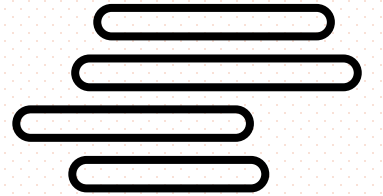


Team Project

T1

Category Partitioning

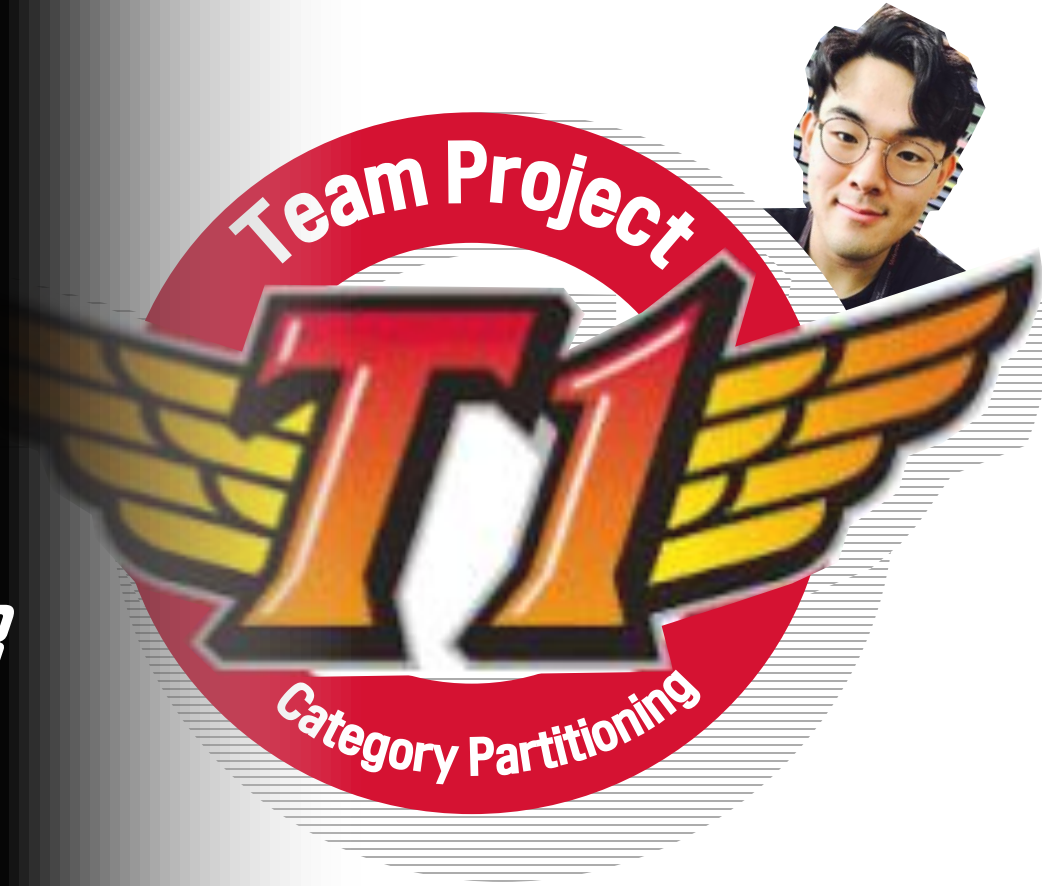
Static
Analysis

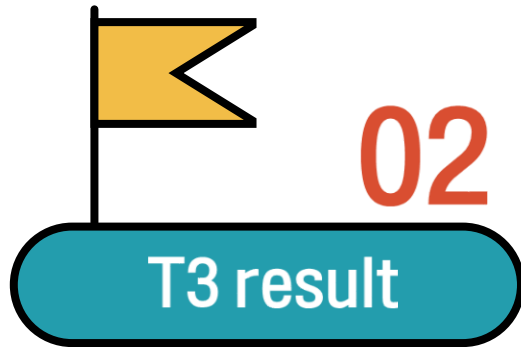
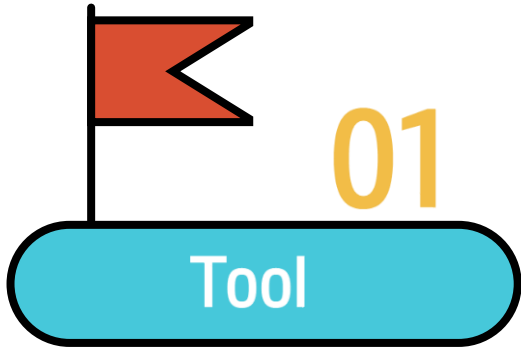
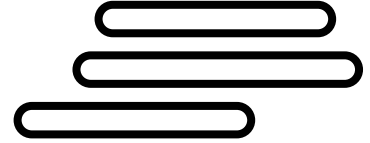
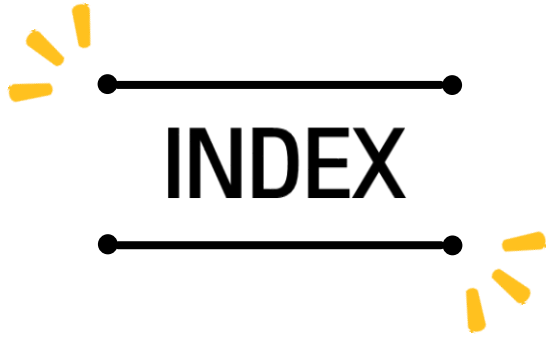
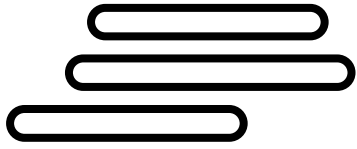


