

# 소프트웨어 모델링 및 분석

정적 분석 결과 보고서

201411278 서희진

201411304 이지수

201411317 조민규

201213189 박성규

2017년 6월 1일

## 내용

개요.....	2
정적 분석.....	2
JDepend.....	2
목적 .....	2
방법 .....	2
결과: T1-마감 하루 전 Category Partitioning Testing Tool.....	2
결과: T4-Priority Category Partitioning Testing Tool.....	3
FindBugs.....	4
목적 .....	4
방법 .....	4
결과: T1-마감 하루 전 Category Partitioning Testing Tool.....	5
결과: T4-Priority Category Partitioning Testing Tool.....	6
CheckStyle.....	7
목적 .....	7
방법 .....	7
결과: T1-마감 하루 전 Category Partitioning Testing Tool.....	7
결과: T4-Priority Category Partitioning Testing Tool.....	9
Specification and code review .....	10
결과: T1-마감 하루 전 Category Partitioning Testing Tool.....	10
결과: T4-Priority Category Partitioning Testing Tool.....	16

## 개요

이 문서는 건국대학교 '소프트웨어 모델링' 과목의 2 조 (조원 서희진, 이지수, 조민규, 박성규)의 발표를 위해 작성된 보고서이다.

2 조는 '소프트웨어 모델링' 과목의 1 조의 프로젝트 산출물 '마감 하루 전 Category Partitioning Testing Tool' , 4 조의 프로젝트 산출물 'Priority Category Partitioning Testing Tool' 의 (Ver.2017.05.25) 를 대상으로 하여 2017 년 5 월 26 일 – 2017 5 월 31 일의 기간 중 정적 분석을 진행 하였다.

이 문서는 대상 소프트웨어에 대한 정적 분석 결과를 포함하고 있다.

## 정적 분석

### 1. JDepend

#### 1-1. 목적

패키지 별로 의존성을 측정한다.

#### 1 -2 . 방법

CC : 인터페이스나 추상클래스가 아닌 Concrete Class 의 수

AC : 추상클래스나 인터페이스의 수 ( 확장성의 척도)

Ca : 현재 패키지의 클래스에 의존하는 패키지의 수 ( 책임의 척도)

A : 추상화 정도 (0 – 구체적, 1 – 추상적)

I : 변화에 대한 안정성 (0 – 안정적, 1 – 불안)

D : Main Sequence 로부터의 거리 (0 – 이상적, 1 – 이상적이지 않음)

추상화 클래스가 많은 패키지는 다른 패키지들이 이 패키지에 의존하여야 한다.

구체적인 클래스가 많은 패키지는 다른 패키지들이 이 패키지에 의존하지 말아야한다.

Cycle : 패키지들 상호 간에 의존성을 가지고 있을 때 발생. (경고아이콘)

#### 1-3. 결과

### T1 – 마감 하루 전 Category Partitioning Testing Tool

Selected object(s)								
Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
Default	16	0	0	1	0.00	1.00	0.00	
org.apache.poi.xssf.usermodel	0	0	1	0	0.00	0.00	1.00	

Packages with cycle								
Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!

Depends upon - efferent dependencies								
Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
org.apache.poi.xssf.usermodel	0	0	1	0	0.00	0.00	1.00	

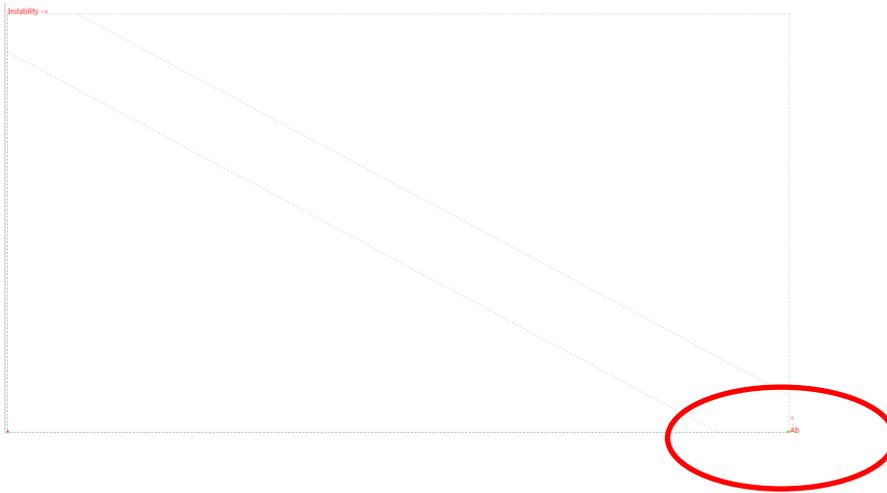
  

Used by - afferent dependencies								
Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
Default	16	0	0	1	0.00	1.00	0.00	

패키지 하나에 모든 클래스가 다 들어 있기 때문에 의존성을 판단하기 쉽지 않다.

항목 I 에 의하면 변화에 대해 매우 안정적이지 않은 것으로 나왔다.

항목 AC 에 의하면 인터페이스나 추상클래스가 존재하지 않기 때문에 확장이 어려워 보인다.



## T4 – Priority Category Partitioning Testing Tool

Dependencies								
Selected object(s)								
Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
GUI	36	0	1	1	0.00	0.50	0.50	⚠
System	7	0	2	4	0.00	0.66	0.33	⚠
com.jtattoo.plaf.acryl	0	0	1	0	0.00	0.00	1.00	
org.apache.poi.hssf.usermodel	0	0	1	0	0.00	0.00	1.00	
org.apache.poi.ss.usermodel	0	0	1	0	0.00	0.00	1.00	
org.junit	0	0	1	0	0.00	0.00	1.00	
test	4	0	0	2	0.00	1.00	0.00	⚠

Packages with cycle

Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
GUI	36	0	1	1	0.00	0.50	0.50	⚠
System	7	0	2	4	0.00	0.66	0.33	⚠

Depends upon - efferent dependencies

Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
GUI	36	0	1	1	0.00	0.50	0.50	⚠
System	7	0	2	4	0.00	0.66	0.33	⚠
com.jtattoo.plaf.acryl	0	0	1	0	0.00	0.00	1.00	
org.apache.poi.hssf.usermodel	0	0	1	0	0.00	0.00	1.00	
org.apache.poi.ss.usermodel	0	0	1	0	0.00	0.00	1.00	
org.junit	0	0	1	0	0.00	0.00	1.00	

Used by - afferent dependencies

Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
GUI	36	0	1	1	0.00	0.50	0.50	⚠
System	7	0	2	4	0.00	0.66	0.33	⚠
test	4	0	0	2	0.00	1.00	0.00	⚠

항목 I 에 의하면 System 패키지는 안정성 0.66 으로 변화에 안정적이지 않다.

