

Static Analysis 보고서

(Clover / Sonar / PMD)

과 목 : 소프트웨어 검증

담당교수 : 유 준범 교수님

팀 : T3

팀원 :

200711453 류진렬 200711477 황진수

200711428 박기성 201360220 황 민

200312489 유현덕

1. Clover

1.1 Clover 소개

코드 커버리지를 확인할 수 있는 정적 분석 도구로 statement 커버리지, branch 커버리지, method 커버리지 등을 상세히 확인할 수 있다. 이를 통해 테스트에 의해 프로그램이 어떻게 작동하는지 확인할 수 있고 전체 테스트 부분 중 얼마만큼 완료되었는지 판단할 수 있다. 또한 그 결과에 따라서 테스트 기능을 개량할 수 있고 검사 항목, 기능 추가 여부를 결정하기도 한다. 이러한 커버리지율을 확인해주는 정적 코드 검사 도구가 Clover 이다.

커버리지 분석이란 테스트의 대상 중에 테스트에 해당하는 부분 중 어느 부분이 테스트가 수행이 되었는지를 체크하는 과정이다. 개발 과정에서의 테스트 커버리지 분석은 일반적으로 코드 커버리지로 분석한다. 코드 커버리지만 테스트가 전체 소스 코드 중 어느 부분을 수행했는지를 검토하는 것이다. 코드 커버리지를 측정할 때 가장 중요한 것은 목표 커버리지율을 결정하는 것이다. 코드 100%를 테스트하는 것이 좋겠지만 Exception, if 문에 대해서 100% 테스트가 불가능하다. 또한 setter 와 getter 만이 있는 ValueObject 의 경우 테스트를 작성하는 것도 쉽고 테스트 자체가 의미가 없나 Coverage rate 는 많이 올릴 수 있다. 만약 커버리지율을 강제로 높이려고 하면 개발팀에서 테스트가 필요하지 않고 테스트가 쉬운 곳에서만 테스트를 집중할 수 있기 때문에 컴포넌트의 우선순위를 결정해서 중요한 컴포넌트에 대해서 커버리지를 관리하는 것이 필요하다. 커버리지율은 잘 만든 테스트라도 50~70% 내외이고, 고 가용성 시스템도 80%를 넘기 힘들기 때문에 컴포넌트의 중요도별로 목표 커버리지율을 융통성 있게 결정하는 것이 필요하다.

장점

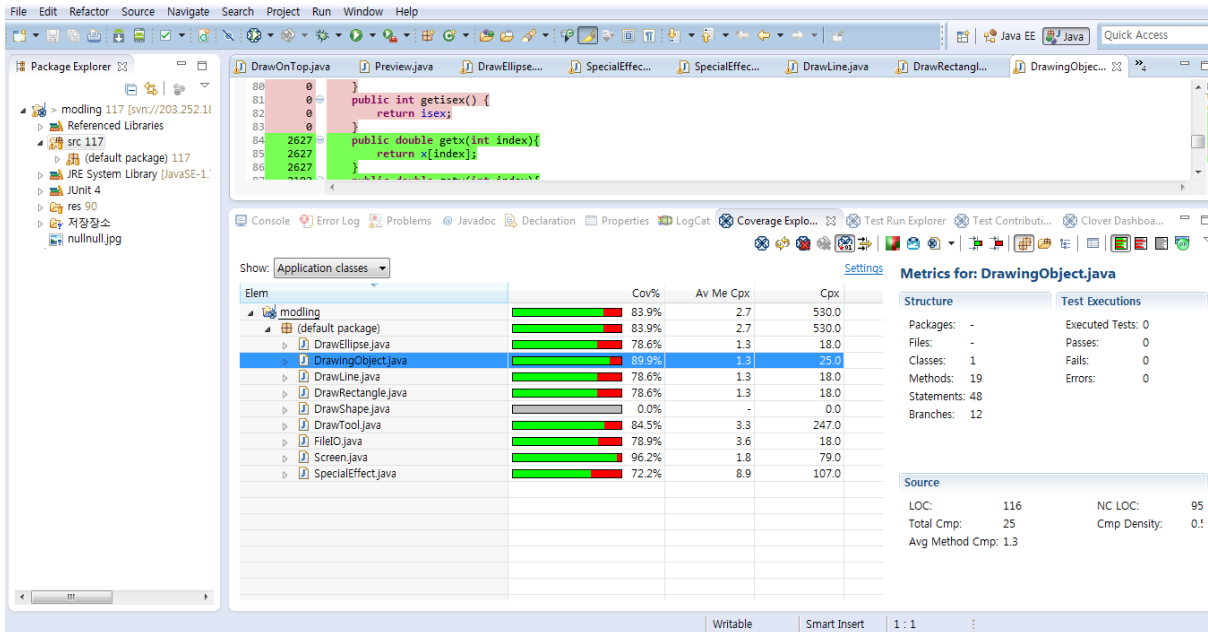
- 기존 소스 코드에서 리팩토링할 부분이 명확히 드러나고 좀 더 나은 설계가 가능
- 소스 코드를 수정할 때 코드 정상 동작 여부나 부작용 확인 가능
- 객체에 대한 개별 검증 가능하여 전체 시스템이 구성되지 않아도 기능 확인이 쉬움
- 실제 서비스 시에 드물게 발생하는 예외적 케이스에 대한 테스트 가능

단점

- 노하우와 지식 부족으로 다양한 경우에 대한 단위 테스트 작성이 힘들
- 테스트 코드 작성에 많은 시각이 소요되어 개발 일정에 무리
- 소스 코드를 변경할 때 테스트 코드도 함께 변경해야 함

1.2 Clover 적용 및 분석 (3 조 프로젝트)

1.2.1 Code Coverage 확인



Clover 를 3 조의 프로젝트에서 enable 시킨 후에 프로젝트를 실행시켜 System Test Case 206 가지를 모두 수행하여 코드 커버리지율을 확인.

수행될 수 없는 Exception (예외 처리 코드), if 문장 등이 존재하기 때문에 우리는 설정되어 있는 UI 를 통해서만 입력을 줄 수 있으므로 예외 상황이 발생할 수 있는 입력이 들어가지 못하는 경우가 있다. 이럴 경우 해당 코드 영역은 아예 진행될 수 없다. 혹은 우리가 테스트 케이스를 완벽하게 모든 해당 코드 영역을 커버할 수 있도록 잘 추려내지 못한 이유가 있을 수도 있다. Test Case 들이 모두 제대로 추려냈다는 가정 하에 수행되지 못하는 다수의 예외 관련, if 문 관련 코드들이 있을 것으로 예상하고 코드 커버리지율을 70% 이상으로 목표를 잡고 커버리지율을 확인하였다.

확인 결과 83.9%의 total 커버리지율 달성

10 개의 Class 중 SpecialEffect.java Class 가 가장 적은 커버리지율을 달성하였다.

코드를 확인해 본 결과 다른 class 들에 비해 line 수가 적은 반면 광원 효과에 대한 테스트를 네 방향에서 모두 수행하고 결과를 확인했어야 했지만 206 개의 test case 에서는 광원 효과를 한 종류로만 취급하여 테스트 하여 관련 기능의 코드가 모두 cover 되지 않았음을 확인하였다. 이를 통해 광원 효과에 대한 Test case 입력이 잘못되었음을 확인할 수 있었고 수정되어야 할 필요성을 발견할 수 있었다.

<pdf 로 report 한 결과 Branch, stmt, Method coverage 율 확인>

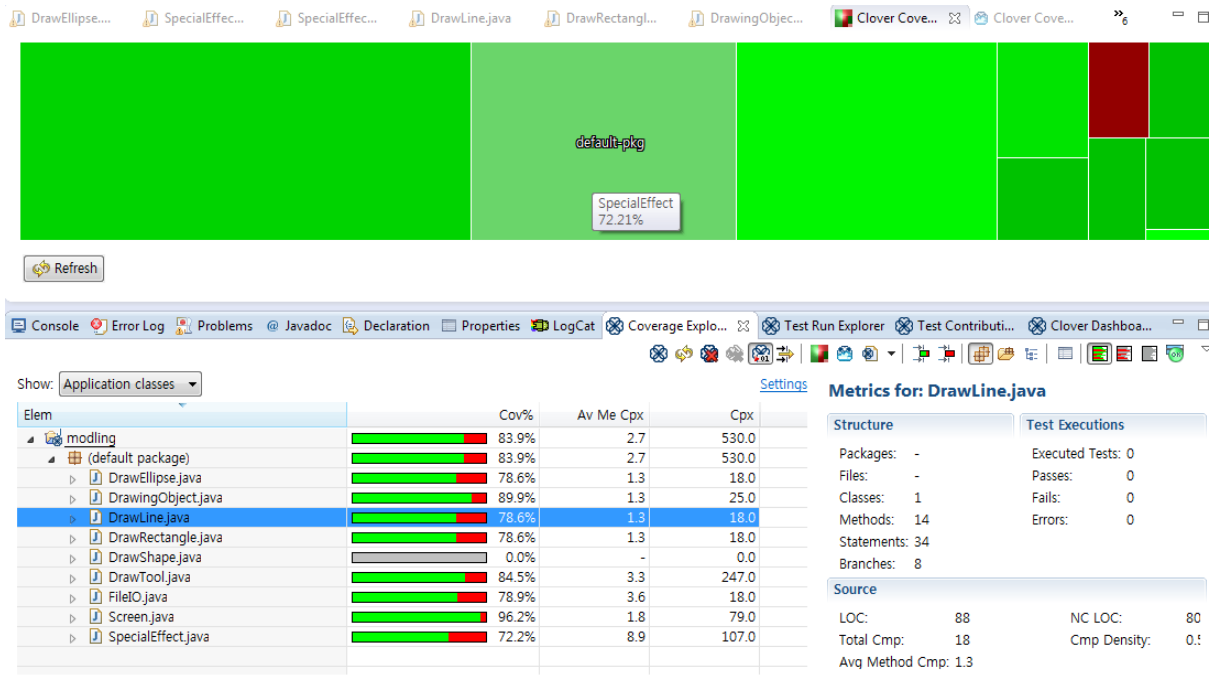
Clover Coverage Report		project stats:	LOC:	2,724	Methods:	197
modling Coverage Report			NCLOC:	2,262	Classes:	10
Coverage timestamp: 6 6 2013 18:57:15 KST			Files:	9	Pkgs:	1
		Branch	Stmt	Method	Total	
Clover database 6 6 2013 17:04:49 KST		71.8%	87.6%	84.8%	83.9%	
Packages		Branch	Stmt	Method	Total	
default-pkg		71.8%	87.6%	84.8%	83.9%	

1.2.2 Treemap Report 확인

트리맵 리포트는 complexity와 code coverage로 클래스들과 패키지 간의 동시 비교가 가능하다. 사각형의 크기는 해당 클래스나 패키지의 complexity를 나타내고, 사각형의 크기가 클수록 complexity가 높다는 것을 알 수 있다.

색깔은 coverage의 레벨을 나타내고, 색깔이 나타내는 의미는 다음과 같다.

- Bright green (most covered)
- Dark green (more coverage)
- Black (around 50% coverage)
- Dark Red (little coverage)
- Bright Red (uncovered)



분석 결과 가장 interface 클래스를 제외하고 클래스들 중에서 가장 커버리지율이 낮고(어두운 green) complexity 가 높은(사각형의 크기가 가장 큰) 클래스는 SpecialEffect 클래스였다.

(이 결과에 대한 분석과 방안은 Could Report 밑에 종합적으로 분석하였습니다.)

1.2.3 Cloud Report 확인

Cloud Report 를 통해서 Clover 는 해당 프로젝트 테스트 수행 후 가장 complex 하지만 가장 덜 커버된 클래스를 강조해주는 역할을 수행한다.