오늘의 발표자

‘김동권’

201312243







Tool

Sonar Qube + Eclipse Plugin



sonarqube 



eclipse
Plug in



metrics

SonarQube

Sonar Qube_Rules

코딩 규칙 품질 프로파일 품질 게이트 환경설정 더보기

Administrator

SMA_staticAnalysis Java

백업하기 Rename Copy 기본값 설정 삭제

코딩 규칙

- 464 개의 규칙 활성화
- 247 Bug
- 10 Vulnerability
- 207 Code Smell

더 많은 규칙 활성화

고유 링크

- [FindBugs](#)
- [PMD](#)
- [Checkstyle](#)

Checkstyle 3.7 installed
Analyze Java code with [Checkstyle](#).

Findbugs 3.4.4 installed
Analyze Java code with [Findbugs](#) 3.0.1.

PMD 2.6 installed
Analyze Java code with [PMD](#).

SonarQube

T3 Result



Code Smell

Bugs

Bugs & 취약점

5 ^C 버그	0 ^A 취약점
----------------------	-----------------------

코드 악취

2.8k ^C 코드 악취	32일 부채
----------------------------	-----------

4일 전에 시작됨

규칙

Indentation	2,411
Line Length	306
Comments Indentation	10
Parameter Name	10
Variable Declaration Usage Distance	4
Malicious code - Field isn't final and can't be	4
Style - Write to static field from instance met	4
Style - Dead store to local variable	2
Methods should not be empty	2
Method Name	1
Bad practice - Method names should start w	1
Bad practice - Method may fail to close strea	1
Style - Return value of method without side e	1
Style - Unused public or protected field	1

검색

규칙

Performance - Method invokes inefficient ne	1
Performance - Method concatenates strings	1
Performance - Should be a static inner class	1
Performance - Unread field	1
Performance - Unused field	1

검색

SonarQube

T3 Result_Code Smell

The screenshot displays the SonarQube interface for a project named 'T3_source'. The left sidebar shows a list of quality rules, with 'Malicious code - Field isn't final and can't be protected from malicious code' selected. The main area shows the source code for 'src/Control/MainSystem.java' with four code smells highlighted in red. Each smell is a 'Code Smell' of 'Info' severity, with a '30min effort' to resolve. The smells are:

- Control.MainSystem.pm isn't final and can't be protected from malicious code
- Control.MainSystem.gm isn't final and can't be protected from malicious code
- Control.MainSystem.nm isn't final and can't be protected from malicious code
- Control.MainSystem.rm isn't final and can't be protected from malicious code

The interface also shows 106 Lines and 82 Issues for the project.

(static) gm, rm, nm, pm 모두 해당. 여러 인스턴스가 조작 될 경우 오류 가능성 존재.

SonarQube

T3 Result_Code Smell

50

```
51 public GUIManager() {
```

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation. ...

40분 전 L51

Code Smell Critical Open 할당되지 않음 5min effort 코멘트

su

52

```
53 }
```

54

```
90 @Override
```

```
91 public void actionPerformed(ActionEvent e) {
```

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation. ...

42분 전 L91

Code Smell Critical Open 할당되지 않음 5min effort 코멘트

suspicious

```
92 }
```

빈 Method





SonarQube

T3 Result_Code Smell

<input checked="" type="checkbox"/> 규칙	
Indentation	2,411
Line Length	306
Comments Indentation	10
Parameter Name	10
Variable Declaration Usage Distance	4
Malicious code - Field isn't final and can't be	4

```
8  
9     public int calculateTestCaseNum(Version ver) {  
10         int errorNum = 0;  
11         int singleNum = 0;  
12         int testcaseNum = 0;  
13     }  
14 }
```

Distance between variable 'testcaseNum' declaration and its first usage is 11, but allowed 3. Consider to make that variable as final if you still need to store its value in advance (before method calls that might do side effect on original value). 20분 전 L12

Code Smell Major Open 할당되지 않음 10min effort 코멘트

변수 선언과 첫번째 사용간의 distance . ->문제가 될 것 같지 않음.



SonarQube

T3 Result_Code Smell

```
41         fis = new FileInputStream(fileChooser.getSelectedFile().toString());
42         ObjectInputStream ois = new ObjectInputStream(fis);
43
44         Project project = (Project)ois.readObject();
45         MainSystem.pm.setProject(project);
46
47         MainSystem.gm.leaderMain.resetProjectPanel();
48         for(int i=0;i<project.getVersionListSize();i++){
```

new GUI.FileChooser(int) may fail to close stream ... 27분 전 L42

Code Smell Major Open 할당되지 않음 1h effort 코멘트 bad

Bad practice - Method may fail to close stream

Code Smell Major bad-practice 다음 기간 이후 적용됨 2017년 5월 30일 상수/이슈: 1h findbugs:OS_OPEN_STREAM

The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.

File stream Close (X).



SonarQube

T3 Result_Code Smell



Variable Declaration Usage Distance	4	53
Malicious code - Field isn't final and can't be	4	54
Style - Write to static field from instance met	4	55
Style - Dead store to local variable	2	56
Methods should not be empty	2	
Method Name	1	
Bad practice - Method names should start w	1	57
Style - Return value of method without side effect is ignored		

```
    }  
    public static void reqSave(){  
        pm.getProject();  
    }
```

Return value of ProjectManager.getProject() ignored, but method has no side effect ... 한시간 전 L56

Code Smell Info Open 할당되지 않음 10min effort 코멘트

Code Smell Info style 다음 기간 이후 적용됨 2017년 5월 30일 상수/이슈: 10min findbugs:RV_RETURN_VALUE_IGNORED_NO_SIDE_EFFECT

This code calls a method and ignores the return value. However our analysis shows that the method (including its implementations in subclasses if any) does not produce any effect other than return value. Thus this call can be removed.