항목 A 와 Ce 에 의하면 GUI 패키지는 구체적이지만 이 패키지의 클래스들은 1 개의 패키지에만 종속하고 있기 때문에 이상적이지 않다.

항목 AC 에 의하면 인터페이스나 추상클래스가 존재하지 않기 때문에 확장이 어려워 보인다.

항목 Ce 에 의하면 System 패키지의 클래스들은 4 개의 패키지에 의존하고있다.

## 2. FindBugs

### 2-1. 목적

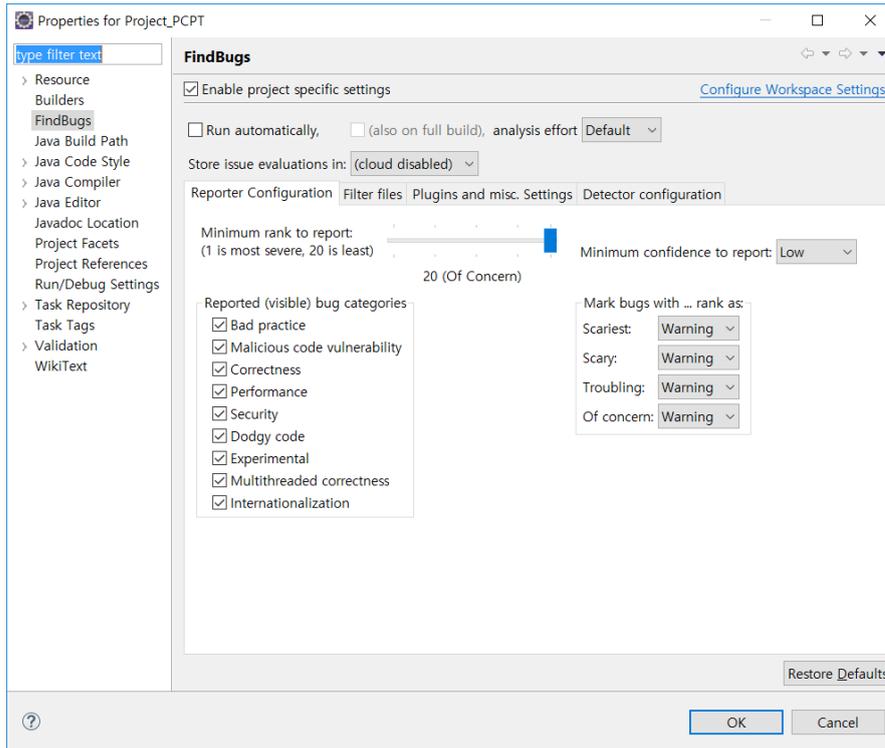
소스 코드에 문제가 될 수 있는 부분을 찾는 정적 분석 도구.

Java 프로그램에서 100 여개의 잠재적인 에러 타입을 찾아주며, 이 잠재적인 에러는 scariest, scary, troubling, concern 으로 구분하여 점수(rank)가 주어진다.

### 2-2. 방법

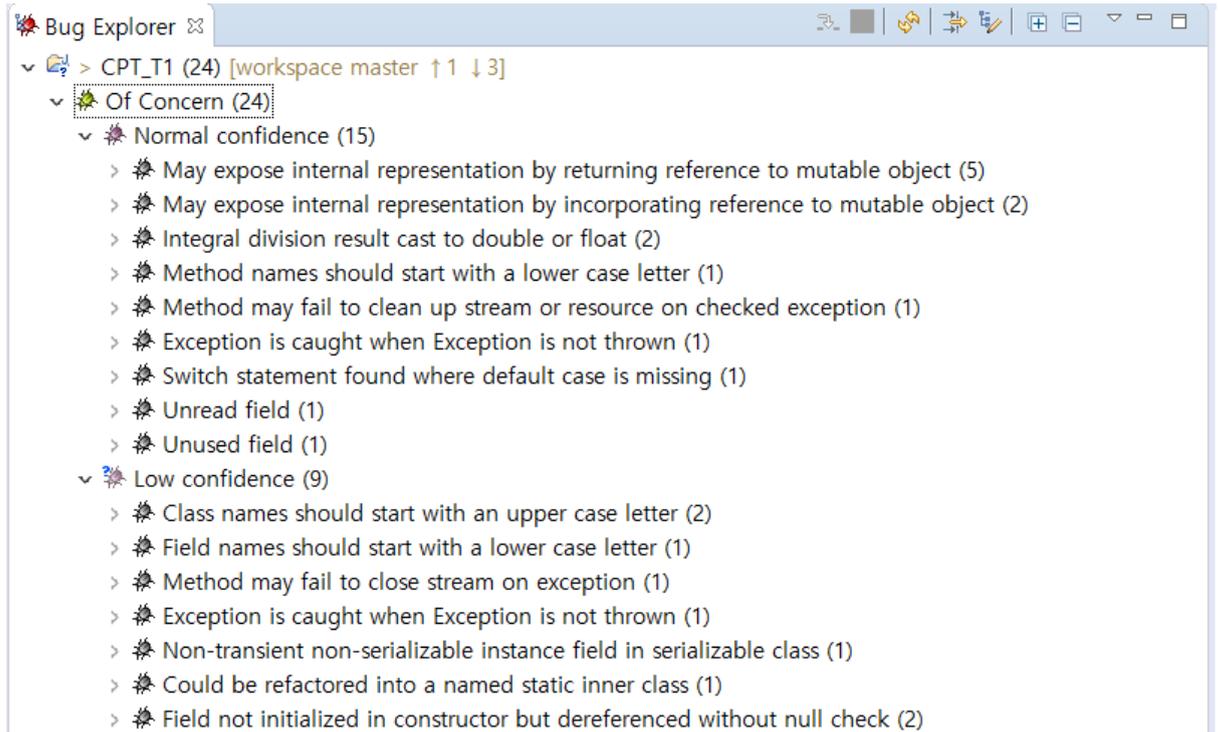
Eclipse - Install – Install new software -> <http://findbugs.cs.umd.edu/eclipse> 를 통해 설치.