더 크고 더 빨란 클래스일수록 더 많은 리스크를 갖는다는 걸 의미한다.

커버리지율에 따라 색상이 결정되고 평균 메소드 complexity 에 따라서 글씨의 크기가 결정된다.

The screenshot displays the Clover Cloud Report interface. At the top, the project name 'SpecialEffect' is shown in a large, bold font. Below this, there are navigation tabs for 'Project risks' and 'Quick wins'. A 'Project modling' dropdown menu is set to 'modling', and a checkbox 'Include classes from sub-packages' is checked. A 'Refresh' button is also present. The main area shows a table of metrics for 'DrawLine.java'. The table has columns for 'Elem', 'Cov%', 'Av Me Cpx', and 'Cpx'. The 'DrawLine.java' row is highlighted in blue, indicating it is the selected class. To the right of the table, there are two panels: 'Structure' and 'Test Executions'. The 'Structure' panel shows the class hierarchy, and the 'Test Executions' panel shows the number of tests, passes, fails, and errors.

Elem	Cov%	Av Me Cpx	Cpx
modling	83.9%	2.7	530.0
(default package)	83.9%	2.7	530.0
DrawEllipse.java	78.6%	1.3	18.0
DrawingObject.java	89.9%	1.3	25.0
DrawLine.java	78.6%	1.3	18.0
DrawRectangle.java	78.6%	1.3	18.0
DrawShape.java	0.0%	-	0.0
DrawTool.java	84.5%	3.3	247.0
FileIO.java	78.9%	3.6	18.0
Screen.java	96.2%	1.8	79.0
SpecialEffect.java	72.2%	8.9	107.0

Metrics for: DrawLine.java

Structure

Packages: -
Files: -
Classes: 1
Methods: 14
Statements: 34
Branches: 8

Test Executions

Executed Tests: 0
Passes: 0
Fails: 0
Errors: 0

Source

LOC: 88 NC LOC: 80
Total Cmp: 18 Cmp Density: 0.5
Avg Method Cmp: 1.3

프로젝트에 적용한 결과 SpecialEffect 클래스가 가장 큰 font size 와 red color 를 가지므로 가장 높은 risk 를 의미한다. 이는 SpecialEffect 클래스가 다른 클래스들에 비해 월등하게 높은 average method complexity 를 가지며 커버리지율 또한 가장 낮음을 의미한다.

- method complexity : 하나의 메소드에 대한 cyclomatic complexity
- average method complexity : 모든 클래스들의 method complexity 의 합 / 클래스 안의 메소드 수
- Average method complexity 가 높다는 것은 코드가 복잡하다는 뜻이고 에러가 발생할 확률이 높다는 것을 의미한다.
- Cyclomatic complexity 란 한 Unit 내(Method 혹은 Function)의 코드의 복잡성을 측정하는 정량적 지표이다. 복잡도 $v(G) = \text{edges} - \text{nodes} + 2$ 라는 공식으로 구하는데 보통 '분기문 + 1'이라는 공식으로 쉽게 계산할 수 있다. 여기서 논의하는 복잡성이란 바로 분기문의 수이다. 분기문의 수가 많을수록 테스트해야 할 경로가 많아지기 때문에 그만큼 유지보수 비용이 많이 든다.

아래 표는 기존의 에러를 고치려다가 의도치않게 또 다른 오류가 발생할 확률을 나타내는 표다.

Cyclomatic Complexity	또 다른 오류가 생길 확률
1~10	5%
20~30	20%
50 이상	40%
거의 100	60%

표에서 보듯이, Cyclomatic Complexity 수치가 높으면 유지보수성에 아주 안좋은 영향을 끼친다. JSF AV C++ Coding Standard(이하 JSF CS)에서 규정하는 Cyclomatic Complexity 의 기준 값은 20 혹은 그 이하이다. 다만 JSF CS 는 switch 문이 들어간 함수에 대해서 이 제한을 넘을 수 있다고 규정하고 있다. 이는 JSF CS 에서는 Cyclomatic Complexity 라는 메트릭을 Maintainability(유지보수성)과 Readability(가독성) 측면에서 바라보고 있기 때문이다. 보통 switch 문은 3~4 개 이상의 분기(label)을 가진다. 그래서 해당 코드가 이해하기 쉬움에도 불구하고 switch 문이 있는 함수의 경우 복잡도가 높게 나오는 경향이 있다. JSF CS 는 '합리적이고 유지보수 가능한 코드 생산' 목표를 하고 있기에 Cyclomatic Complexity 을 절대적인 잣대로 재지 않는다.

Treemap Report 와 Cloud Report 분석 결과 Risk 가 가장 높게 나온 SpecialEffect Class 의 경우 중첩된 if-else 문을 좀 더 쪼개서 Complexity 를 낮춰주는 것이 좋다고 판단되었다.

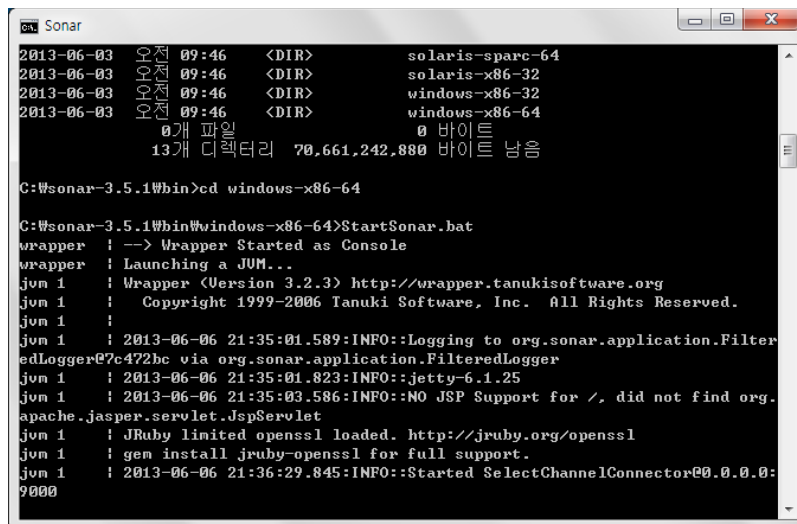
2. Sonar

2.1 Sonar 란?

- Sonar 는 정적분석 도구로 소스코드에 대한 전반적인 품질을 확보할 수 있도록 정보를 제공하는 통합 플랫폼
- Server / Client 구조
- C/C++/Java 등 15 개 이상의 다양한 프로그램 언어 지원
- 플러그인 설치로 다양한 도구와 유연한 통합
- 웹 기반 애플리케이션으로 다양한 결과를 서버에서 통합 관리 용이

2.2 Sonar 수행과정

1. Sonar Server 구동

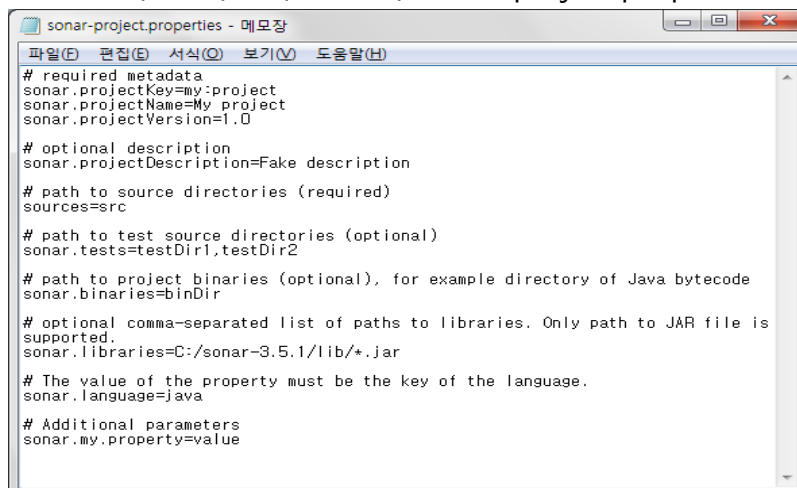


```
2013-06-03 오전 09:46 <DIR> solaris-sparc-64
2013-06-03 오전 09:46 <DIR> solaris-x86-32
2013-06-03 오전 09:46 <DIR> windows-x86-32
2013-06-03 오전 09:46 <DIR> windows-x86-64
0개 파일 0 바이트
13개 디렉터리 70,661,242,880 바이트 남음

C:\#sonar-3.5.1\#bin>cd windows-x86-64

C:\#sonar-3.5.1\#bin\windows-x86-64>StartSonar.bat
wrapper | --> Wrapper Started as Console
wrapper | Launching a JVM...
jvm 1 | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1 |
jvm 1 | 2013-06-06 21:35:01.589:INFO::Logging to org.sonar.application.FilteredLogger@7c472bc via org.sonar.application.FilteredLogger
jvm 1 | 2013-06-06 21:35:01.823:INFO::jetty-6.1.25
jvm 1 | 2013-06-06 21:35:03.586:INFO::NO JSP Support for /, did not find org.apache.jasper.servlet.JspServlet
jvm 1 | JRuby limited openssl loaded. http://jruby.org/openssl
jvm 1 | gem install jruby-openssl for full support.
jvm 1 | 2013-06-06 21:36:29.845:INFO::Started SelectChannelConnector@0.0.0.0:9000
```

2. 프로젝트 최상위 경로에 sonar-project.properties 파일 작성



```
# required metadata
sonar.projectKey=my:project
sonar.projectName=My project
sonar.projectVersion=1.0

# optional description
sonar.projectDescription=Fake description

# path to source directories (required)
sources=src

# path to test source directories (optional)
sonar.tests=testDir1,testDir2

# path to project binaries (optional), for example directory of Java bytecode
sonar.binaries=binDir

# optional comma-separated list of paths to libraries. Only path to JAR file is supported.
sonar.libraries=C:/sonar-3.5.1/lib/*.jar

# The value of the property must be the key of the language.
sonar.language=java

# Additional parameters
sonar.my.property=value
```

3. Sonar Runner 실행


```

C:\Windows\system32\cmd.exe
C:\Users\WJS\Desktop\modling>sonar-runner
C:\Users\WJS\Desktop\modling>sonar-runner
C:\Users\WJS\Desktop\modling>sonar-runner-1.3
Runner configuration file: C:\Users\WJS\Desktop\modling\sonar-runner.properties
Project configuration file: C:\Users\WJS\Desktop\modling\sonar-project.properties

Runner version: 1.3
Java version: 1.7.0_17, vendor: Oracle Corporation
OS name: "Windows 7", version: "6.1", arch: "amd64"
Server: http://localhost:9000
Work directory: C:\Users\WJS\Desktop\modling\W.sonar
21:38:35.160 INFO   .s.b.b.BatchSettings - Load batch settings
21:38:35.660 INFO   o.s.h.c.FileCache - User cache: C:\Users\WJS\sonar\W.cache
21:38:35.784 INFO   atchPluginRepository - Install plugins
21:38:38.171 INFO   .s.b.b.TaskContainer - ----- Executing Project Scan
21:38:39.372 INFO   b.h.JdbcDriverHolder - Install JDBC driver
21:38:39.388 INFO   .b.ProjectExclusions - Apply project exclusions
21:38:39.388 WARN   .c.p.DefaultDatabase - H2 database should be used for evaluation purpose only
21:38:39.388 INFO   o.s.c.p.Database - Create JDBC datasource for jdbc:h2tcp://localhost/sonar
21:38:39.840 INFO   actDatabaseConnector - Initializing Hibernate
21:38:46.268 INFO   .s.h.s.ScanContainer - ----- Inspecting My project
21:38:46.283 INFO   .b.b.ProjectSettings - Load module settings
21:38:48.140 INFO   .s.b.ProfileProvider - Quality profile : Inname-Sonar way.language=java1
21:38:48.202 INFO   s.f.ExclusionFilters - Excluded tests:
21:38:48.202 INFO   s.f.ExclusionFilters - **/package-info.java
21:38:48.280 INFO   nPluginsConfigurator - Configure Maven plugins
21:38:48.654 INFO   org.sonar.INFO - Compare to previous analysis (2013-06-03)
21:38:48.717 INFO   org.sonar.INFO - Compare over 5 days (2013-06-01, analysis of 2013-06-03 13:01:03.486)
21:38:48.764 INFO   org.sonar.INFO - Compare over 30 days (2013-05-07, analysis of 2013-06-03 13:01:03.486)
21:38:48.982 INFO   s.f.FileSystemLogger - Base dir: C:\Users\WJS\Desktop\modling\W.sonar
21:38:48.982 INFO   s.f.FileSystemLogger - Working dir: C:\Users\WJS\Desktop\modling\W.sonar
21:38:48.982 INFO   s.f.FileSystemLogger - Source dirs: C:\Users\WJS\Desktop\modling\src
21:38:48.982 INFO   s.f.FileSystemLogger - Binary dirs: C:\Users\WJS\Desktop\modling\binDir
21:38:48.982 INFO   s.f.FileSystemLogger - Source encoding: UTF-8, default locale: ko_KR
  