Return 외에 아무 기능을 하지 않음.

SonarQube

T3 Result_Bugs

Bug	5
Vulnerability	0
Code Smell	2,758

해결 상태

Unresolved	5	Fixed	0
False Positive	0	Won't fix	0
Removed	0		

심각도

상태

New Issues

규칙

Performance - Method invokes inefficient ne	1
Performance - Method concatenates strings	1
Performance - Should be a static inner class	1
Performance - Unread field	1
Performance - Unused field	1

검색

Tag

▲ 1 / 5 ▼ 리로드 새로운 검색 대규모 변경

T3_source src/Control/NetworkManager.java

- Unused field: Control.NetworkManager\$ReceiveThreadInNetworkManager.rtms ... 한시간 전 L1 > Bug ▼ Major ▼ Open ▼ 할당되지 않음 ▼ 30min effort 코멘트 performance ▼
- Unread field: Control.NetworkManager.name ... 한시간 전 L15 > Bug ▼ Major ▼ Open ▼ 할당되지 않음 ▼ 30min effort 코멘트 performance ▼
- Should Control.NetworkManager\$ReceiveThreadInNetworkManager be a _static_ inner class? ... 한시간 전 L68 > Bug ▼ Major ▼ Open ▼ 할당되지 않음 ▼ 30min effort 코멘트 performance ▼

T3_source src/Control/ResultManager.java

- Control.ResultManager.recurse(int, int, ArrayList, ArrayList, ArrayList, String, ArrayList, Version) concatenates strings using + in a loop ... 38분 전 L359 > Bug ▼ Major ▼ Open ▼ 할당되지 않음 ▼ 30min effort 코멘트 performance ▼

T3_source src/Model/Version.java

- Model.Version.cloneVersion() invokes inefficient new String(String) constructor ... 한시간 전 L152 > Bug ▼ Major ▼ Open ▼ 할당되지 않음 ▼ 30min effort 코멘트 performance ▼

SonarQube

T3 Result_Bugs

Unresolved	1	Fixed	0	151
False Positive	0	Won't fix	0	152
Removed	0			

```
String con = (String)con1.next();
String cpcon = new String(con);
cprpv.getConstraintsList().add(cpcon);
```

Model.Version.cloneVersion() invokes inefficient new String(String) constructor ... 한시간 전 L152

Bug Major Open 할당되지 않음 30min effort [코멘트](#) performance

Performance - Method invokes inefficient new String(String) constructor

Bug Major performance 다음 기간 이후 적용됨 2017년 5월 30일 상수/이슈: 30min findbugs:DM_STRING_CTOR

Using the `java.lang.String(String)` constructor wastes memory because the object so constructed will be functionally indistinguishable from the `String` passed as a parameter. Just use the argument `String` directly.

String 사용 방식 수정 제안.



SonarQube

T3 Result_Bugs



```
for(int i = 0; i < it; i++)
    tmpresult = tmpresult + " ";
```

Control.ResultManager.recurse(int, int, ArrayList, ArrayList, ArrayList, String, ArrayList, Version) concatenates strings using + in a loop

Bug Major Open 할당되지 않음 30min effort 코멘트 performance

7년 5월 30일 상수/이슈: 30min

findbugs:SBSC_USE_STRINGBUFFER_CONCATENATIO

i in a loop. In each iteration, the String is converted to a StringBuffer/StringBuilder, appended to, and converted back to a String. This can lead to a cost quadratic in t iteration.

·StringBuilder in Java 1.5) explicitly.

루프에서 연결을 사용하여 String을 만드는 것 -> 복잡도 증가

SonarQube 제안 ->

For example:

```
// This is bad
String s = "";
for (int i = 0; i < field.length; ++i) {
    s = s + field[i];
}
```

```
// This is better
StringBuffer buf = new StringBuffer();
for (int i = 0; i < field.length; ++i) {
    buf.append(field[i]);
}
String s = buf.toString();
```



SonarQube

T3 Result_Bugs



```
13 public class NetworkManager {  
14     private Socket socket;  
15     private String name = "";
```

Unread field: Control.NetworkManager.name ...
Bug ▼ Major ▼ Open ▼ 할당되지 않음 ▼ 30min effort 코멘트

한시간 전 ▼ L15 ▼
 performance ▼



Find Bugs

Eclipse Plugin_T3 Result

GUIManager.java: 244

Navigation

Dead store to fc in Control.GUIManager.openFileChooser()
Local variable named fc

Bug: Dead store to fc in Control.GUIManager.openFileChooser()

This instruction assigns a value to a local variable, but the value is not read or used in any subsequent instruction. Often, this indicates an error, because the value computed is never used.

Note that Sun's javac compiler often generates dead stores for final local variables. Because FindBugs is a bytecode-based tool, there is no easy way to eliminate these false positives.