project -> properties -> Findbugs -> enable 지정 후 다음과 같은 설정을 통해 테스트하였다.



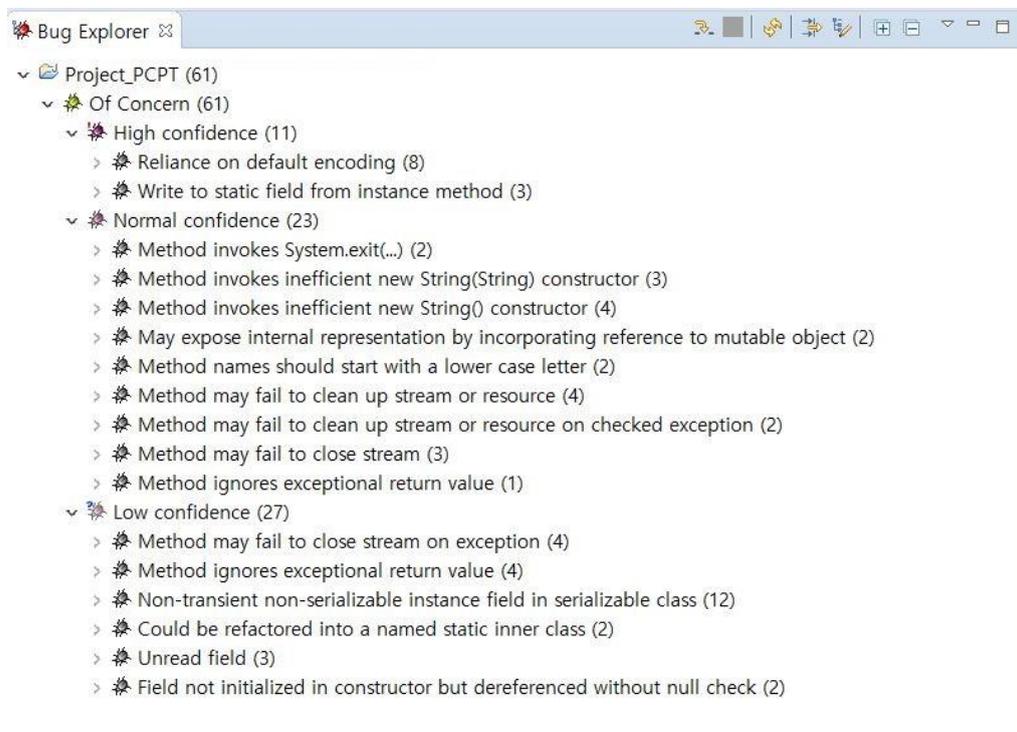
## 2-3. 결과

### T1 – 마감 하루 전 Category Partitioning Testing Tool



FindBugs 를 사용한 정적 분석 결과, 총 24 개의 Bug 를 발견하였으며, 이 중 Low 등급은 9 개, Normal 등급은 15 개, High 등급은 없었다.

## T4 – Priority Category Partitioning Testing Tool



FindBugs 를 사용한 정적 분석 결과, 총 61 개의 Bug 를 발견하였으며, 이 중 Low 등급은 27 개, Normal 등급은 23 개, High 등급은 11 개이다.

### 3. CheckStyle

#### 3-1. 목적

CheckStyle 은 프로그래머가 Java 의 소스코드(.java 파일)를 코딩규약에 준수하여 작성하고 있는지를 체크하는 정적해석 툴 이다. 협업을 위한 프로젝트를 진행할 때 팀만의 코딩규약을 설정할 수 있다.