```

```

C:\Windows\system32\cmd.exe
21:39:06.269 INFO   s.c.c.ScanGraphStore - Persist graphs of components
21:39:06.360 INFO   .h.p.UpdateStatusJob - ANALYSIS SUCCESSFUL, you can browse http://localhost:9000
21:39:06.360 INFO   h.p.PostJobsExecutor - Executing post-job class org.sonar.plugins.core.batch.IndexProjectPostJob
21:39:06.438 INFO   h.p.PostJobsExecutor - Executing post-job class org.sonar.plugins.dbcleaner.ProjectPurgePostJob
21:39:06.454 INFO   .p.d.p.KeepOneFilter - -> Keep one snapshot per day between 2013-05-09 and 2013-06-05
21:39:06.454 INFO   .p.d.p.KeepOneFilter - -> Keep one snapshot per week between 2012-06-07 and 2013-05-09
21:39:06.454 INFO   .p.d.p.KeepOneFilter - -> Keep one snapshot per month between 2008-06-12 and 2012-06-07
21:39:06.454 INFO   .d.p.DeleteAllFilter - -> Delete data prior to: 2008-06-12
21:39:06.478 INFO   o.s.c.purge.PurgeDao - -> Clean My project fid=1
21:39:06.478 INFO   o.s.c.purge.PurgeDao - -> Clean snapshot 8
Total time: 36.894s
Final Memory: 15M/367M
C:\Users\WJS\Desktop\modling>
  
```

2.3 Sonar 분석 결과

<p>Lines of code 2,352 2,857 lines 1,382 statements 10 files</p>	<p>Classes 11 1 packages 127 methods 36 accessors</p>	<p>Violations 1,007</p> <p>Rules compliance 42.9%</p> <p> ▲ Blocker 0 ▲ Critical 0 ▲ Major 179 ▼ Minor 807 ▼ Info 21 </p>																									
<p>Hotspots by Lines of code</p> <table border="1"> <tr><td>DrawTool</td><td>902</td></tr> <tr><td>Screen</td><td>524</td></tr> <tr><td>SpecialEffect</td><td>402</td></tr> <tr><td>DrawingObject</td><td>95</td></tr> <tr><td>SpecialEffectTest</td><td>90</td></tr> </table>	DrawTool	902	Screen	524	SpecialEffect	402	DrawingObject	95	SpecialEffectTest	90	<p>Most violated rules</p> <table border="1"> <tr><td>Magic Number</td><td>671</td></tr> <tr><td>Trailing Comment</td><td>126</td></tr> <tr><td>If Else Stmt's Must Use Braces</td><td>30</td></tr> <tr><td>Visibility Modifier</td><td>24</td></tr> <tr><td>Avoid Duplicate Literals</td><td>16</td></tr> </table>	Magic Number	671	Trailing Comment	126	If Else Stmt's Must Use Braces	30	Visibility Modifier	24	Avoid Duplicate Literals	16						
DrawTool	902																										
Screen	524																										
SpecialEffect	402																										
DrawingObject	95																										
SpecialEffectTest	90																										
Magic Number	671																										
Trailing Comment	126																										
If Else Stmt's Must Use Braces	30																										
Visibility Modifier	24																										
Avoid Duplicate Literals	16																										
<p>Comments 7.9% 203 lines 0.0% docu. API 154 undocu. API</p>	<p>Duplications 23.1% 659 lines 30 blocks 6 files</p>	<p>Most violated resources</p> <table border="1"> <tr><td>DrawTool</td><td>▲0</td><td>▲73</td><td>▼457</td><td>▼4</td></tr> <tr><td>Screen</td><td>▲0</td><td>▲44</td><td>▼152</td><td>▼0</td></tr> <tr><td>SpecialEffect</td><td>▲0</td><td>▲34</td><td>▼138</td><td>▼4</td></tr> <tr><td>FileIO</td><td>▲0</td><td>▲20</td><td>▼6</td><td>▼0</td></tr> <tr><td>SpecialEffectTest</td><td>▲0</td><td>▲5</td><td>▼31</td><td>▼2</td></tr> </table>	DrawTool	▲0	▲73	▼457	▼4	Screen	▲0	▲44	▼152	▼0	SpecialEffect	▲0	▲34	▼138	▼4	FileIO	▲0	▲20	▼6	▼0	SpecialEffectTest	▲0	▲5	▼31	▼2
DrawTool	▲0	▲73	▼457	▼4																							
Screen	▲0	▲44	▼152	▼0																							
SpecialEffect	▲0	▲34	▼138	▼4																							
FileIO	▲0	▲20	▼6	▼0																							
SpecialEffectTest	▲0	▲5	▼31	▼2																							
<p>Complexity 4.1/method 47.5/class 52.3/file Total: 523</p>	<p>Unit tests coverage -</p> <p>Unit test success 0 tests</p>																										
<p>Events</p> <p>2013/06/07 Version 1.0</p> <p>Fake description</p> <p>Key: my.project Language: Java Profile: Sonar way (version 1) Alerts: RSS Feed</p>	<p>Size: Lines of code Color: Rules compliance 0.0% - 100.0%</p>																										

2.4 각 부분 설명

1. 프로젝트에 대한 기본 정보
2. 클래스별 라인 수
3. Comments : 주석 라인 수 / Duplications : 중복된 라인 수
4. Bad Distribution of Complexity
5. No coding standards and Potential bugs
6. Most violated rules
7. Most violated classes
8. Treemap - size:Lines of codes / color:Rules compliance

2.5 분석 결과 도출

1. 뒷장에 첨부한 Major Violations 참고(p11 ~ p36)
2. 중복 코드 refactoring
3. 상세한 주석 요망
4. Trailing comments(//로 시작하는 주석) 자제
5. Magic Number(아무런 설명 없이 등장하는 수치 상수)의 사용 자제

DrawTool

```
37 // @ Author :
38 //
39 //
40 public class DrawTool extends JComponent{
41     Color color;
```

Visibility Modifier | 3 일

Variable 'color' must be private and have accessor methods.

```
42     private String brushShape;
43     private static String figureShape;
44     private int lineThickness;
45     private String state;
```

Unused Private Field | 3 일

Avoid unused private fields such as 'state'.

```
46     private DrawingObject drawobject;
47     private int red, green, blue;
48     final JFrame frame1;
```

Visibility Modifier | 3 일

Variable 'frame1' must be private and have accessor methods.

```
49     private BufferedImage image;
50     public ArrayList<DrawShape> list = new ArrayList<DrawShape>();
```

Visibility Modifier | 3 일

Variable 'list' must be private and have accessor methods.

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```
51     public ArrayList<DrawShape> list2 = new ArrayList<DrawShape>();
```

Visibility Modifier | 3 일

Variable 'list2' must be private and have accessor methods.

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```
52     private int[] fillx = new int[1000]; //fillcolor ?? x,y ??g ??? ???
53     private int[] filly = new int[1000];
54     int count = 0;
```

Visibility Modifier | 3 일

Variable 'count' must be private and have accessor methods.

```
55     public void reset() {
56         color = color.black;
57         brushShape = "normal";
```

Avoid Duplicate Literals | 3 일

The String literal "normal" appears 5 times in this file; the first occurrence is on line 57

```
58     figureShape = "rectangle";
```

Avoid Duplicate Literals | 3 일

The String literal "rectangle" appears 6 times in this file; the first occurrence is on line 58

```
59         lineThickness = 1;
60         count = 0;
61     }
62     public void setFillDecalxy(int x, int y,int width){
86     }
87     public int getcount() {
88         return count;
89     }
90     public void tempaddlist(double sx, double sy, double ex, double ey,String
state){
```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 19 (max allowed is 10).

```
91         if (state.equals("rectangle")) {
92             if (sx <= ex && sy <= ey) {
93                 list2.add(new DrawRectangle(sx, sy, ex - sx, ey -
94                 sy, color, lineThickness));
101             }
102             else if (sx > ex && sy > ey) {
103                 list2.add(new DrawRectangle(ex, ey, sx - ex, sy -
104                 ey, color, lineThickness));
105             }
106             else if (state.equals("circle")) {
```

Avoid Duplicate Literals | 3 일

The String literal "circle" appears 4 times in this file; the first occurrence is on line 105

```
106         if (sx < ex && sy < ey) {
107             list2.add(new DrawEllipse(sx, sy, ex - sx, ey -
108             sy, color, lineThickness));
109             }
110             else if (sx > ex && sy < ey) {
111                 list2.add(new DrawEllipse(ex, ey, sx - ex, sy -
112                 ey, color, lineThickness));
113             }
114             }
115         }
116     public void spBrush2Cal(double sx, double sy, double ex, double ey, int
117     linethickness) {
```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 46 (max allowed is 10).

Ncss Method Count | 3 일

The method spBrush2Cal() has an NCSS line count of 87

```
121         switch (linethickness) {
122             case 1:
123                 if (sx <= ex && sy <= ey) { //1
124                     list.add(new DrawLine(sx + 1, sy - 1, ex + 1, ey -
125                     1, color.black, 1));
126                 }
127             }
128         }
129     public void spBrush1Cal(double sx, double sy, double ex, double ey, int
130     linethickness) {
```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 46 (max allowed is 10).

Ncss Method Count | 3 일

The method spBrush1Cal() has an NCSS line count of 89

```
220         int dividethick = linethickness / 5;
```

Unused local variable | 3 일

Avoid unused local variables such as 'dividethick'.

```
221         int []spbound = {1, 3, 5, 6, 7};
222         switch (linethickness) {
223             case 1:
224                 }
225                 break;
226             }
227         }
228     public void addlist(double sx, double sy, double ex, double ey, String state) {
```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 27 (max allowed is 10).

```
318         if (state.equals("pencil")) {
319             list.add(new DrawLine(sx, sy, ex, ey, color, 1)); } }
```

```

320         else if(state.equals("eraser")){
321             list.add(new
DrawLine(sx, sy, ex, ey, Color.white, lineThickness));
392         frame.setBounds(100, 100, 270, 198);
393         frame.getContentPane().setLayout(null);
394
395         /*
396         JTextPane textPane = new JTextPane();

```

▲ Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```

397         textPane.setBounds(0, 94, 6, 21);
398         frame.getContentPane().add(textPane);
399         */
400         final JTextPane txtpnR = new JTextPane();
433             txtpnB.setText(" B : "+ scrollBar.getValue());
434         }
435     });
436
437     final JScrollBar scrollBar_1 = new JScrollBar(0,0,0,0,255);

```

▲ Local Final Variable Name | 3 일

Name 'scrollBar_1' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```

438         scrollBar_1.setOrientation(JScrollBar.HORIZONTAL);
439         scrollBar_1.setBounds(60, 115, 195, 23);
440         frame.getContentPane().add(scrollBar_1);
441
442             txtpnG.setText(" G : "+ scrollBar_1.getValue());
450         }
451     });
452
453     final JScrollBar scrollBar_2 = new JScrollBar(0,0,0,0,255);

```

▲ Local Final Variable Name | 3 일

Name 'scrollBar_2' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```

454         scrollBar_2.setOrientation(JScrollBar.HORIZONTAL);
455         scrollBar_2.setBounds(60, 94, 195, 23);
456         frame.getContentPane().add(scrollBar_2);
457         scrollBar_2.addAdjustmentListener(new AdjustmentListener() {
470             JButton btnNewButton = new JButton("OK");
471             btnNewButton.setBounds(200, 0, 70, 93);
472             frame.getContentPane().add(btnNewButton);
473             btnNewButton.addActionListener(new ActionListener() {
474                 @SuppressWarnings("deprecation")

```

▲ Avoid Duplicate Literals | 3 일

The String literal "deprecation" appears 19 times in this file; the first occurrence is on line 474

```

475         public void actionPerformed(ActionEvent arg0) {
476             color = new Color(red,green,blue);
477             frame1.enable();
478
480         }
481     });
482
483
484     frame.addWindowListener(new WindowListener(){

```

▲ Anon Inner Length | 3 일

Anonymous inner class length is 42 lines (max allowed is 20).