Rank: Of Concern (15), **confidence:** High

Pattern: DLS_DEAD_LOCAL_STORE

Type: DLS, **Category:** STYLE (Dodgy code)

XML output:

```
<BugInstance type="DLS_DEAD_LOCAL_STORE" priority="1" rank="15" abbrev="DLS" category="STYLE" first="1">
  <Class classname="Control.GUIManager">
    <SourceLine classname="Control.GUIManager" sourcefile="GUIManager.java" sourcepath="Control/GUIManager.java"/>
  </Class>
  <Method classname="Control.GUIManager" name="openFileChooser" signature="()V" isStatic="false">
    <SourceLine classname="Control.GUIManager" start="244" end="246" startBytecode="0" endBytecode="8" sourcefile="GUIManager.java" sourcepath="Control/GUIManager.java"/>
  </Method>
  <LocalVariable name="fc" register="1" pc="9" role="LOCAL_VARIABLE_NAMED"/>
  <SourceLine classname="Control.GUIManager" start="244" end="244" startBytecode="8" endBytecode="8" sourcefile="GUIManager.java" sourcepath="Control/GUIManager.java"/>
  <SourceLine classname="Control.GUIManager" start="244" end="244" startBytecode="8" endBytecode="8" sourcefile="GUIManager.java" sourcepath="Control/GUIManager.java"/>
  <Property name="edu.umd.cs.findbugs.detect.DeadLocalStoreProperty.DEAD_OBJECT_STORE" value="true"/>
  <Property name="edu.umd.cs.findbugs.detect.DeadLocalStoreProperty.LOCAL_NAME" value="fc"/>
  <Property name="edu.umd.cs.findbugs.detect.DeadLocalStoreProperty.METHOD_RESULT" value="true"/>
</BugInstance>
```


2 Find Bugs

Eclipse Plugin_T3 Result

MainSystem.java: 22

Navigation

Write to static field Control.MainSystem.gm from instance method new Control.MainSystem()
Field Control.MainSystem.gm

Bug: Write to static field Control.MainSystem.gm from instance method new Control.MainSystem()

This instance method writes to a static field. This is tricky to get correct if multiple instances are being manipulated, and generally bad practice.

Rank: Of Concern (15), **confidence:** High

Pattern: ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD

Type: ST, **Category:** STYLE (Dodgy code)

XML output:

```
<BugInstance type="ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD" priority="1" rank="15" abbrev="ST" category="STYLE" first="1">
  <Class classname="Control.MainSystem">
    <SourceLine classname="Control.MainSystem" sourcefile="MainSystem.java" sourcepath="Control/MainSystem.java"/>
  </Class>
  <Method classname="Control.MainSystem" name="<init>" signature="()V" isStatic="false">
    <SourceLine classname="Control.MainSystem" start="20" end="25" startBytecode="0" endBytecode="106" sourcefile="MainSystem.java" sourcepath="Control/MainSystem.java"/>
  </Method>
  <Field classname="Control.MainSystem" name="gm" signature="LControl/GUIManager;" isStatic="true">
    <SourceLine classname="Control.MainSystem" sourcefile="MainSystem.java" sourcepath="Control/MainSystem.java"/>
  </Field>
  <SourceLine classname="Control.MainSystem" start="22" end="22" startBytecode="21" endBytecode="21" sourcefile="MainSystem.java" sourcepath="Control/MainSystem.java"/>
  <SourceLine classname="Control.MainSystem" start="22" end="22" startBytecode="21" endBytecode="21" sourcefile="MainSystem.java" sourcepath="Control/MainSystem.java"/>
</BugInstance>
```

(static) gm, rm, nm, pm 모두 해당. 여러 인스턴스가 조작 될 경우 오류 가능성 존재.



Metrics

Eclipse Plugin_T3 Result

Cyclomatic Complexity	또 다른오류가생길확률
1~10	5%
20~30	20%
50 이상	40%
거의 100	60%



복잡도 $v(G) = \text{분기문} + 1$
Control package : 42
GUI package : 11
Server package : 9
Model package : 4

Metric	Total	Mean	Std. Dev.	Maximum
▼ McCabe Cyclomatic Complexity (avg/max per method)		2.152	3.83	42
▼ (default package)		2.152	3.83	42
> Control		2.387	5.936	42
> GUI		2.704	2.258	11
> Server		3	2.598	9
> Model		1.204	0.649	4