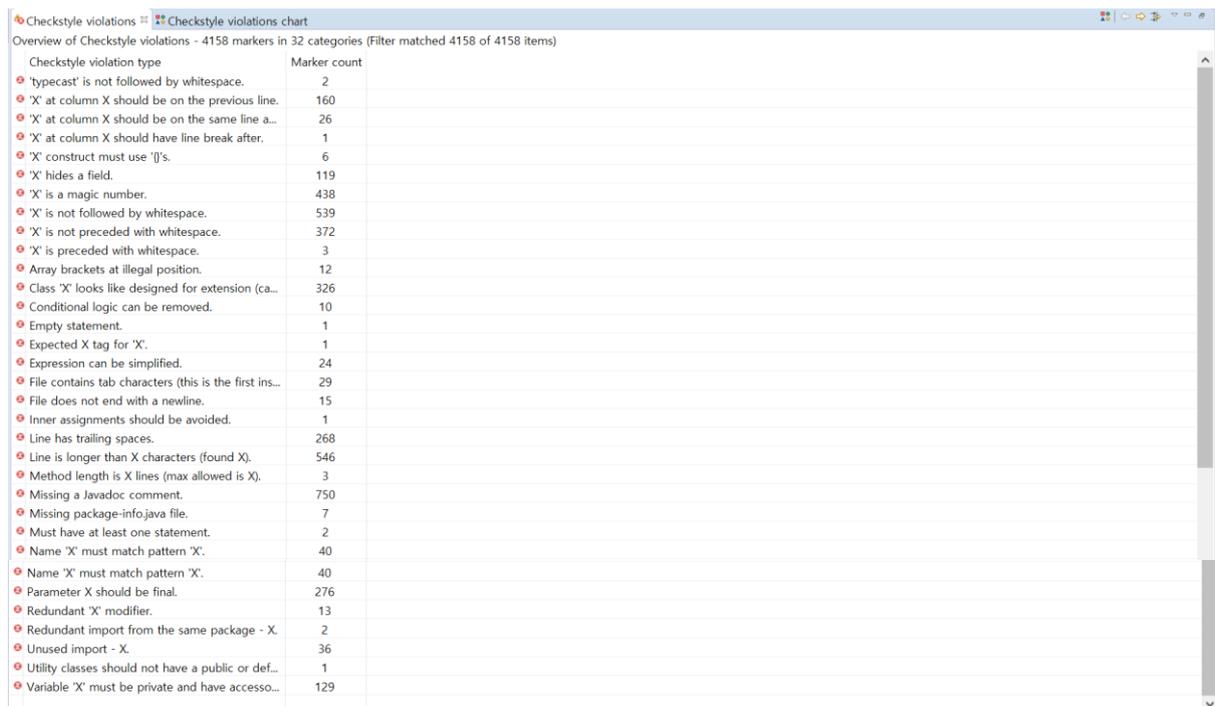
#### 3 -2 . 방법

Checkstyle violation type: 어떤 종류의 코딩 규약에 위배되었는지를 설명

Marker Count: 해당 Checkstyle violation type 을 위반한 개수

#### 3-3. 결과

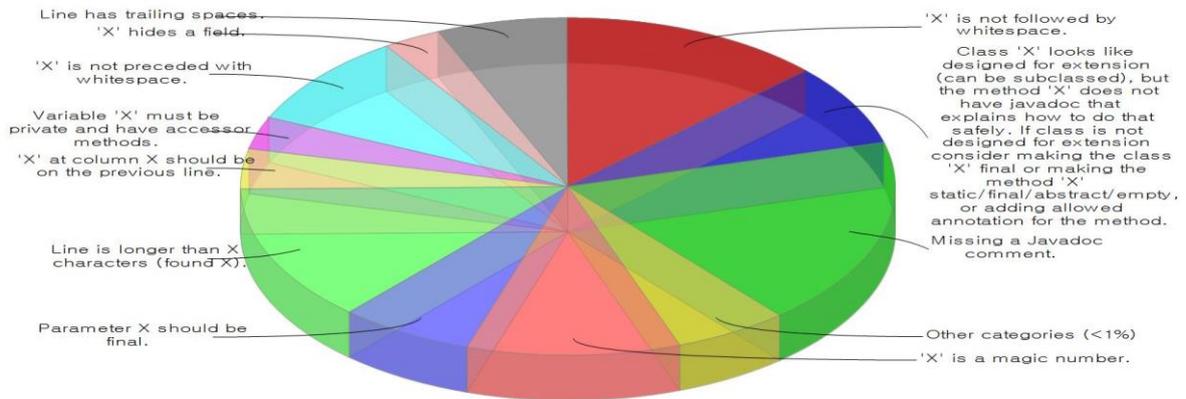
### T1 – 마감 하루 전 Category Partitioning Testing Tool



Overview of Checkstyle violations - 4158 markers in 32 categories (Filter matched 4158 of 4158 items)

Checkstyle violation type	Marker count
'typecast' is not followed by whitespace.	2
'X' at column X should be on the previous line.	160
'X' at column X should be on the same line a...	26
'X' at column X should have line break after.	1
'X' construct must use 'j's.	6
'X' hides a field.	119
'X' is a magic number.	438
'X' is not followed by whitespace.	539
'X' is not preceded with whitespace.	372
'X' is preceded with whitespace.	3
Array brackets at illegal position.	12
Class 'X' looks like designed for extension (ca...	326
Conditional logic can be removed.	10
Empty statement.	1
Expected X tag for 'X'.	1
Expression can be simplified.	24
File contains tab characters (this is the first ins...	29
File does not end with a newline.	15
Inner assignments should be avoided.	1
Line has trailing spaces.	268
Line is longer than X characters (found X).	546
Method length is X lines (max allowed is X).	3
Missing a Javadoc comment.	750
Missing package-info.java file.	7
Must have at least one statement.	2
Name 'X' must match pattern 'X'.	40
Name 'X' must match pattern 'X'.	40
Parameter X should be final.	276
Redundant 'X' modifier.	13
Redundant import from the same package - X.	2
Unused import - X.	36
Utility classes should not have a public or def...	1
Variable 'X' must be private and have accesso...	129

T1 의 프로젝트는 총 32 개 종류의 CheckStyle 을 위배하였으며, 위배된 코드의 총 개수는 4158 개이다



위의 원형 그래프는 T1 이 위배한 Style 을 비율로 보여주고 있다. T1 이 가장 많이 위배한 Style 을 정리하면 아래와 같다.

1. Missing a javadoc comment -> 18%
2. Line is longer than X characters (found X) -> 13.1%
3. 'X' is not followed by whitespace ->13.0%

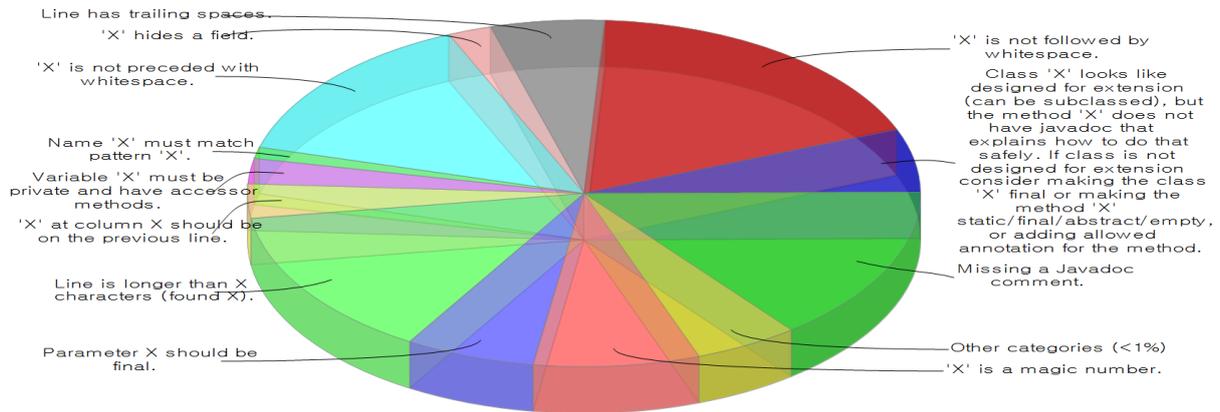
## T4 – Priority Category Partitioning Testing Tool