```

485         @Override
486         public void windowActivated(WindowEvent arg0) {
487
488
554     });
555     btnNewButton.setBounds(0, 0, 126, 23);
556     frame.getContentPane().add(btnNewButton);
557
558     JButton btnNewButton_1 = new JButton("special1");

```

▲ Local Variable Name | 3 일

Name 'btnNewButton_1' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```

559         btnNewButton_1.addActionListener(new ActionListener() {
560             @SuppressWarnings("deprecation")
561             public void actionPerformed(ActionEvent arg0) {
562                 brushShape = "special1";
580             }
581         });
582         btnSpecial.setBounds(0, 66, 126, 23);
583         frame.getContentPane().add(btnSpecial);
584         frame.addWindowListener(new WindowListener() {

```

Anon Inner Length | 3 일

Anonymous inner class length is 42 lines (max allowed is 20).

```

585
586             @Override
587             public void windowActivated(WindowEvent arg0) {
588
643             }
644         });
645
646         btnNewButton.setBounds(0, 0, 33, 33);
647         if(figureShape == "rectangle")

```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```

648         {
649             btnNewButton.setIcon(new ImageIcon("res\\select_rec.png"));
650         }
651         else
652             btnNewButton.setIcon(new ImageIcon("res\\rectan.jpg"));

```

If Else Stmtns Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

653
654         frame.getContentPane().add(btnNewButton);
655
656         JButton btnNewButton_1 = new JButton("");

```

Local Variable Name | 3 일

Name 'btnNewButton_1' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```

657         btnNewButton_1.addActionListener(new ActionListener() {
658             @SuppressWarnings("deprecation")
659             public void actionPerformed(ActionEvent arg0) {
660                 figureShape = "circle";
661                 frame.setVisible(false);
664             }
665         });
666         btnNewButton_1.setBounds(35, 0, 33, 33);
667         if(figureShape=="circle"){

```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```

668             btnNewButton_1.setIcon(new
669             ImageIcon("res\\select_circle.png"));
670         }
671         else
672             btnNewButton_1.setIcon(new ImageIcon("res\\circle.jpg"));

```

If Else Stmtns Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

672         frame.getContentPane().add(btnNewButton_1);
673
674         frame.addWindowListener(new WindowListener() {

```

Anon Inner Length | 3 일

Anonymous inner class length is 42 lines (max allowed is 20).

```

675
676             @Override
677             public void windowActivated(WindowEvent arg0) {
678
748             }

```

```
749         });
750         btnNewButton.setBounds(0, 28, 120, 25);
751         if(lineThickness == 5)
752             btnNewButton.setIcon(new ImageIcon("res\\Line_5_sel.png"));
```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
753         else
754             btnNewButton.setIcon(new ImageIcon("res\\Line_5.png"));
```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
755
756         frame10.getContentPane().add(btnNewButton);
757
758         JButton btnNewButton_1 = new JButton("");
```

Local Variable Name | 3 일

Name 'btnNewButton_1' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```
759         btnNewButton_1.addActionListener(new ActionListener() {
760             @SuppressWarnings("deprecation")
761             public void actionPerformed(ActionEvent e) {
762                 lineThickness = 10;
763             }
764         });
765         btnNewButton_1.setBounds(0, 55, 120, 25);
766         if(lineThickness == 10)
767             btnNewButton_1.setIcon(new
770         ImageIcon("res\\Line_10_sel.png"));
```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
771         else
772             btnNewButton_1.setIcon(new ImageIcon("res\\Line_10.png"));
```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
773
774         frame10.getContentPane().add(btnNewButton_1);
775
776         JButton btnNewButton_2 = new JButton("");
```

Local Variable Name | 3 일

Name 'btnNewButton_2' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```
777         btnNewButton_2.addActionListener(new ActionListener() {
778             @SuppressWarnings("deprecation")
779             public void actionPerformed(ActionEvent e) {
780                 lineThickness = 1;
781             }
782         });
783         btnNewButton_2.setBounds(0, 1, 120, 25);
784         if(lineThickness == 1)
785             btnNewButton_2.setIcon(new ImageIcon("res\\Line_1_sel.png"));
```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
789         else
790             btnNewButton_2.setIcon(new ImageIcon("res\\Line_1.png"));
```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
791
792         frame10.getContentPane().add(btnNewButton_2);
793
794         JButton btnNewButton_3 = new JButton("");
```

Local Variable Name | 3 일

Name 'btnNewButton_3' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```
795         btnNewButton_3.addActionListener(new ActionListener() {
```

```

796             @SuppressWarnings("deprecation")
797             public void actionPerformed(ActionEvent e) {
798                 lineThickness = 15;
802             }
803         });
804         btnNewButton_3.setBounds(0, 82, 120, 25);
805         if(lineThickness == 15)
806             btnNewButton_3.setIcon(new
ImageIcon("res\\Line_15_sel.png"));

```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

807         else
808             btnNewButton_3.setIcon(new ImageIcon("res\\Line_15.png"));

```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

809
810             frame10.getContentPane().add(btnNewButton_3);
811
812             JButton btnNewButton_4 = new JButton("");

```

Local Variable Name | 3 일

Name 'btnNewButton_4' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```

813             btnNewButton_4.addActionListener(new ActionListener() {
814                 @SuppressWarnings("deprecation")
815                 public void actionPerformed(ActionEvent e) {
816                     lineThickness = 20;
820                 }
821             });
822             btnNewButton_4.setBounds(0, 109, 120, 25);
823             if(lineThickness == 20)
824                 btnNewButton_4.setIcon(new
ImageIcon("res\\Line_20_sel.png"));

```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

825         else
826             btnNewButton_4.setIcon(new ImageIcon("res\\Line_20.png"));

```

If Else Stmts Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

827             frame10.getContentPane().add(btnNewButton_4);
828
829
830             frame10.addWindowListener(new WindowListener(){

```

Anon Inner Length | 3 일

Anonymous inner class length is 42 lines (max allowed is 20).

```

831
832             @Override
833             public void windowActivated(WindowEvent arg0) {
834
879                 Robot robot = null;
880                 try {
881                     robot = new Robot();
882                 } catch (AWTException e) {
883                     e.printStackTrace();

```

Avoid Print Stack Trace | 3 일

Avoid printStackTrace(); use a logger call instead.

```

884                 }
885                 BufferedImage outImage = new
BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
886                 outImage=robot.createScreenCapture(imageRec);
887                 //y = y - 3;

```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```

888                 int k = outImage.getRGB(x, y);

```



```
889     int red = (k >> 16) & 0xff;
```

Hidden Field | 3 일

'red' hides a field.

```
890     int green = (k >> 8) & 0xff;
```

Hidden Field | 3 일

'green' hides a field.

```
891     int blue = (k) & 0xff;
```

Hidden Field | 3 일

'blue' hides a field.

```
892     setcolor(new Color(red,green,blue));
```

```
893     }
894     public class temp{
895         double x;
```

Visibility Modifier | 3 일

Variable 'x' must be private and have accessor methods.

```
898         double y;
```

Visibility Modifier | 3 일

Variable 'y' must be private and have accessor methods.

```
899         public double getsx(){
900             return x;
901         }
902         public double getsy() {
903             this.x = x;
904             this.y = y;
905         }
906     }
907     public BufferedImage filldecalcolor(int panelx, int pany, int width, int
908     height, Color c) {
```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 14 (max allowed is 10).

```
911     ArrayList<temp> list = new ArrayList<temp>();
```

Hidden Field | 3 일

'list' hides a field.

```
912     Rectangle imageRec = new Rectangle(panelx,pany+38,width,height);
913     //????? ???? ????
914     Robot robot = null;
915     try {
916         robot = new Robot();
917     } catch (AWTException e) {
918         e.printStackTrace();
```

Avoid Print Stack Trace | 3 일

Avoid printStackTrace(); use a logger call instead.

```
918     }
919     BufferedImage outImage = new
920     BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
921     outImage=robot.createScreenCapture(imageRec);
922     for(int i = 0;i<count;i++) {
923
924         int color = outImage.getRGB(fillx[i],filly[i]);
```

Hidden Field | 3 일

'color' hides a field.

```
925
926         list.add(new temp((double)fillx[i],(double)filly[i]));
927         int flag = 0;
```

Unused local variable | 3 일

Avoid unused local variables such as 'flag'.

```

928
929         while(true){
930             if(color == c.getRGB()) {
931                 break;
954             }
955         }
956         list.remove(0);
957     }
958     else break;

```

▲ **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces

```

959         }
960     }
961     return outImage;
962 }
963
964 //public BufferedImage decalImage(int start x, int start y, int width, int
height){

```

▲ **Avoid commented-out lines of code** | 3 일

This block of commented-out lines of code should be removed.

```

965     public BufferedImage decalImage(JPanel panel, int width, int height){
966         Rectangle imageRec = new Rectangle(panel.getLocationOnScreen().x,
panel.getLocationOnScreen().y+38, width, height); //????? 0?? ?,??? κ??
967         Robot robot = null;
968         try {
969             robot = new Robot();
970         } catch (AWTException e) {
971             e.printStackTrace();

```

▲ **Avoid Print Stack Trace** | 3 일

Avoid printStackTrace(); use a logger call instead.

```

972     }
973     BufferedImage outImage = new
BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
974     outImage=robot.createScreenCapture(imageRec);
975
976     {
977         for(int j = 0; j<height; j++)
978         {
979             if(outImage.getRGB((width/2) - i, j) <=
980 outImage.getRGB((width/2) + i, j))
981 outImage.getRGB((width/2) - i, j));

```

▲ **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces

```

982         else
983 outImage.getRGB((width/2) + i, j));

```

▲ **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces

```

984 outImage.getRGB((width/2) + i, j)) //if(outImage.getRGB((width/2) - i, j) !=

```

▲ **Avoid commented-out lines of code** | 3 일

This block of commented-out lines of code should be removed.

```

985 outImage.getRGB((width/2) - i, j)); //outImage.setRGB((width/2) + i, j,
986     }
987 }
988     for(int i = 0; i<(width/2); i++)
994     }
995     return outImage;
996 }
997
998     public BufferedImage fillColor(int panelx, int panely,int x, int y,int width,
int height,Color c) {

```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 13 (max allowed is 10).