2

Metrics

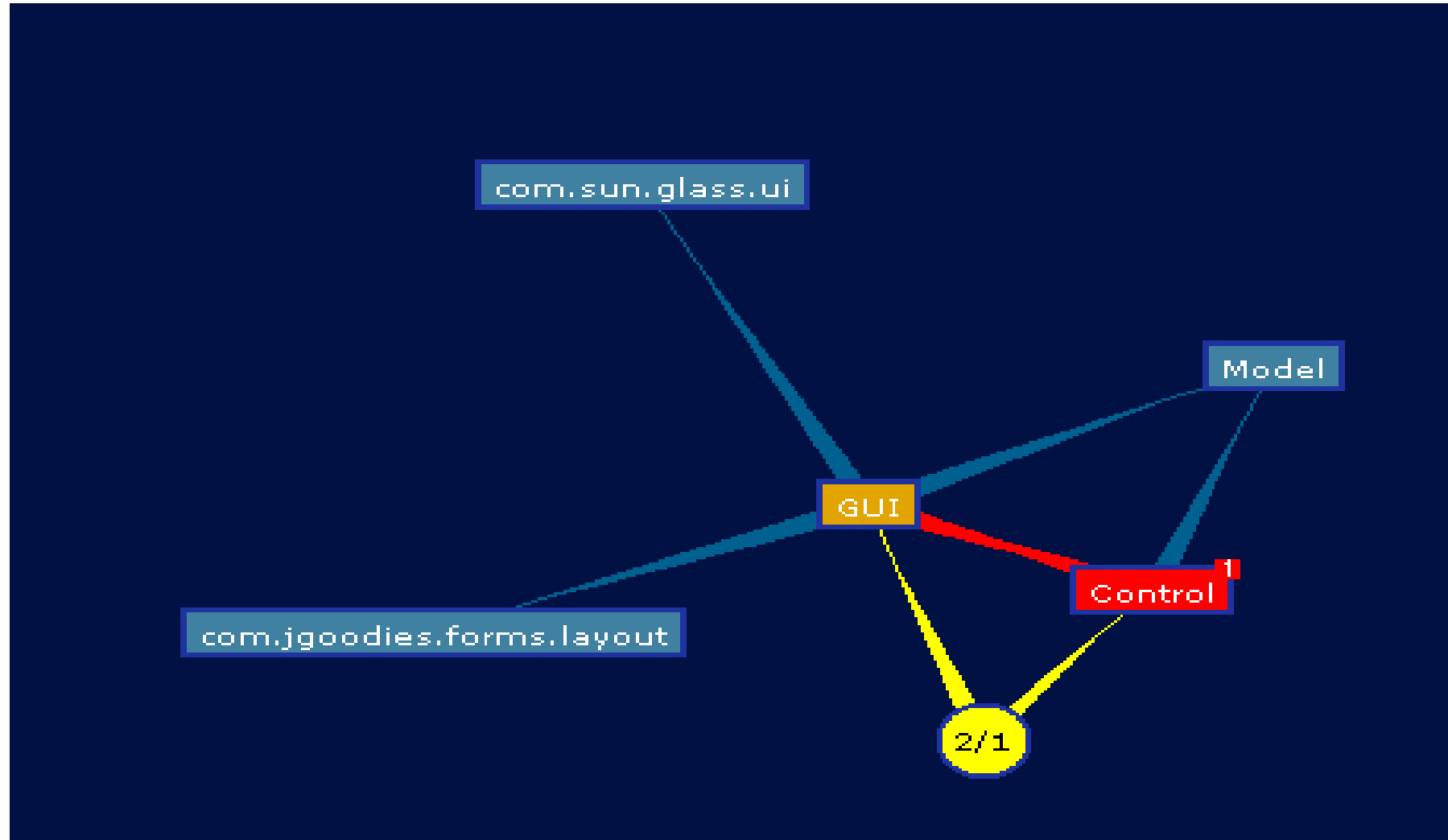
Eclipse Plugin_T3 Result

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		2.152	3.83	42	/T3_source/Control/ResultManager.java	calculateTestCaseNum
> Number of Parameters (avg/max per method)		0.865	0.87	8	/T3_source/Control/ResultManager.java	recurse
> Nested Block Depth (avg/max per method)		1.68	1.148	8	/T3_source/Control/ResultManager.java	recurse
> Afferent Coupling (avg/max per packageFragment)		6.5	6.801	17	/T3_source/Control	
> Efferent Coupling (avg/max per packageFragment)		5.75	6.942	17	/T3_source/GUI	
> Instability (avg/max per packageFragment)		0.551	0.431	1	/T3_source/Server	
> Abstractness (avg/max per packageFragment)		0	0	0	/T3_source/Control	
> Normalized Distance (avg/max per packageFragment)		0.449	0.431	1	/T3_source/Model	
> Depth of Inheritance Tree (avg/max per type)		3.833	2.368	6	/T3_source/GUI/NewProjectView2.java	
> Weighted methods per Class (avg/max per type)	383	12.767	13.431	78	/T3_source/Control/ResultManager.java	
> Number of Children (avg/max per type)	0	0	0	0	/T3_source/Control/GUIManager.java	
> Number of Overridden Methods (avg/max per type)	2	0.067	0.249	1	/T3_source/Control/NetworkManager.java	
> Lack of Cohesion of Methods (avg/max per type)		0.623	0.372	1.333	/T3_source/GUI/MemberVersionPanel.java	
> Number of Attributes (avg/max per type)	189	6.3	4.212	14	/T3_source/Control/GUIManager.java	
> Number of Static Attributes (avg/max per type)	4	0.133	0.718	4	/T3_source/Control/MainSystem.java	
> Number of Methods (avg/max per type)	161	5.367	5.504	24	/T3_source/Control/GUIManager.java	
> Number of Static Methods (avg/max per type)	17	0.567	2.872	16	/T3_source/Control/MainSystem.java	
> Specialization Index (avg/max per type)		0.056	0.212	1	/T3_source/Control/NetworkManager.java	
> Number of Classes (avg/max per packageFragment)	30	7.5	6.225	18	/T3_source/GUI	
> Number of Interfaces (avg/max per packageFragment)	0	0	0	0	/T3_source/Control	
> Number of Packages	4					
> Total Lines of Code	2920					
> Method Lines of Code (avg/max per method)	1936	10.876	21.151	173	/T3_source/Control/ResultManager.java	calculateTestCaseNum