Checkstyle violations 35 Checkstyle violations chart

Overview of Checkstyle violations - 6248 markers in 35 categories (Filter matched 6248 of 6248 items)

Checkstyle violation type	Marker count
⊗ 'X' at column X should be on the previous line.	199
⊗ 'X' at column X should be on the same line a...	40
⊗ 'X' at column X should have line break after.	15
⊗ 'X' construct must use 'l's.	6
⊗ 'X' hides a field.	130
⊗ 'X' is a magic number.	498
⊗ 'X' is not followed by whitespace.	1123
⊗ 'X' is not preceded with whitespace.	884
⊗ 'X' is preceded with whitespace.	5
⊗ 'X' should be on a new line.	2
⊗ Array brackets at illegal position.	13
⊗ Class 'X' looks like designed for extension (ca...	369
⊗ Conditional logic can be removed.	10
⊗ Empty statement.	1
⊗ Expected X tag for 'X'.	4
⊗ Expression can be simplified.	24
⊗ File contains tab characters (this is the first ins...	43
⊗ File does not end with a newline.	16
⊗ First sentence should end with a period.	1
⊗ Inner assignments should be avoided.	1
⊗ Line has trailing spaces.	343
⊗ Line is longer than X characters (found X).	877
⊗ Method length is X lines (max allowed is X).	4
⊗ Missing a Javadoc comment.	909
⊗ Missing package-info.java file.	10
⊗ Must have at least one statement.	8
⊗ Name 'X' must match pattern 'X'.	66
⊗ Parameter X should be final.	390
⊗ Redundant 'X' modifier.	13
⊗ Redundant import from the same package - X.	2
⊗ Unused import - X.	46
⊗ Using the '*' form of import should be avoid...	25
⊗ Utility classes should not have a public or def...	2
⊗ Variable 'X' must be private and have accesso...	147

T4 의 프로젝트는 총 35 개 종류의 CheckStyle 을 위배하였으며, 위배된 코드의 총 개수는 6248 개이다



위의 원형 그래프는 T4 이 위배한 Style 을 비율로 보여주고 있다. T1 이 가장 많이 위배한 Style 을 정리하면 아래와 같다.

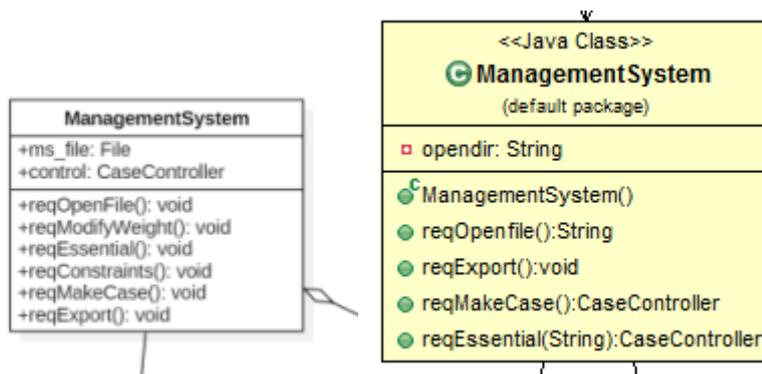
1. 'X' is not followed by whitespace -> 18.0%
2. Missing a javadoc comment -> 18%
3. Line is longer than X characters (found X) -> 13.1%

#### 4. Specification and code review

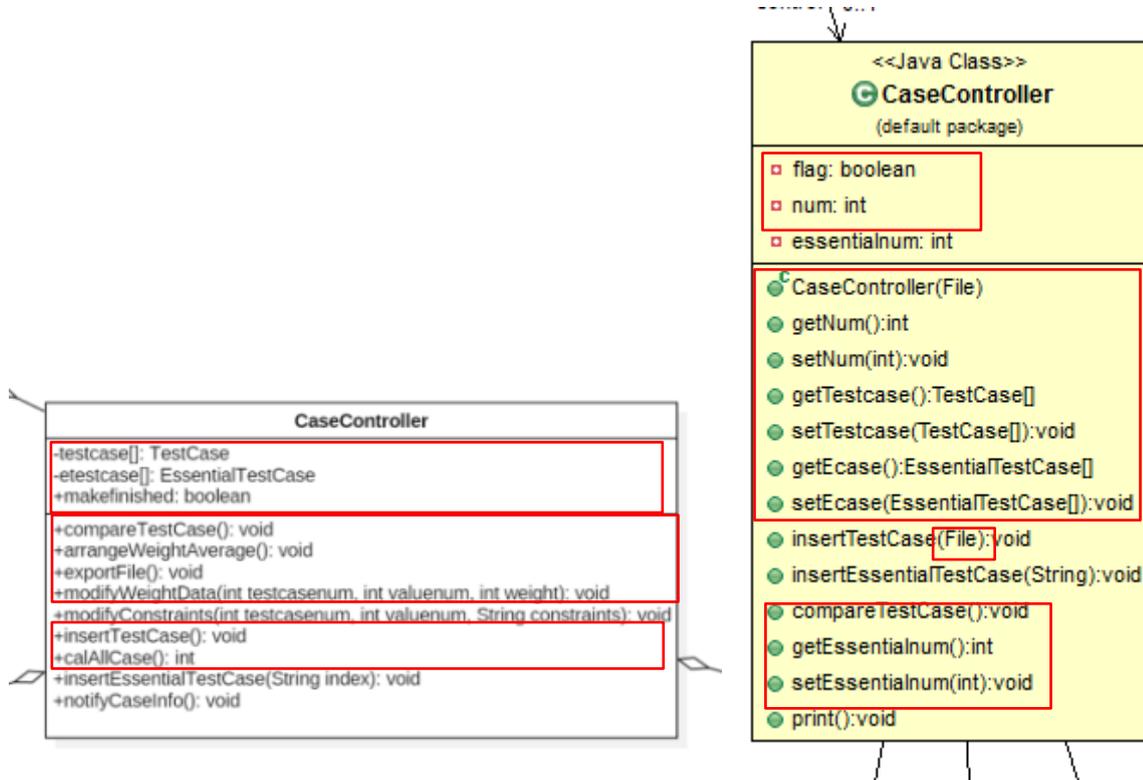
### T1 – 마감 하루 전 Category Partitioning Testing Tool