999 `ArrayList<temp> list = new ArrayList<temp>();`

Hidden Field | 3 일

'list' hides a field.

```
1000 //Rectangle imageRec = new Rectangle(panelx+3,panely+86,width,height);
1001 //Rectangle imageRec = new Rectangle(panelx,panely+38,width,height);
1002
1003 Robot robot = null;
1004 try {
1005     robot = new Robot();
1006 } catch (AWTException e) {
1007     e.printStackTrace();

```

Avoid Print Stack Trace | 3 일

Avoid `printStackTrace()`; use a logger call instead.

```
1008 }
1009 BufferedImage outImage = new
1010 BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
1011 outImage=robot.createScreenCapture(imageRec);
1012 int color = outImage.getRGB(x, y);

```

Hidden Field | 3 일

'color' hides a field.

```
1013
1014
1015
1016 list.add(new temp((double)x,(double)y));
1017
1018 int flag = 0;

```

Unused local variable | 3 일

Avoid unused local variables such as 'flag'.

```
1019 while(true){
1020     if(color == c.getRGB()) {
1021         break;
1022     }
1023 }
1024 list.remove(0);
1025 }
1026 else break;

```

If Else Stmt's Must Use Braces | 3 일

Avoid using `if...else` statements without curly braces

```
1050 }
1051 return outImage;
1052 }
1053 public ArrayList<DrawShape> getlist(){

```

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```
1054 return list;
1055 }
1056 public void setlist(ArrayList<DrawShape> list) {

```

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```
1057 this.list = list;
1058 }
1059 public void setlist2(ArrayList<DrawShape> list) {

```

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```
1060         this.list2 = list;
1061     }
1062     public void setImage(BufferedImage image) {
1063         this.image = image;
```

Screen

40
41
42
43
44

```
/**  
 *  
 */  
Graphics g;
```

Visibility Modifier | 3 일

Variable 'g' must be private and have accessor methods.

45

```
Cursor mycursor;
```

Visibility Modifier | 3 일

Variable 'mycursor' must be private and have accessor methods.

46

```
public String state = "pencil"; // pencil pencil
```

Visibility Modifier | 3 일

Variable 'state' must be private and have accessor methods.

Avoid Duplicate Literals | 3 일

The String literal "pencil" appears 7 times in this file; the first occurrence is on line 46

47

```
public double []X; //specialeffect
```

Member name | 3 일

Name 'X' must match pattern '[a-z][a-zA-Z0-9]*\$'.

Visibility Modifier | 3 일

Variable 'X' must be private and have accessor methods.

Naming - Suspicious constant field name | 3 일

The field name indicates a constant but its modifiers do not

48

```
public double []Y;
```

Member name | 3 일

Name 'Y' must match pattern '[a-z][a-zA-Z0-9]*\$'.

Visibility Modifier | 3 일

Variable 'Y' must be private and have accessor methods.

Naming - Suspicious constant field name | 3 일

The field name indicates a constant but its modifiers do not

49

```
private int flagcount = 0; //
```

50

```
private FileIO fileop = new FileIO();
```

51

```
private DrawTool drawtool;
```

52

```
private int sx,sy,ex,ey;
```

53

```
private int count = 0; //
```

54

```
private SpecialEffect specialeffect = new SpecialEffect();
```

55

```
Panel showcolor panel = new Panel();
```

Member name | 3 일

Name 'showcolor_panel' must match pattern '[a-z][a-zA-Z0-9]*\$'.

Visibility Modifier | 3 일

Variable 'showcolor_panel' must be private and have accessor methods.

56

```
private ArrayList<DrawShape> list = new ArrayList<DrawShape>();
```

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

57

```
final JTextPane txtthick = new JTextPane();
```

Visibility Modifier | 3 일

Variable 'txthick' must be private and have accessor methods.

```
58 public JFrame frame = this;
```

Visibility Modifier | 3 일

Variable 'frame' must be private and have accessor methods.

```
59 BufferedImage image3;
```

Member name | 3 일

Name 'image3' must match pattern '[a-z][a-zA-Z0-9]*\$'.

Visibility Modifier | 3 일

Variable 'image3' must be private and have accessor methods.

```
60 int isfillFlag = 0;
```

Visibility Modifier | 3 일

Variable 'isfillFlag' must be private and have accessor methods.

```
61 public JPanel panel;
```

Visibility Modifier | 3 일

Variable 'panel' must be private and have accessor methods.

```
62 Container c;
```

Visibility Modifier | 3 일

Variable 'c' must be private and have accessor methods.

```
63  
64 public Screen() {  
65     setSize(900,600);  
66     drawtool = new DrawTool(frame);  
67     setResizable(false);  
68     initialize();  
}
```

Constructor Calls Overridable Method | 3 일

Overridable method 'resetxy' called during object construction

```
69 }  
70  
71 /**  
72  * Initialize the contents of the frame.  
73  */  
74 private void initialize() {
```

Ncss Method Count | 3 일

The method initialize() has an NCSS line count of 109

```
75     c = getContentPane();  
76  
77     X = new double[10000];  
78     Y = new double[10000];  
115     JToggleButton rbrush = new JToggleButton(""); // ? ? ? ? ?  
116     rbrush.setBounds(68, 0, 32, 32);  
117     rbrush.setIcon(new ImageIcon("res\\brush.png"));  
118     rbrush.addActionListener(new ActionListener() {  
119         @SuppressWarnings("deprecation")
```

Avoid Duplicate Literals | 3 일

The String literal "deprecation" appears 4 times in this file; the first occurrence is on line 119

```
120     public void actionPerformed(ActionEvent arg0) {  
121         frame.disable();  
122         rbrush();  
123         drawtool.showBrushForm();  
167     panel.add(pickColor);  
168  
169  
170  
171     JToggleButton fill_button = new JToggleButton(""); // ? ? ä ? ? ? ? ? ? ? ? ? ?
```

Local Variable Name | 3 일

```
172 Name 'fill_button' must match pattern '^([a-z][a-zA-Z0-9]*)$'.
173 fill_button.setBounds(204,0,32,32);
174 fill_button.setIcon(new ImageIcon("res\\fill.png"));
175 fill_button.addActionListener(new ActionListener() {
176     public void actionPerformed(ActionEvent arg0) {
277
278         showcolor_panel.setBounds(416, 0, 32, 32); //setbounds(xg, yg ,
279         panel.add(showcolor_panel); //
280
281         Panel showthick_panel = new Panel();
```

Local Variable Name | 3 일

```
282 Name 'showthick_panel' must match pattern '^([a-z][a-zA-Z0-9]*)$'.
283 showthick_panel.setBounds(455, 0, 32, 32);
284 panel.add(showthick_panel);
285 showthick_panel.add(txtthick); //
294 JPanel panel3 = new JPanel();
295
296 panel3.setBounds(0, 0, 900, 22);
297 panel3.setLayout(null);
298 JButton open_file = new JButton("load"); //
```

Local Variable Name | 3 일

```
299 Name 'open_file' must match pattern '^([a-z][a-zA-Z0-9]*)$'.
300 open_file.addActionListener(new ActionListener() {
301     public void actionPerformed(ActionEvent arg0) {
302         // init();
```

Avoid commented-out lines of code | 3 일

```
302 This block of commented-out lines of code should be removed.
303 fileop.loadFile(list, drawtool);
304 init1();
305 repaint();
308 open_file.setBounds(0, 0, 97, 20);
309 panel3.add(open_file);
310
311
312 JButton save_file = new JButton("save"); //
```

Local Variable Name | 3 일

```
313 Name 'save_file' must match pattern '^([a-z][a-zA-Z0-9]*)$'.
314 save_file.addActionListener(new ActionListener() {
315     public void actionPerformed(ActionEvent arg0) {
316
317         fileop.saveFile(drawtool,panel.getLocationOnScreen().x,
318         panel.getLocationOnScreen().y);
361         getxy(ex,ey);
362         sx = ex;
363         sy = ey;
364     }
365     else if(state.equals("eraser")) { //
```

Avoid Duplicate Literals | 3 일

```
366 The String literal "eraser" appears 4 times in this file; the first occurrence is on line 365
367 drawtool.addlist(sx, sy, ex, ey, state);
368 getxy(ex,ey);
369 sx = ex;
370 sy = ey;
371 }
372 else if(state.equals("brush")) { //
```

Avoid Duplicate Literals | 3 일

```
372 The String literal "brush" appears 5 times in this file; the first occurrence is on line 371
373 drawtool.addlist(sx, sy, ex, ey, state);
374 getxy(ex,ey);
```

```

374             sx = ex;
375             sy = ey;
376         }
377     else if(state.equals("shape")){

```

Avoid Duplicate Literals | 3 일

The String literal "shape" appears 4 times in this file; the first occurrence is on line 377

```

378             if(drawtool.getFigureShape().equals("rectangle")){
379                 list = new ArrayList<DrawShape>();
380                 drawtool.setlist2(list);
381
382             drawtool.tempaddlist(sx,sy,ex,ey,drawtool.getFigureShape());
392             repaint();
393         }
394
395     public void mouseMoved(MouseEvent arg0) { //??????
396         //drawtool.setBounds(0,60,frame.getWidth()-16,frame.getHeight()-98);

```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```

397         drawtool.setBounds(0,60,900,540);
398         showcolor_panel.setBackground(drawtool.getcolor()); //???
399         txtthick.setText(""+drawtool.getthick());
400         //???
401         drawtool.setlist2(new ArrayList<DrawShape>());
429     public void mousePressed(MouseEvent arg0) {
430         sx = arg0.getX();
431         sy = arg0.getY();
432         if(state.equals("pickcolor")){
433             //drawtool.setpixelColor(getX(),getY(),sx, sy,
drawtool.getWidth(), drawtool.getHeight());

```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```

434         drawtool.setpixelColor(getX(),getY(),sx, sy, 900-6,600-88);
435         showcolor panel.setBackground(drawtool.getcolor());
436     }
437     else if(state.equals("shadow")){ //special effect
438         //???
439     }
477     else if(state.equals("fill")){
478         drawtool.setFillDecalxy(sx, sy,drawtool.getWidth()/2);
479         Image3 =
480 drawtool.fillColor(panel.getLocationOnScreen().x,panel.getLocationOnScreen().y,sx, sy,
drawtool.getWidth(), drawtool.getHeight(),drawtool.getcolor());
BufferedImage Image =
481 drawtool.fillColor(panel.getLocationOnScreen().x,panel.getLocationOnScreen().y,sx, sy,
drawtool.getWidth(), drawtool.getHeight(),drawtool.getcolor());

```

Local Variable Name | 3 일

Name 'Image' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```

482         drawtool.setImage(Image);
483         repaint();
484     }
485 }
488         specialeffect.addlist(X, Y, Color.white, 5, state);
489         this.resetxy();
490     }
491     else if(state.equals("pencil") || state.equals("brush")){
492         //specialeffect.addlist(X, Y, drawtool.getcolor(), 5, state);

```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```

493         specialeffect.addlist(X, Y, drawtool.getcolor(),
drawtool.getthick(), state);
494         this.resetxy();
495     }
496     if(state.equals("line")) {
517         specialeffect.setlist(drawtool.getlist()); //???
518         //list = specialeffect.excutedecal(drawtool.getWidth()/2); //?

```



```
519 list = specialeffect.excutedecal((900-6)/2); //
520
521 drawtool.setlist(list); //
BufferedImage Image = drawtool.decalImage(panel, 900-6, 600-89);
```

Local Variable Name | 3 일

Name 'Image' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```
522 drawtool.setImage(Image);
523 repaint();
524 }
525 public void r() {
593 Toolkit tk = Toolkit.getDefaultToolkit();
594
595 String imgspath =null;
596
597 if(state == "pencil" ){ //
```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```
598 imgspath = "res\\pen_cur.png";
599 xp=0; yp=0; t=true;
600 }
601
602 else if(state == "line" || state == "shape"){ //
```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```
603 imgspath = "res\\line_cur.png";
604 xp=16; yp=16; t=true;
605 }
606 else if(state == "pickcolor") //
```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```
607 {
608 imgspath = "res\\spoid_cur.png";
609 xp=0; yp=0; t=true;
610 }
611 else if(state == "shadow") //
```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```
612 {
613 imgspath = "res\\Light_cur.png";
614 xp=16; yp=16; t=true;
615 }
616 else if(state == "brush")
```

String Literal Equality | 3 일

Literal Strings should be compared using equals(), not '=='.

```
617 {
618 t= true; xp=16; yp=16;
619 imgspath = "res\\brush_cur.png";
620 }
621 else
622 t=false;
```

If Else Stmt's Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
623 if(t==true){
```

Simplify Boolean Expression | 3 일

Expression can be simplified.