Metric

Eclipse Plugin_T3 Result





SonarQube

Sonar Qube_T4 Result



Bugs & 취약점

7 ^C
버그

0 ^A
취약점

코드 악취

1.9k ^D
코드 악취

24일
부채

5일 전에 시작됨

Code Smell

규칙

Indentation 1,578

Line Length 335

l18n - Reliance on default encoding 8

Multiple Variable Declarations 4

Bad practice - Method may fail to close stream 3

Style - Write to static field from instance method 3

Variable Declaration Usage Distance 2

Method Name 2

Bad practice - Method invokes System.exit(...) 2

Malicious code - May expose internal representation 2

Bad practice - Method names should start with lowercase 2

Bad practice - Method ignores exceptional return value 1

Methods should not be empty 1

검색

Bugs

규칙

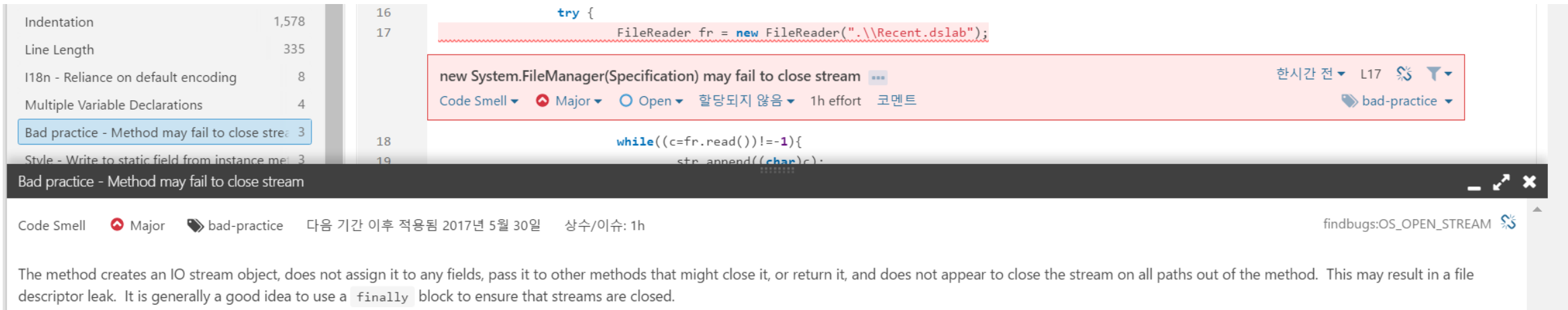
Performance - Method invokes inefficient native method 4

Performance - Method invokes inefficient native method 3

검색

SonarQube

T4 Result_Code Smell



The screenshot displays a code editor with a SonarQube overlay. On the left, a sidebar lists quality rules, with 'Bad practice - Method may fail to close stream' highlighted. The main editor shows a Java code snippet with a red squiggly line under the `new FileReader` call. A tooltip for this issue is shown, indicating it is a 'Major' code smell with a '1h effort' to resolve. The tooltip text reads: 'new System.FileManager(Specification) may fail to close stream'. Below the tooltip, a dark bar contains the text 'Bad practice - Method may fail to close stream'. At the bottom, a detailed description explains that the method creates an IO stream object but does not ensure it is closed on all paths, which can lead to a file descriptor leak. It suggests using a `finally` block to handle this.

```
try {
    16
    17     FileReader fr = new FileReader("\\Recent.dslab");
    new System.FileManager(Specification) may fail to close stream
    Code Smell Major Open 할당되지 않음 1h effort 코멘트 bad-practice
    18
    19     while((c=fr.read())!=-1){
        str.append((char)c);
    }
}
```

Code Smell Major bad-practice 다음 기간 이후 적용됨 2017년 5월 30일 상수/이슈: 1h findbugs:OS_OPEN_STREAM

The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.

File stream Close (X).

SonarQube

T4 Result_Code Smell

심각도
 상태
 New Issues
 규칙

Indentation	1,578
Line Length	335
118n - Reliance on default encoding	8
Multiple Variable Declarations	4
Bad practice - Method may fail to close stre	3

```
23     public RepresentativeValue(String representativeValueName, Hashtable<String, Property> pt) {  
24         this.representativeValueName = new String();  
25         this.propertyConstraint = new ArrayList<Property>();  
26         this.if_propertyConstraint = new ArrayList<Property>();  
27         this.representativeValueName = representativeValueName;  
28         this.propertyTable = pt;  
29         this.priority=1;  
30     }
```

new System.RepresentativeValue(String, Hashtable) may expose internal representation by storing an externally mutable object into RepresentativeValue.propertyTable ... 3일 전 L28

Code Smell Info Open 할당되지 않음 30min effort 코멘트 malicious-code

Malicious code - May expose internal representation by incorporating reference to mutable object

Code Smell Info malicious-code 다음 기간 이후 적용됨 2017년 5월 30일 상수/이슈: 30min findbugs:EI_EXPOSE_REP2

This code stores a reference to an externally mutable object into the internal representation of the object. If instances are accessed by untrusted code, and unchecked changes to the mutable object would compromise security or other important properties, you will need to do something different. Storing a copy of the object is better approach in many situations.

외부에서 변경 가능한 객체에 대한 참조 -> 인스턴스 신뢰할 수 없을 경우 문제가 생기므로 '사본' 을 만들어 저장하는 것을 추천하고 있음.

SonarQube

T4 Result_Code Smell

The screenshot shows the SonarQube interface with a code smell detected in a Java test method. The code is as follows:

```
28 @Test // @@@@@ @@@ @G@@@@ @@@@T
29 public void testNewSpecification1() {
30     file = new File("newTest.yzb");
31     if(file.exists()){
32         file.delete();
33     }
34 }
```

The code smell message is: "Exceptional return value of java.io.File.delete() ignored in test.FileManagerTest.testNewSpecification1()". It is categorized as a "Code Smell" with a "Major" severity, "Open" status, and "1h effort". The message also indicates it was found "3일 전" (3 days ago) at "L32" and is associated with the "bad-practice" rule.

The description of the code smell states: "This method returns a value that is not checked. The return value should be checked since it can indicate an unusual or unexpected function execution. For example, the `File.delete()` method returns false if the file could not be successfully deleted (rather than throwing an Exception). If you don't check the result, you won't notice if the method invocation signals unexpected behavior by returning an atypical return value."