좌측 : stage2040\_T1.pdf – 2045. ClassDiagram

우측 : Eclipse Plugin – ObjectAid ('마감 하루 전 Category Partitioning Testing Tool' Ver.2017.05.25)



문서에 정의된 함수(예 - reqConstraints())를 사용하지 않았다.  
 문서에는 생성자가 정의 되어있지 않다.

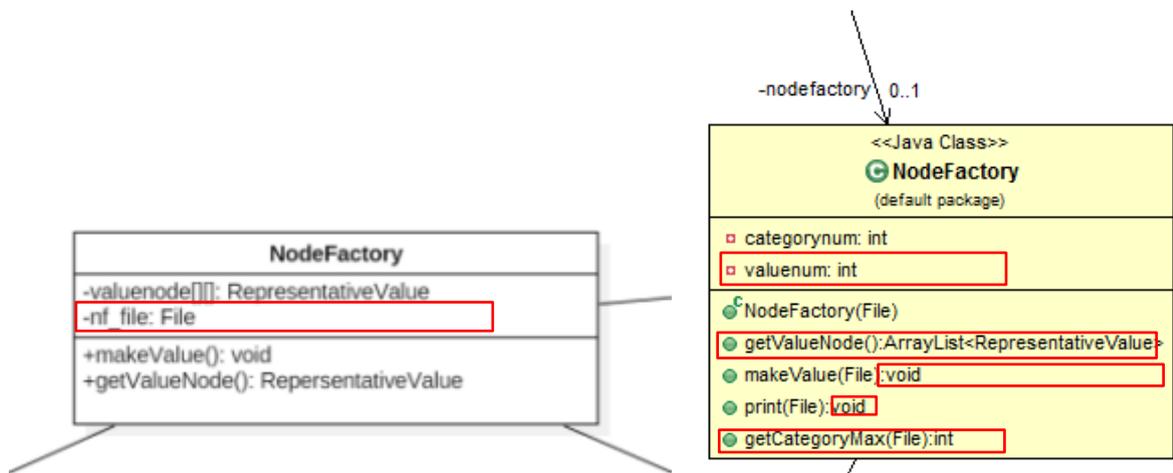


문서에서 정의한 변수들을 사용하지 않았다.

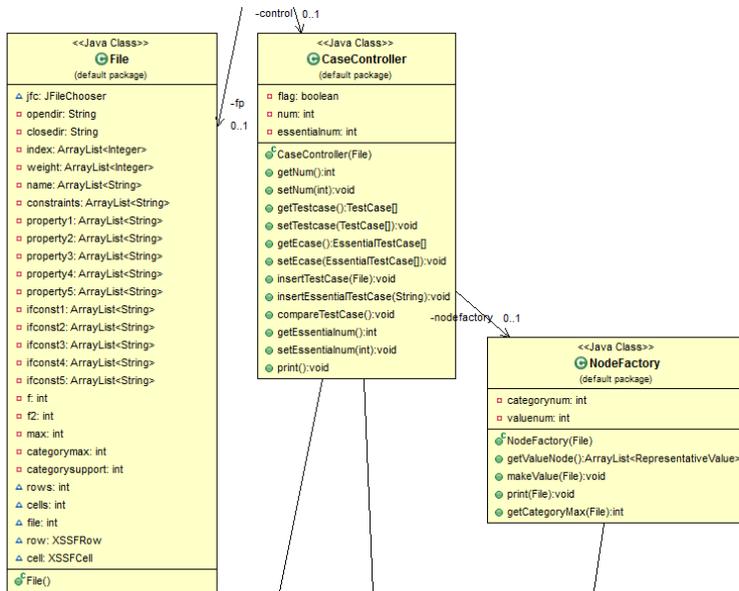
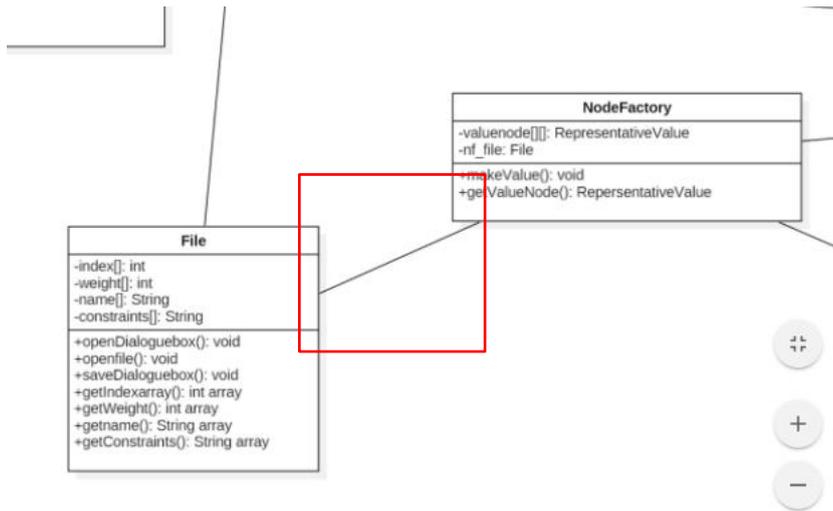
문서에서 정의한 함수들과 input 의 형식이 맞지않는다.