```
624         Image cursuring = tk.getImage(imgspath);  
625         mycursor = tk.createCustomCursor(cursuring, new Point(xp,yp),  
626     null );  
626     }  
627     else{
```

SpecialEffect

```
21     private boolean islighton = false; //???? ?
22
23     private static int decalflag = 0; //decal ????.????
24     //????? ????α???? ¥????V?
25     private ArrayList<DrawingObject> list = new ArrayList<DrawingObject>();
26 //????? ???? ?ü?ε? ????
27 Loose coupling | 3 일
28 Avoid using implementation types like 'ArrayList'; use the interface instead
29     private ArrayList<DrawShape> list1 = new ArrayList<DrawShape>(); //?? ? p?
30 ?ü
31 Loose coupling | 3 일
32 Avoid using implementation types like 'ArrayList'; use the interface instead
33     private ArrayList<DrawShape> list2 = new ArrayList<DrawShape>(); //????
34 ???? ?ü
35 Loose coupling | 3 일
36 Avoid using implementation types like 'ArrayList'; use the interface instead
37     public void addlist(double[] sx, double[] sy, Color c, int size, String state){
38         list.add(new DrawingObject(sx,sy,c,size,state));
39     }
40     public void addlist(double sx, double sy,double ex, double ey, Color c, int
41 size, String state,boolean flag){
42
43         if(state.equals("shape")){
44 Avoid Duplicate Literals | 3 일
45 The String literal "shape" appears 6 times in this file; the first occurrence is on line 33
46         if(sx<ex && sy<ey) {
47             double width = ex-sx;
48             double height = ey-sy;
49             list.add(new
50 DrawingObject(sx,sy,sx+width,sy,c,size,"line",flag));
51 Avoid Duplicate Literals | 3 일
52 The String literal "line" appears 27 times in this file; the first occurrence is on line 37
53             list.add(new
54 DrawingObject(sx,sy,sx,sy+height,c,size,"line",flag));
55             list.add(new
56 DrawingObject(sx+width,sy,sx+width,sy+height,c,size,"line",flag));
57             list.add(new
58 DrawingObject(sx,sy+height,sx+width,sy+height,c,size,"line",flag));
59         }
60             list.add(new
61 DrawingObject(ex,ey+height,ex+width,ey+height,c,size,"line",flag));
62         }
63     }
64     else
65         list.add(new DrawingObject(sx,sy,ex,ey,c,size,state,flag));
66 If Else Stmtns Must Use Braces | 3 일
67 Avoid using if...else statements without curly braces
68 }
69     public int getdecalflag(){
70         return decalflag;
71     }
72     public boolean getlighton(){
73         return islighton;
74     }
75     public void setlist(ArrayList<DrawShape> list) {
76 Loose coupling | 3 일
77 Avoid using implementation types like 'ArrayList'; use the interface instead
78     this.list1 = list;
79 }
80
81 }
```

92 public void setlist2(ArrayList<DrawingObject> list) {

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

93 this.list = list;

94 }

95 public ArrayList<DrawShape> excutedecal(int x) { //I X p

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 21 (max allowed is 10).

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

Ncss Method Count | 3 일

The method excutedecal() has an NCSS line count of 78

96 decalflag = 1;
97 list2 = new ArrayList<DrawShape>();
98 ArrayList<DrawingObject> templist = new ArrayList<DrawingObject>();
99 for(DrawingObject i:list){

102 //excute (buffer overflow)

103 templist.add(i);

104 }

105 else { if(i.getstate().equals("pencil") ||
106 i.getstate().equals("brush") || i.getstate().equals("eraser")){

Avoid Duplicate Literals | 3 일

The String literal "brush" appears 6 times in this file; the first occurrence is on line 106

107
108 double[]temp1 = new double[10000];
109 double[]temp2 = new double[10000];
110 double[]temp3 = new double[10000];

151 setlist2(templist);

152
153 for(DrawShape i:list1){
154 if(i.getstate().equals("line")) {
list2.add(new

155 DrawLine(i.getsx(),i.getsy(),i.getex(),i.getey(),i.getcolor(),i.getthickness(),"decal",i.
getshadow()));

Avoid Duplicate Literals | 3 일

The String literal "decal" appears 9 times in this file; the first occurrence is on line 155

156 if(i.getdecal() == 0) {
157 if(i.getsx() > x){
158 double k = i.getsx() - x;
159 double k2 = i.getex() - x;

167 }

168 }

169 }
170 }
171 else if(i.getstate().equals("rectangle")) {

Avoid Duplicate Literals | 3 일

The String literal "rectangle" appears 4 times in this file; the first occurrence is on line 171

172 list2.add(new
DrawRectangle(i.getsx(),i.getsy(),i.getex(),i.getey(),i.getcolor(),i.getthickness(),"deca
l",i.getshadow()));

173
174 if(i.getdecal() == 0) {
175 if(i.getsx() > x){

176 double t = 2*x - i.getex() -

i.getsx();

177 //double t1 = i.getsx() - 2*x -

t;

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```
178 DrawRectangle(t,i.getsy(),i.getex(),i.getey(),i.getcolor(),i.getthickness(),"decal",i.get
    shadow());
179                                     }
180                                     else{
181 i.getsx();
    list2.add(new
183 DrawRectangle(t,i.getsy(),t1,i.getey(),i.getcolor(),i.getthickness(),"decal",i.getshadow(
    ));
184                                     }
185                                     }
186                                     }
187                                     else if(i.getstate().equals("ellipse")) {
```

Avoid Duplicate Literals | 3 일

The String literal "ellipse" appears 4 times in this file; the first occurrence is on line 187

```
188 DrawEllipse(i.getsx(),i.getsy(),i.getex(),i.getey(),i.getcolor(),i.getthickness(),"decal"
    ,i.getshadow());
189
190                                     if(i.getdecal() == 0) {
191                                     if(i.getsx() > x){
192 i.getsx();
    double t = 2*x - i.getex() -
193 t;
    //double t1 = i.getsx() - 2*x -
```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```
194 DrawEllipse(t,i.getsy(),i.getex(),i.getey(),i.getcolor(),i.getthickness(),"decal",i.getsh
    adow());
195                                     }
196                                     else{
197 i.getsx();
    double t = x + x - i.getex() -
204                                     }
205 return list2;
206 }
207
208 public void calDotLine(DrawingObject i,int thick) {
```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 22 (max allowed is 10).

Ncss Method Count | 3 일

The method calDotLine() has an NCSS line count of 74

```
209 if(i.getstate().equals("pencil") || i.getstate().equals("brush")){
210     int tempflag = 0;
211     int flag = 2;
212     double tempx = 0, tempy = 0;
220 + (ty-ty1));
    double temp = Math.abs((tx-tx1)
221                                     if(temp >= thick){
222                                     if(i.getshadowflag())
223                                     {
    if(i.getstate().equals("brush"))
224     list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),i.getthick(),"shadow"));
```


If Else Stmt Must Use Braces | 3 일

Avoid using if...else statements without curly braces

Avoid Duplicate Literals | 3 일

The String literal "shadow" appears 24 times in this file; the first occurrence is on line 224


225 else
226 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),1,"shadow"));

 **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces


227 }
228 else {
229 if(i.getstate().equals("brush"))

230 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),i.getthick()));

 **If Else Stmtns Must Use Braces** | 3 일


Avoid using if...else statements without curly braces

231 } else
232 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),1));

 **If Else Stmtns Must Use Braces** | 3 일


Avoid using if...else statements without curly braces

233 }
234 }
235 else{
236 tempflag = 1;
242 double temp = Math.abs((tempx-
tx1) + (tempy-ty1));
243 if(temp >= thick){
244 if(i.getshadowflag())
245 if(i.getstate().equals("brush"))
246 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),i.getthick(),"shadow"));

 **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces


247 } else
248 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),1,"shadow"));

 **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces


249 }
250 else {
251 if(i.getstate().equals("brush"))

252 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),i.getthick()));

 **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces

253 } else
254 list2.add(new DrawLine(tx,ty,tx+1,ty+1,i.getcolor(),1));

 **If Else Stmtns Must Use Braces** | 3 일

Avoid using if...else statements without curly braces

255 }
256 tempflag = 0;
257 }
258 }
262 flag++;
263 }
264 }
265 else if(i.getstate().equals("eraser")){
266 for(int ch = 0;ch<i.getcurrsize()-1;ch++)

 **For Loops Must Use Braces** | 3 일

Avoid using for statements without curly braces

```

267                                     if(i.getshadowflag()) {
268 DrawLine(i.getx(ch),i.gety(ch),i.getx(ch+1),i.gety(ch+1),Color.white,i.getthick(),"shadow
");
269                                     }
270                                     else {
284     else if(k <= 40) {
285         t = k/(thick+5);
286     }
287     else
288         t = k/(thick+10);

```

If Else Stmt's Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```

289                                     t=t/3;
290                                     double sxx,syy;
291                                     sxx = (ex-sx)/t;
292                                     sy = sy+syy;
303                                     }
304     }
305 }
306
307 public ArrayList<DrawShape> excuteDottedLine() { //????, ??????

```

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```

308     list2 = new ArrayList<DrawShape>();
309     for(DrawingObject i:list){
310         switch(i.getthick()) {
311             case 1:
312             }
313     }
314     return list2;
315 }
316
317 public ArrayList<DrawShape> excuteshadow() {

```

Cyclomatic Complexity | 3 일

Cyclomatic Complexity is 38 (max allowed is 10).

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

Ncss Method Count | 3 일

The method excuteshadow() has an NCSS line count of 132

```

332     list2 = new ArrayList<DrawShape>();
333
334     int []kx;
335     int []ky;

```

FileIO

```
21 import javax.swing.JFrame;  
22 import javax.swing.JOptionPane;
```

```
23  
24 public class FileIO extends JFrame {  
25     public Image imgs;
```

Visibility Modifier | 3 일

Variable 'imgs' must be private and have accessor methods.

```
26     public BufferedImage buf;
```

Visibility Modifier | 3 일

Variable 'buf' must be private and have accessor methods.