**File.delete (): 파일을 성공적으로 삭제할 수 없으면 예외를 throw하지 않고 false를 반환
-> return값 확인 추천.**

SonarQube

T4 Result_Code Smell

Vulnerability	Count
Vulnerability	0
Code Smell	2

해결 상태			
Unresolved	2	Fixed	0
False Positive	0	Won't fix	0
Removed	2		

```
18 public Category(String categoryName, Hashtable<String, Property> pt){
19     this.categoryName = new String(categoryName);
20     this.representativeValues = new ArrayList<RepresentativeValue>();
21     this.pt = pt;
```

new System.Category(String, Hashtable) may expose internal representation by storing an externally mutable object into Category.pt

Code Smell Info Open 할당되지 않음 30min effort 코멘트 malicio

Malicious code - May expose internal representation by incorporating reference to mutable object

Code Smell Info malicious-code 다음 기간 이후 적용됨 2017년 5월 30일 상수/이슈: 30min findbugs:EI_EXPOSE_REP2

This code stores a reference to an externally mutable object into the internal representation of the object. If instances are accessed by untrusted code, and unchecked changes to the mutable object would compromise security or other important properties, you will need to do something different. Storing a copy of the object is better approach in many situations.

pt -> private Variable, 하지만 get set 사용하지 않고 Category.pt 로 접근.



SonarQube

T4 Result_Code Smell



규칙	개수
Indentation	1,578
Line Length	335
l18n - Reliance on default encoding	8
Multiple Variable Declarations	4
Bad practice - Method may fail to close stream	3
Style - Write to static field from instance method	3
Variable Declaration Usage Distance	2

29
30
31

32
33
34
35

```
@Before  
public void Before()  
{  
}
```

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation. [...](#) 한시간 전 L31

Code Smell Critical Open 할당되지 않음 5min effort [코멘트](#) suspicious

빈 Method





SonarQube

T4 Result_Bug1



```
filePath = new String(filePath+".yzb");
```

GUI.Explorer.doit(int, FileManager, Specification) invokes inefficient new String(String) constructor

Bug  Major  Open 할당되지 않음 30min effort

java.lang.String (String) 생성자를 사용하면
생성된 객체가 매개 변수로 전달된 String과 기능적으로 구분되지 않으므로 **메모리가 낭비됩니다.**

->String 인자를 직접 사용하십시오.(뒷장에 같은 오류)



SonarQube

T4 Result_Bug2,3

```
T4_source
src/System/Category.java
1 package System;
2
3 import java.util.ArrayList;
4 import java.util.Hashtable;
5
6 import javax.swing.JOptionPane;
7
8 public class Category {
9     private String categoryName;
10    private ArrayList<RepresentativeValue> representativeValues;
11    private Hashtable<String, Property> pt;
12
13    //Constructor
14    public Category(){
15        this.categoryName = new String();
16
17        this.representativeValues = new ArrayList<RepresentativeValue>();
18    }
19    public Category(String categoryName, Hashtable<String, Property> pt){
20        this.categoryName = new String(categoryName);
21    }
22 }
```

new System.Category() invokes inefficient new String() constructor ...
Bug Major Open 할당되지 않음 30min effort

new System.Category(String, Hashtable) invokes inefficient new String(String) constructor ...
Bug Major Open 할당되지 않음 30min effort



SonarQube

T4 Result_Bug4,5





T4_source

src/System/Property.java



```
1 package System;
2
3 import java.util.LinkedList;
4
5 public class Property {
6     private String name;
7     private LinkedList<String> valueNames;
8
9     public Property(){
10         this.name = new String();
```

new System.Property() invokes inefficient new String() constructor ...

Bug  Major  Open 할당되지 않음 30min effort

```
11     }
12
13     public Property(String name){
14         this.name = new String(name);
```

new System.Property(String) invokes inefficient new String(String) constructor ...

Bug  Major  Open 할당되지 않음 30min effort



SonarQube

T4 Result_Bug6,7

src/System/RepresentativeValue.java

```
1 package System;
2
3 import java.util.ArrayList;
4 import java.util.Hashtable;
5
6 import javax.swing.JOptionPane;
7
8 public class RepresentativeValue {
9     private String representativeValueName;
10    private ArrayList<Property> propertyConstraint;
11    private ArrayList<Property> if_propertyConstraint;
12    private int singleErrorConstraint;
13    private int priority;
14    private Hashtable<String, Property> propertyTable;
15
16    //Constructor
17    public RepresentativeValue() {
18        this.representativeValueName = new String();
19
20        this.propertyConstraint = new ArrayList<Property>();
21        this.if_propertyConstraint = new ArrayList<Property>();
22        this.priority=1;
23    }
24    public RepresentativeValue(String representativeValueName, Hashtable<String, Property> pt) {
25        this.representativeValueName = new String();
```

new System.RepresentativeValue() invokes inefficient new String() constructor ...

Bug Major Open 할당되지 않음 30min effort

new System.RepresentativeValue(String, Hashtable) invokes inefficient new String() constructor ...

Bug Major Open 할당되지 않음 30min effort



Find Bugs

Sonar Qube_T4 Result_Code Smell

T4_source src/GUI/FirstFrame.java

GUI.FirstFrame\$6.actionPerformed(ActionEvent) invokes System.exit(...), which shuts down the entire virtual machine ...

Code Smell Major Open 할당되지 않음 1h effort

T4_source src/GUI/MainFrame.java

GUI.MainFrame\$18.actionPerformed(ActionEvent) invokes System.exit(...), which shuts down the entire virtual machine ...