문서에서 정의한 함수들이 존재하지 않는다.

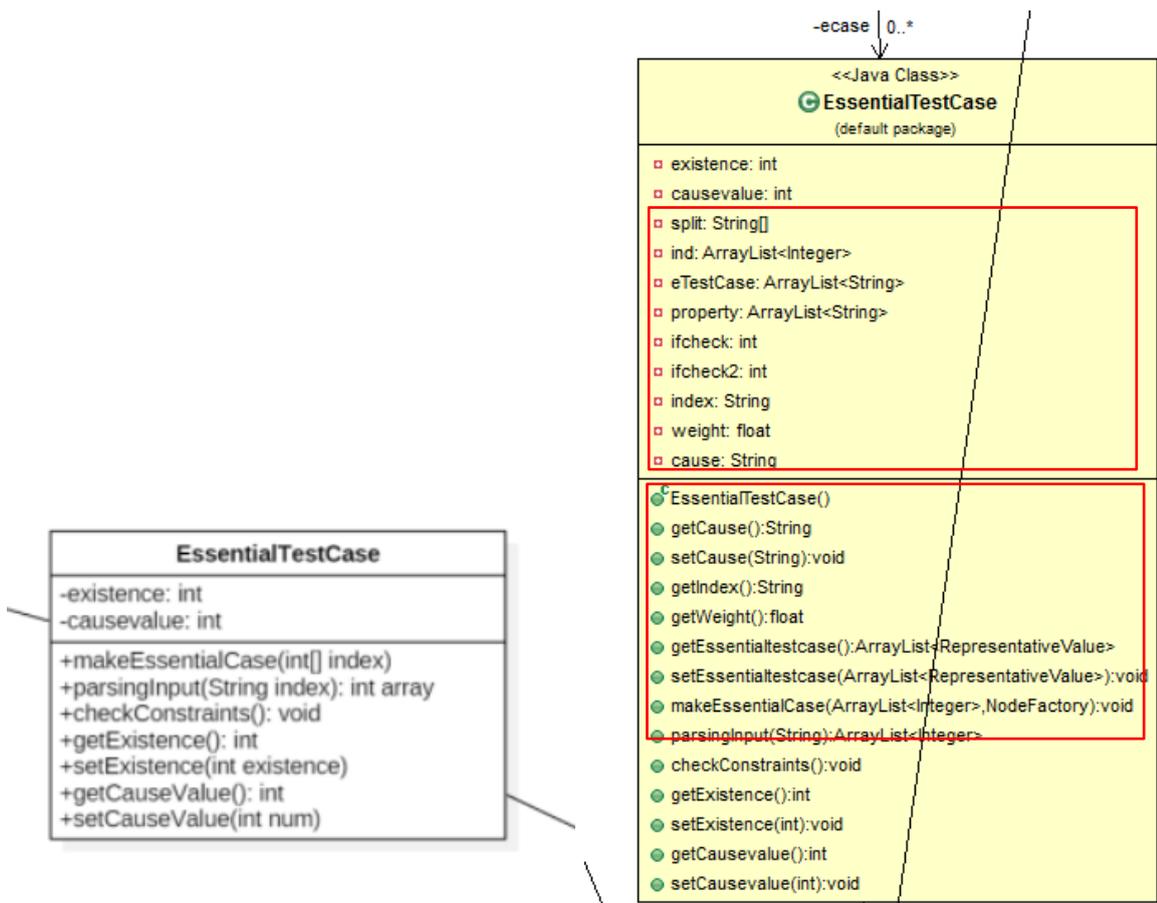
문서에서 정의하지 않은 함수들을 사용했다.



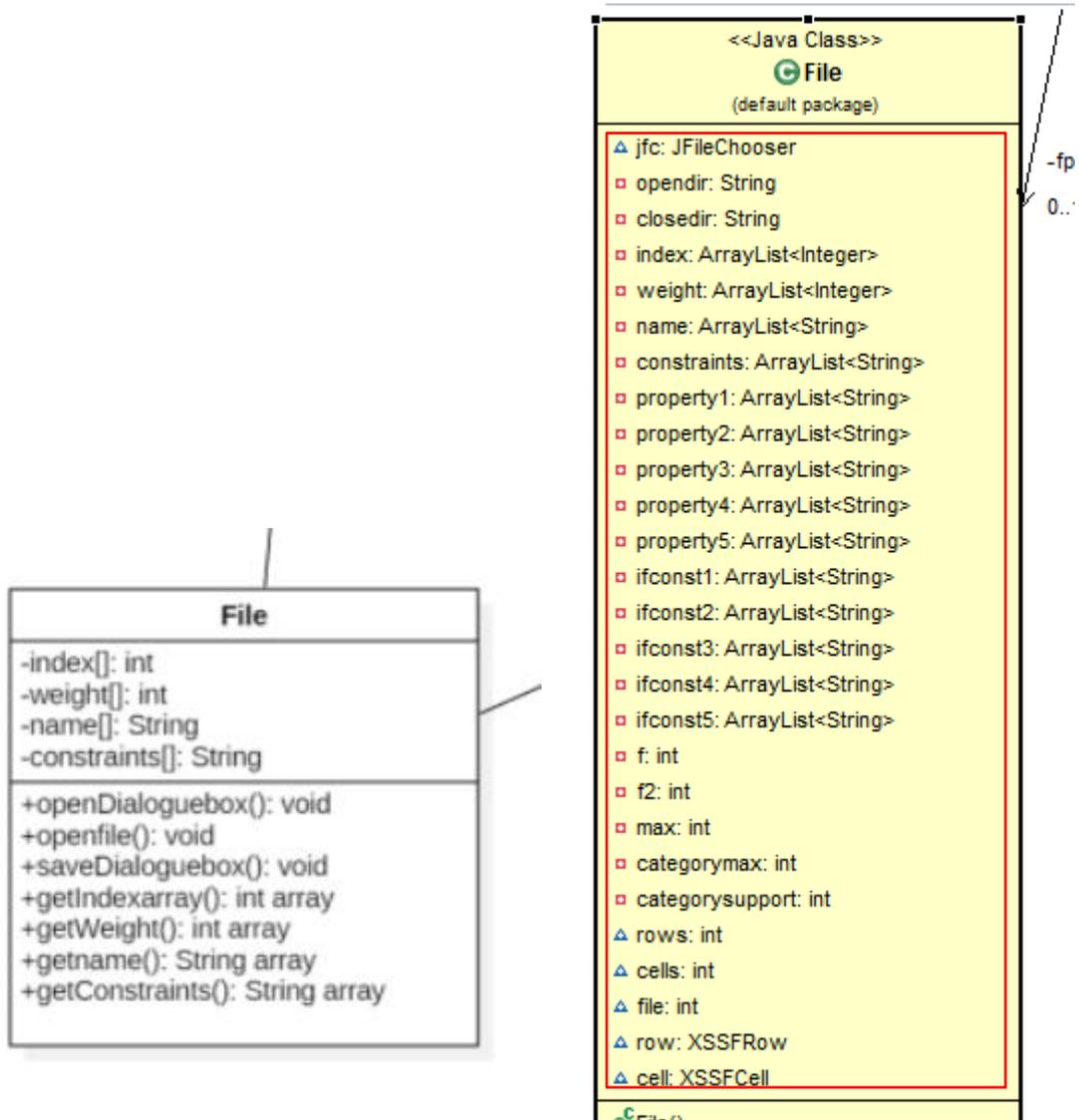
문서에서 정의하지 않은 함수들을 사용했다.