```
27     private int loadflag = 0;  
28     public FileIO() {  
29     }  
30     public int getflag() {  
32     }  
33     public void saveFile(DrawTool drawtool,int x, int y)//BufferedImage imgBuffer)  
34     {  
35         //         this.getBounds().x  
36         //Rectangle imageRec = new  
Rectangle(x+8,y+90,drawtool.getWidth(),drawtool.getHeight());
```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```
37         //Rectangle imageRec = new Rectangle(x+3,y+86,900-6,600-89);  
38         Rectangle imageRec = new Rectangle(x,y+38,900-6,600-89);  
39  
40         Robot robot = null;  
41         try {  
42             robot = new Robot();  
43         } catch (AWTException e) {  
44             // TODO Auto-generated catch block  
45             e.printStackTrace();
```

Avoid Print Stack Trace | 3 일

Avoid printStackTrace(); use a logger call instead.

```
46     }  
47     BufferedImage outImage = new  
BufferedImage(drawtool.getWidth(),drawtool.getHeight(),BufferedImage.TYPE_INT_RGB);  
48     outImage=robot.createScreenCapture(imageRec);  
49  
54     String file = save.getFile()+".jpg";  
55  
56     ///////////////////////////////////  
57     try {  
58         //File saveFile = new File(file);
```

Avoid commented-out lines of code | 3 일

This block of commented-out lines of code should be removed.

```
59         File savefile = new File(dir+file);  
60  
61         ImageIO.write(outImage, "jpg", savefile);  
62  
63  
64     } catch (IOException e) {  
65         e.printStackTrace();
```

Avoid Print Stack Trace | 3 일

Avoid printStackTrace(); use a logger call instead.

```
66     }  
67  
68     if(dir == null || file == null) return;
```

If Stmt's Must Use Braces | 3 일

Avoid using if statements without curly braces

```
69     try{  
70
```



```
71         ObjectInputStream SAFILE = new ObjectInputStream((new
BufferedInputStream(new FileInputStream(new File(dir, file)))));
```

Local Variable Name | 3 일

Name 'SAFILE' must match pattern '[a-z][a-zA-Z0-9]*\$'.

```
72             SAFILE.close();
73         }catch(IOException ee){
74         }
75
79         public boolean accept(String name)
80         {
81             boolean isaccpet = false;
82
83             if((name.toLowerCase().endsWith(".jpg")== true)
|| (name.toLowerCase().endsWith(".png")== true) ||
```

Simplify Boolean Expression | 3 일

Expression can be simplified.

Simplify Boolean Expression | 3 일

Expression can be simplified.

```
84         true) || (name.toLowerCase().endsWith(".gif")== true))
            (name.toLowerCase().endsWith(".bmp")==
```

Simplify Boolean Expression | 3 일

Expression can be simplified.

Simplify Boolean Expression | 3 일

Expression can be simplified.

```
85             isaccpet = true;
```

If Else Stmtns Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
86             else
87                 isaccpet = false;
```

If Else Stmtns Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
88
89             return isaccpet;
90
91         }
```

```
92
93
94         public void loadFile(ArrayList<DrawShape> list, DrawTool drawtool)
```

Loose coupling | 3 일

Avoid using implementation types like 'ArrayList'; use the interface instead

```
95         {
96             FileDialog open = new FileDialog(this, "?????", FileDialog.LOAD);
97             open.setVisible(true);
98             String directory = open.getDirectory();
99             String filename = open.getFile();
100             boolean ok = accept(filename);
101
102             if(directory == null || filename ==null)
```

If Stmtns Must Use Braces | 3 일

Avoid using if statements without curly braces

```
103                 return;
104
105                 if(ok == true){
```

Simplify Boolean Expression | 3 일

Expression can be simplified.

```
106                 final File op = new File(directory, filename);
107
```

```
108
109         try{
110
111             buf = ImageIO.read(op);
112             list = new ArrayList<DrawShape>();
```

Parameter Assignment | 3 일

Assignment of parameter 'list' is not allowed.

```
113             drawtool.setlist(list);
114             drawtool.setImage(buf);
115             loadflag = 1;
116             repaint();
117
118         }catch(Exception e)
119         {
120             e.printStackTrace();
```

Avoid Print Stack Trace | 3 일

Avoid printStackTrace(); use a logger call instead.

```
121         }
122     }
123     else
124         JOptionPane.showMessageDialog(null, "???",
        JOptionPane.ERROR_MESSAGE);
```

If Else Stmt's Must Use Braces | 3 일

Avoid using if...else statements without curly braces

```
125     }
126 }
127
```

SpecialEffectTest

```
15 import junit.framework.TestCase;
16
17
18 public class SpecialEffectTest extends TestCase {
19     SpecialEffect special = new SpecialEffect();
```

 **Visibility Modifier** | 3 일


Variable 'special' must be private and have accessor methods.

```
20     DrawTool drawtool = new DrawTool(null);
```

 **Visibility Modifier** | 3 일

Variable 'drawtool' must be private and have accessor methods.


```
21     Screen sc = new Screen();
```

 **Visibility Modifier** | 3 일

Variable 'sc' must be private and have accessor methods.

```
22     public SpecialEffectTest(String name) {
23         super(name);
24     }
```


```
25
26     protected void setUp() throws Exception {
```

 **Useless Overriding Method** | 3 일

Overriding method merely calls super

```
27         super.setUp();
28     }
```

```
29
30     protected void tearDown() throws Exception {
```

 **Useless Overriding Method** | 3 일

Overriding method merely calls super


```
31         super.tearDown();
32     }
```

```
33     @Test
```

```
34     public void testSetlight() {
```

DrawingObject

```
17
18 public class DrawingObject {
19     private String state;
20     private int lineThickness;
21     private String brushShape;
```

 **Unused Private Field** | 3 일


Avoid unused private fields such as 'brushShape'.

```
22     private String figureShape;
```

 **Unused Private Field** | 3 일

Avoid unused private fields such as 'figureShape'.

```
23     private double sx, sy, ex, ey;
24     private double []x;
25     private double []y;
26     private Color color;
36         for(int i =0;i<10000;i++) {
37             x[i] = 0;
38             y[i] = 0;
39         }
40         for(int i = 0;i<getSize(sx);i++){
```

 **Constructor Calls Overridable Method** | 3 일

Overridable method 'getSize' called during object construction

```
41             x[i] = sx[i];
42             y[i] = sy[i];
43         }
44         this.state = state;
```

3. PMD

3.1 PMD 소개

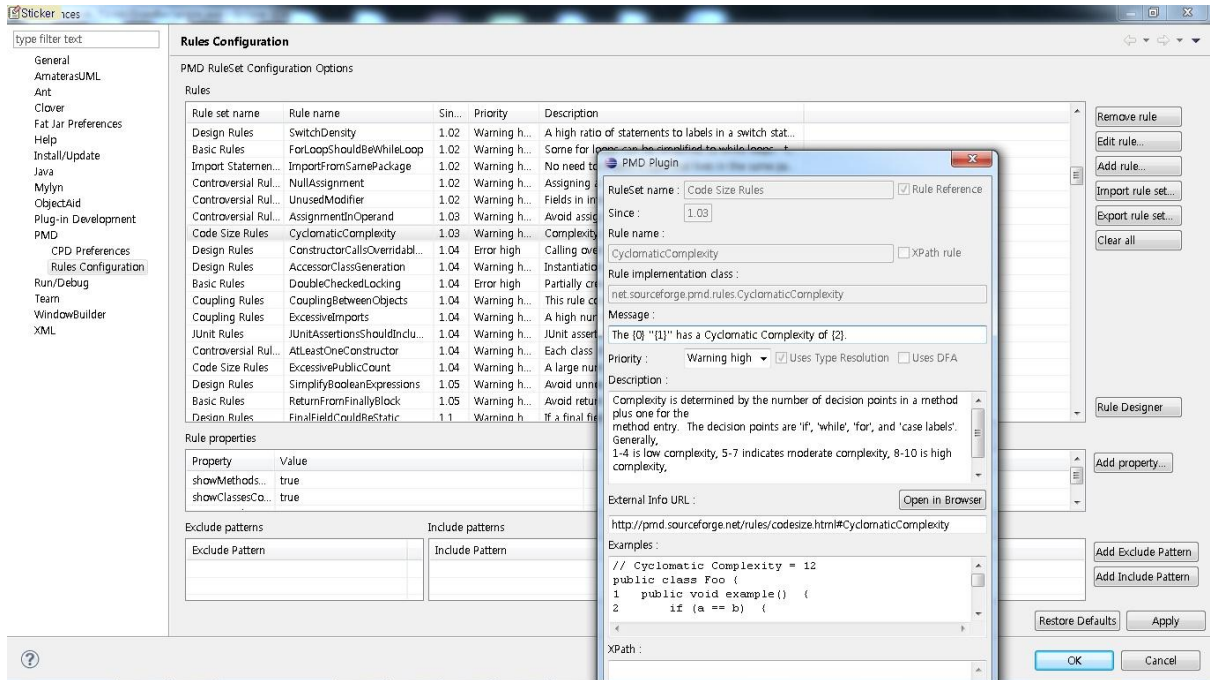
룰 기반의 정적 자바소스 코드 분석기로 CPD(Copy and Paste Detector)를 포함하고 있다. 이클립스 plug-in 형태로 제공되며 Java, JavaScript, XML, XSL 언어에 사용될 수 있지만 CPD 기능의 경우 C, C++ 등의 언어에서도 사용될 수 있다. 지정된 Rule set에 따라 코드를 검사하고 Priority(High Error - Error - High Warning - Warning - Information)에 따라 경고한다. Rule을 사용자가 편집할 수 있으며 Priority도 수정이 가능해서 특정 rule에 대해 엄격히 작업을 수행하고자 한다면 해당 rule을 (high error / error) priority로 설정한다면 그 rule을 위반하여 코딩하였을 경우, 해당 코드가 에러처리 되어 compile이 불가능하다. 이러한 rule 기반의 정적 코드 검사를 통하여 사용자에게 피해야할 특정한 코딩 패턴을 학습할 수 있게 해주며 CPD 기능으로 중복된 코드를 확인하여 리팩토링해야 할 부분에 대한 정보를 제공한다.

- 코드 작성시 피해야 할 패턴들을 학습하게 해준다.
- 기본적으로 세팅되어 있는 룰이 너무 많아 코드 작성의 제약 사항이 많아질 수 있다.

-> 따라서 진행하는 프로젝트 특성에 따라 필요한 룰셋들만을 설정하여 사용하는 것이 좋다.

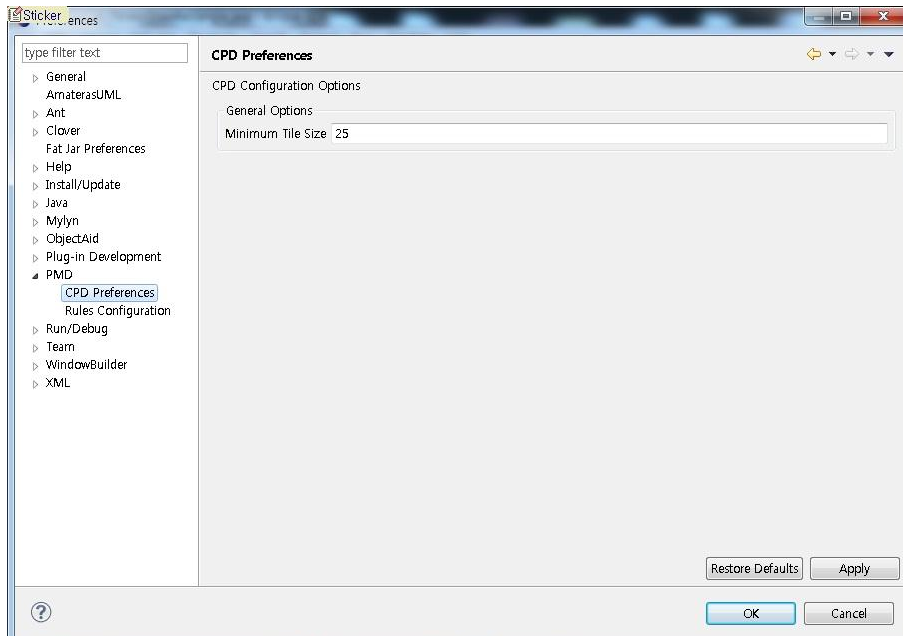
- Rule 기반 검사, 중복된 코드 확인 등으로 refactoring에 유용하게 사용될 수 있는 code inspection tool이다.

3.1.1 Rule Configuration 소개



Windows – preference – PMD - rule configuration에서 사용자가 진행하는 프로젝트 특성에 맞게 Rule들을 추가 / 삭제 / 수정이 가능하다. 각 룰에 대한 priority , rule 이름, 설명을 확인할 수 있다.

3.1.2 CPD Preference



중복코드: 코드가 문자 단위로 모두 일치, 동일한 기능을 하는 코드 CPD를 통해 중복 코드를 확인할 수 있다.