Code Smell Major Open 할당되지 않음 1h effort

T4_source src/System/Category.java

new System.Category(String, Hashtable) may expose internal representation by storing an externally mutable object into Category.pt ...

Code Smell Info Open 할당되지 않음 30min effort

외부에서 변경 가능한 객체를 저장-> expose internal representation



Find Bugs

Eclipse Plugin_T4 Result



FileManagerTest.java: 46

Navigation

Write to static field test.FileManagerTest.file from instance method test.FileManagerTest.testLoadSpecification1()
Field test.FileManagerTest.file

Bug: Write to static field test.FileManagerTest.file from instance method test.FileManagerTest.testLoadSpecification1()

This instance method writes to a static field. This is tricky to get correct if multiple instances are being manipulated, and generally bad practice.

Rank: Of Concern (15), **confidence:** High

Pattern: ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD

Type: ST, **Category:** STYLE (Dodgy code)

XML output:

```
<BugInstance type="ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD" priority="1" rank="15" abbrev="ST" category="STYLE" first="1">
  <Class classname="test.FileManagerTest">
    <SourceLine classname="test.FileManagerTest" sourcefile="FileManagerTest.java" sourcepath="test/FileManagerTest.java"/>
  </Class>
  <Method classname="test.FileManagerTest" name="testLoadSpecification1" signature="()V" isStatic="false">
    <SourceLine classname="test.FileManagerTest" start="46" end="54" startBytecode="0" endBytecode="137" sourcefile="FileManagerTest.java" sourcepath="test/FileManagerTest.java"/>
  </Method>
  <Field classname="test.FileManagerTest" name="file" signature="Ljava/io/File;" isStatic="true">
    <SourceLine classname="test.FileManagerTest" sourcefile="FileManagerTest.java" sourcepath="test/FileManagerTest.java"/>
  </Field>
  <SourceLine classname="test.FileManagerTest" start="46" end="46" startBytecode="9" endBytecode="9" sourcefile="FileManagerTest.java" sourcepath="test/FileManagerTest.java"/>
  <SourceLine classname="test.FileManagerTest" start="46" end="46" startBytecode="9" endBytecode="9" sourcefile="FileManagerTest.java" sourcepath="test/FileManagerTest.java"/>
</BugInstance>
```

LoadSpecification1, 2, new Specification1() 모두 해당.



Metrics

Eclipse Plugin_T4 Result

Cyclomatic Complexity	또 다른오류가생길확률
1~10	5%
20~30	20%
50 이상	40%
거의 100	60%



복잡도 $v(G) = \text{분기문} + 1$
System package : 20
GUI package : 14
test package : 4

Metric	Total	Mean	Std. Dev.	Maximum
▼ McCabe Cyclomatic Complexity (avg/max per method)		2.516	3.185	20
▼ src		2.516	3.185	20
> System		3.035	3.494	20
> GUI		3.2	4.167	14
> test		1.115	0.319	2



Metrics

Eclipse Plugin_T4 Result



Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		2.516	3.185	20	/T4_source/src/System/FileManager.java	loadSpecification
> Number of Parameters (avg/max per method)		0.731	0.985	5	/T4_source/src/System/Specification.java	f
> Nested Block Depth (avg/max per method)		1.946	1.498	8	/T4_source/src/System/FileManager.java	loadSpecification
> Afferent Coupling (avg/max per packageFragment)		3	3.559	8	/T4_source/src/System	
> Efferent Coupling (avg/max per packageFragment)		4.333	0.943	5	/T4_source/src/GUI	
> Instability (avg/max per packageFragment)		0.702	0.311	1	/T4_source/src/test	
> Abstractness (avg/max per packageFragment)		0	0	0	/T4_source/src/GUI	
> Normalized Distance (avg/max per packageFragment)		0.298	0.311	0.727	/T4_source/src/System	
> Depth of Inheritance Tree (avg/max per type)		2.071	2.052	6	/T4_source/src/GUI/NewSpecification.java	
> Weighted methods per Class (avg/max per type)	234	16.714	18.223	58	/T4_source/src/System/FileManager.java	
> Number of Children (avg/max per type)	0	0	0	0	/T4_source/src/GUI/NewSpecification.java	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/T4_source/src/GUI/NewSpecification.java	
> Lack of Cohesion of Methods (avg/max per type)		0.381	0.342	0.846	/T4_source/src/GUI/FirstFrame.java	
> Number of Attributes (avg/max per type)	46	3.286	3.594	13	/T4_source/src/GUI/FirstFrame.java	
> Number of Static Attributes (avg/max per type)	14	1	2	6	/T4_source/src/test/RepresentativeValueTest.java	
> Number of Methods (avg/max per type)	89	6.357	4.922	18	/T4_source/src/System/RepresentativeValue.java	
> Number of Static Methods (avg/max per type)	4	0.286	0.452	1	/T4_source/src/System/MainSystem.java	
> Specialization Index (avg/max per type)		0	0	0	/T4_source/src/GUI/NewSpecification.java	
> Number of Classes (avg/max per packageFragment)	14	4.667	0.943	6	/T4_source/src/System	
> Number of Interfaces (avg/max per packageFragment)	0	0	0	0	/T4_source/src/GUI	
> Number of Packages	3					
> Total Lines of Code	1888					
> Method Lines of Code (avg/max per method)	1481	15.925	53.513	498	/T4_source/src/GUI/MainFrame.java	MainFrame



Metrics

Eclipse Plugin_T4 Result_Dependency Graph View