문서와 다르게 구현상에는 File 과 NodeFactory 의 관계가 존재하지않는다.



전체적으로 문서에서 정의되지 않은 새로운 변수와 함수들이 추가되었다.



전체적으로 문서에서 정의되지 않은 새로운 변수와 함수들이 추가되었다.  
 문서에서 정의된 변수들과 형식이 다르다.

문서에 누락된 내용이 상당수 존재하고, 함수명, 변수명 등의 차이가 굉장히 심해 더 이상 class diagram 을 비교하는 것은 무의미 할 것으로 판단된다. sequence diagram 을 대조하는 것 또한 무의미 할 것으로 판단된다.

## T4 – Priority Category Partitioning Testing Tool

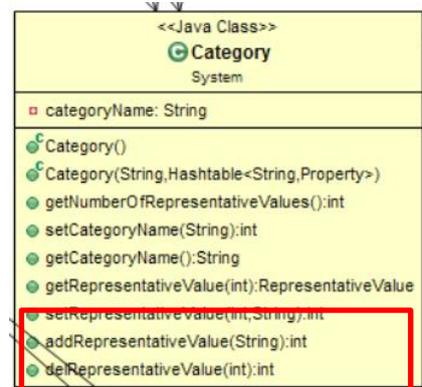
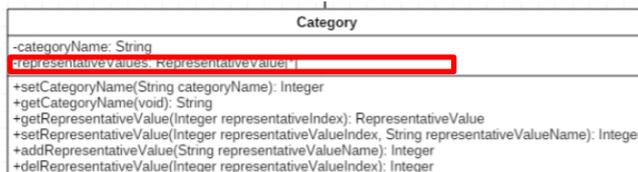
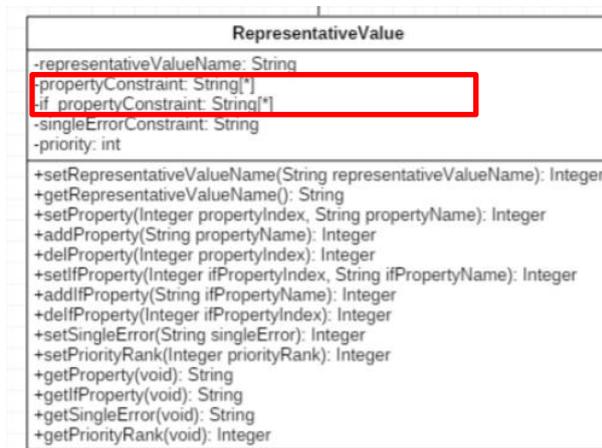
### 1. RepresentativeValue, Category, FileManager

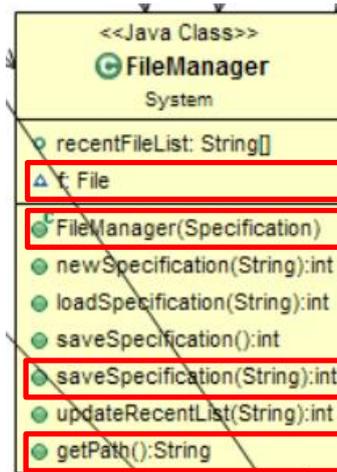
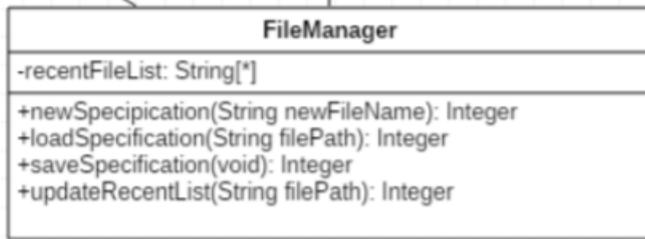
- 문서에 정의된 변수가 존재하지 않는다.

(예 - if\_propertyConstraint, propertyConstraint).

- 문서에 정의되지 않은 메소드를 정의하고 사용하였다.

(예- getNumberOfPropertyConstraints())





## 2. Specification

- 문서에 정의된 변수가 존재하지 않으며, 존재하는 변수 또한 자료형이 다르다.

(예- testcase : String[\*] -> HashMap<ArrayList<String>, Integer>)