Windows – preference – PMD – CPD preferences를 통해서 CPD configuration option을 설정할 수 있다. 검사하는 최소 tile size를 지정할 수 있다. (tile = token)

Tile은 콤마, identifier 등 토큰을 의미하며 대략적으로 100개의 token은 5~10라인에 해당한다.

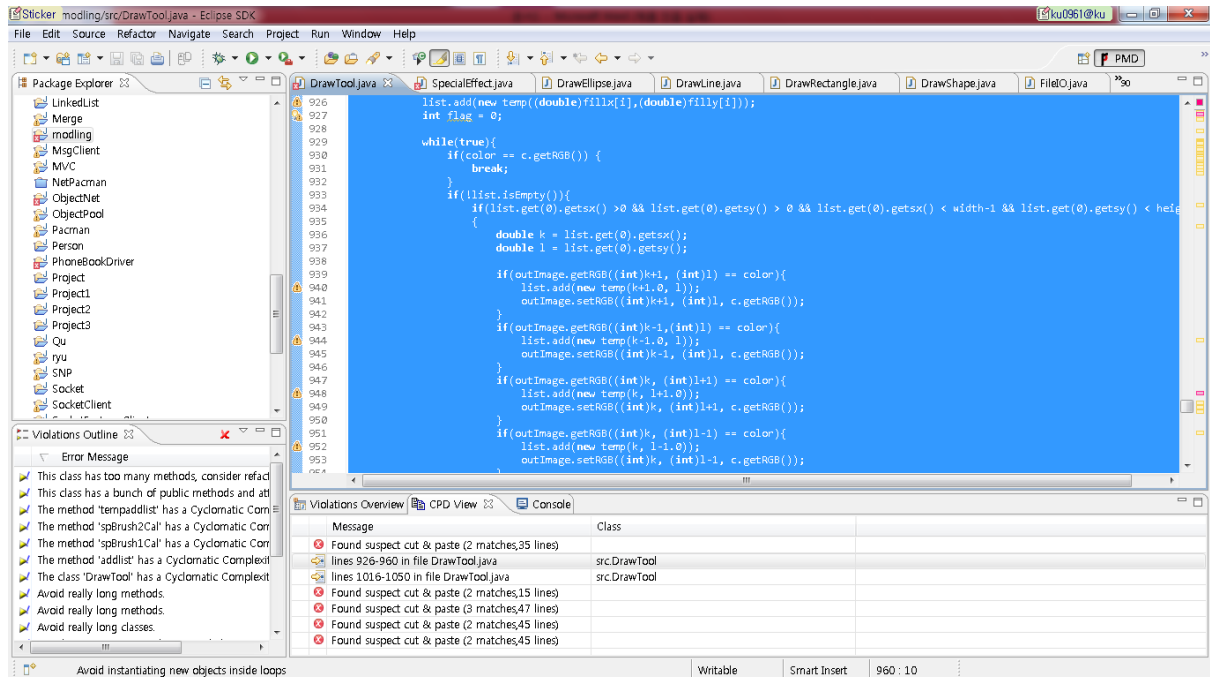
너무 적은 값의 tile을 셋팅할 경우 1line의 중복도 검사결과로 나타나므로 100으로 설정하는 것이 적당하다.

중복 코드는 라인 수에 따라 다음과 같이 구분된다.

High (51 라인이상), Normal (26~50 라인), Low(25 라인이하)

3.2 PMD 적용 및 분석 (3조의 프로젝트)

3.2.1 PMD – CPD 적용 / 분석



Minimum tile size : 100로 설정하여 CPD 수행

프로젝트 전체에서 총 다섯 개의 suspect Cut & Paste 메시지 발견되었습니다.

각각 35 lines 2match, 15 lines 2match, 47 lines 3match, 45 lines 2match, 45 lines 2match 중복이 발생된 부분들로 너무 많은 양의 코드가 중복되었음을 확인할 수 있었습니다.

이렇게 많은 부분에서 중복적으로 사용된 코드를 확인하고 메소드로 변환시켜서 코드를 줄일 수 있고 효율적으로 수정시킬 수 있습니다.

3.2.2 PMD – Rule based code check / 분석

The screenshot shows the Eclipse IDE with the PMD plugin. The main editor displays the source code of the `SpecialEffect` class. Below the editor, the 'Violations Overview' table is visible, showing a total of 520 violations. The 'Violations Outline' pane on the left lists specific rule violations, such as 'The class 'SpecialEffect' has a Cyclomatic Complexity of 10 (highest = 38)' and 'Found non-transient, non-static member...'. A red box highlights the 'Violations Overview' table and the 'Violations Outline' pane.

# Violations	# Violations/LOC	# Violati...	Project
520	207.9 / 1000	3.25	modling

소프트웨어 모델링 수업 3 조 프로젝트에 적용 결과 520 개의 violation 탐지.
(기본적으로 설정되어 있는 rule set 들이 너무 많다.)

Rule 에 기반한 violation 들을 표시해준다.

- 기본적으로 불필요한 rule들이 너무 많이 설정돼 필요하다고 판단되는 몇 가지 룰만 적용
- (Basic Rule, Naming Rule, Code Size Rule, Optimization Rule 모두 세부 rule의 대분류 이름으로 각 룰에는 수십 가지의 세부 rule들이 존재함)

- 적용된 룰

1. Basic Rules

(ex) empty catch block과 같은 코드가 있을 경우 알려주는 Rule

2. Naming Rules

(ex)클래스 안에 생성자가 아닌 메소드에 그 클래스의 이름과 같은 이름으로 선언 시 warning을 발생해주는 naming과 관련된 Rule

3. Code size Rules

(ex)CyclomaticComplexity 확인, 너무 많은 메소드 사용 시 경고 등의 Rule

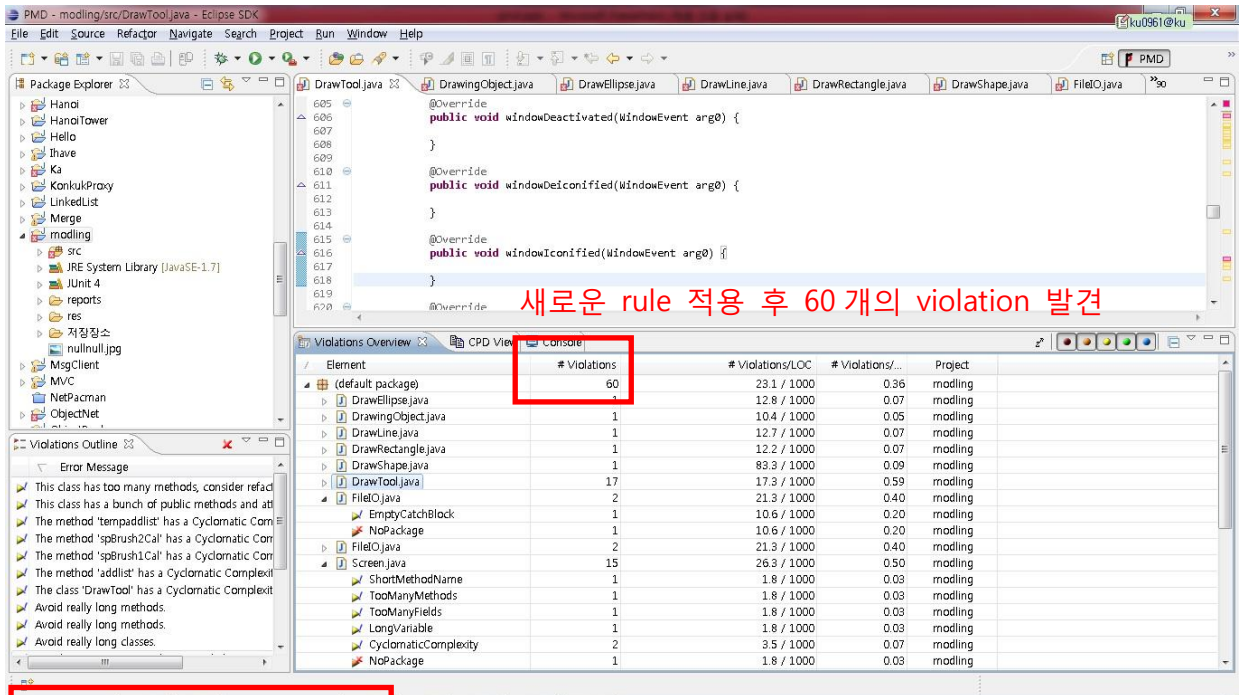
4. Optimization Rules

(ex)empty string을 더한 경우와 같은 rule

```
String s = "" + 123; // bad
```

```
String t = Integer.toString(456); // ok
```

<새로운 Rule 적용 후 violation 검출 결과>



DrawTool.java	17	17.3 / 1000	0.59	modling
FileO.java	2	21.3 / 1000	0.40	modling
EmptyCatchBlock	1	10.6 / 1000	0.20	modling
NoPackage	1	10.6 / 1000	0.20	modling
FileO.java	2	21.3 / 1000	0.40	modling
Screen.java	15	26.3 / 1000	0.50	modling
ShortMethodName	1	1.8 / 1000	0.03	modling
TooManyMethods	1	1.8 / 1000	0.03	modling
TooManyFields	1	1.8 / 1000	0.03	modling
LongVariable	1	1.8 / 1000	0.03	modling
CyclomaticComplexity	2	3.5 / 1000	0.07	modling
NoPackage	1	1.8 / 1000	0.03	modling

패키지 이름을 설정하지 않았거나 빈 CatchBlock 사용, 너무 많은 메소드 사용에 대한 경고, 설정한 CyclomaticComplexity 증가에 대한 경고를 해주고 있습니다. 이를 확인하고 수정하면서 더 안정적이고 효율적인 코딩 습관을 갖게 해주는 데 도움이 됩니다.

PMD report

Problems found

#	File	Line	Problem
1	src/DrawEllipse.java	4	All classes and interfaces must belong to a named package
2	src/DrawLine.java	4	All classes and interfaces must belong to a named package
3	src/DrawRectangle.java	5	All classes and interfaces must belong to a named package
4	src/DrawShape.java	4	All classes and interfaces must belong to a named package
5	src/DrawTool.java	1	This class has a bunch of public methods and attributes
6	src/DrawTool.java	40	All classes and interfaces must belong to a named package
7	src/DrawTool.java	40	Avoid really long classes.
8	src/DrawTool.java	40	The class 'DrawTool' has a Cyclomatic Complexity of 9 (Highest = 46).
9	src/DrawTool.java	40	This class has too many methods, consider refactoring it.
10	src/DrawTool.java	90	The method 'tempaddlist' has a Cyclomatic Complexity of 19.
11	src/DrawTool.java	120	The method 'spBrush2Cal' has a Cyclomatic Complexity of 46.
12	src/DrawTool.java	219	The method 'spBrush1Cal' has a Cyclomatic Complexity of 46.
13	src/DrawTool.java	317	The method 'addlist' has a Cyclomatic Complexity of 27.
14	src/DrawTool.java	384	Avoid really long methods.
15	src/DrawTool.java	732	Avoid really long methods.
16	src/DrawTool.java	896	All classes and interfaces must belong to a named package
17	src/DrawTool.java	926	Avoid instantiating new objects inside loops
18	src/DrawTool.java	940	Avoid instantiating new objects inside loops
19	src/DrawTool.java	944	Avoid instantiating new objects inside loops
20	src/DrawTool.java	948	Avoid instantiating new objects inside loops
21	src/DrawTool.java	952	Avoid instantiating new objects inside loops
22	src/DrawingObject.java	18	All classes and interfaces must belong to a named package
23	src/FileIO.java	24	All classes and interfaces must belong to a named package

<Generate Reports 기능을 통해서 Rule 기반 코드 검사를 한 결과를 XML, txt, HTML로 Report>