 'unknown0'. The 'Thread Visualizer's Legend' dialog is open in the foreground, showing the legend for thread states and interactions. The legend includes: Thread states: Unknown (black), Running (green), Sleeping (blue), Waiting (orange), Blocked (yellow), Deadlocked (red), Stopped (grey). Thread interactions: Start Thread (up arrow), Join / Terminate (up arrow), Wait / Notify (up arrow with green dot), Interrupt (up arrow with red dot), Release / Acquire (up arrow with blue dot), Wait Timeout Exceed (down arrow with blue dot). The dialog has an 'OK' button at the bottom right.

exit;_