Static Analysis Report for Team 5

Project Team Team 5

Latest update on: 2017-05-31

Team Information
컴퓨터공학부 201111383 전 훈
컴퓨터공학부 201311259 권오승
컴퓨터공학부 201311292 유효상
컴퓨터공학부 201511266 배율희

Table of Contents

1. PMD

1. 코드 상의 불필요한 부분을 줄이는 것에 초점을 맞추어 분석하였다.
2. 빈 코드, 불필요하거나 사용되지 않은 코드, 단순화시킬 수 있는 코드를 찾을 수 있도록 설정하였다.
3. PMD는 검사 가능한 Rule을 종류별로 모아 RuleSet으로 묶어 나누었는데, 이 목록에서 필요한 Rule을 부분적으로 선택하여 설정하였다.
4. PMD 설정은 다음과 같다.

RuleSet	Rule
Basic	All
Design	SimplifyBooleanExpressions SimplifyBooleanReturns SimplifyConditional UnnecessaryLocalBeforeReturn
Empty Code	All
Optimization	SimplifyStartsWith
Unnecessary	All
Unused Code	All

5. Rule이 의미하는 바는 다음과 같다.

Rule	Description
UselessParenteses	필요 없는 괄호
UnnecessaryReturn	필요 없는 return
UnusedImports	사용되지 않은 라이브러리
UnusedLocalVariable	사용되지 않은 지역변수
EmptyIfStmt	비어 있는 조건문
SimplifyBooleanExpressions	축약 가능한 Boolean형
CollapsibleIfStatements	하나로 합칠 수 있는 If 중첩문

1.1 T7

Calculate.java

P	Line	created	Rule	Error Message
▶	109	Wed May 31 18:53:49 KST 2017	UselessParentheses	Useless parentheses.

Input_ManagerTest.java

P	Line	created	Rule	Error Message
▶	5	Wed May 31 18:53:49 KST 2017	UnusedImports	Avoid unused imports such as 'java.awt.List'
▶	6	Wed May 31 18:53:49 KST 2017	UnusedImports	Avoid unused imports such as 'java.util.ArrayList'

RV.java

P	Line	created	Rule	Error Message
▶	75	Wed May 31 18:53:49 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions

TC.java

P	Line	created	Rule	Error Message
▶	15	Wed May 31 18:53:49 KST 2017	UnusedLocalVariable	Avoid unused local variables such as 'tmp_rv'.

Test.java

P	Line	created	Rule	Error Message
▶	23	Wed May 31 18:53:49 KST 2017	UnusedImports	Avoid unused imports such as 'javax.swing.JRadioButton'
▶	10	Wed May 31 18:53:49 KST 2017	UnusedImports	Avoid unused imports such as 'java.util.ArrayList'
▶	20	Wed May 31 18:53:49 KST 2017	UnusedImports	Avoid unused imports such as 'Package.UI.ActionHandler'

UI.java

P	Line	created	Rule	Error Message
▶	4	Wed May 31 18:53:49 KST 2017	UnusedImports	Avoid unused imports such as 'java.awt.EventQueue'

1.2 T8

FeedbackController.java

P	Line	created	Rule	Error Message
▶	34	Wed May 31 19:30:10 KST 2017	CollapsibleIfStatements	These nested if statements could be combined
▶	42	Wed May 31 19:30:10 KST 2017	CollapsibleIfStatements	These nested if statements could be combined
▶	29	Wed May 31 19:30:10 KST 2017	CollapsibleIfStatements	These nested if statements could be combined
▶	55	Wed May 31 19:30:10 KST 2017	CollapsibleIfStatements	These nested if statements could be combined
▶	29	Wed May 31 19:30:10 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions
▶	34	Wed May 31 19:30:10 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions

Main.java

P	Line	created	Rule	Error Message
▶	530	Wed May 31 18:53:48 KST 2017	EmptyIfStmt	Avoid empty if statements
▶	528	Wed May 31 18:53:48 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions

MainController.java

P	Line	created	Rule	Error Message
▶	49	Wed May 31 18:53:48 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions
▶	45	Wed May 31 18:53:48 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions

Table.java

P	Line	created	Rule	Error Message
▶	46	Wed May 31 18:53:48 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions
▶	50	Wed May 31 18:53:48 KST 2017	SimplifyBooleanExpressions	Avoid unnecessary comparisons in boolean expressions
▶	1	Wed May 31 18:53:48 KST 2017	UnusedImports	Avoid unused imports such as 'javax.swing.JTable'

TestCaseController.java

P	Line	created	Rule	Error Message
▶	3	Wed May 31 18:53:48 KST 2017	UnusedImports	Avoid unused imports such as 'java.util.ArrayList'

TestCaseControllerTest.java

P	Line	created	Rule	Error Message
▶	29	Wed May 31 19:30:10 KST 2017	UnnecessaryReturn	Avoid unnecessary return statements
▶	128	Wed May 31 19:30:10 KST 2017	UnnecessaryReturn	Avoid unnecessary return statements
▶	42	Wed May 31 19:30:10 KST 2017	UnnecessaryReturn	Avoid unnecessary return statements
▶	142	Wed May 31 19:30:10 KST 2017	CollapsibleIfStatements	These nested if statements could be combined
▶	99	Wed May 31 19:30:10 KST 2017	CollapsibleIfStatements	These nested if statements could be combined
▶	138	Wed May 31 19:30:10 KST 2017	UselessParentheses	Useless parentheses.
▶	173	Wed May 31 19:30:10 KST 2017	UselessParentheses	Useless parentheses.
▶	138	Wed May 31 19:30:10 KST 2017	UselessParentheses	Useless parentheses.

2. JDepend

CC : 구체적인 클래스의 숫자

AC : 추상클래스 숫자

Ca : 연재 패키지의 클래스를 다른 패키지가 종속(사용)하는 숫자

Ce : 연재 패키지의 클래스가 다른 패키지를 종속(사용)하는 숫자

A : 인터페이스와 추상클래스의 개수 – 추상화된 패키지 비율이 높으면 A가 1에가까워짐

I : 1에 가까울수록 불안정한 패키지의 값 (CA+CE)/(CE)

D : 추상적인 정도와 불안정성 사이의 균형 1에 가까울수록 결합력이 강함

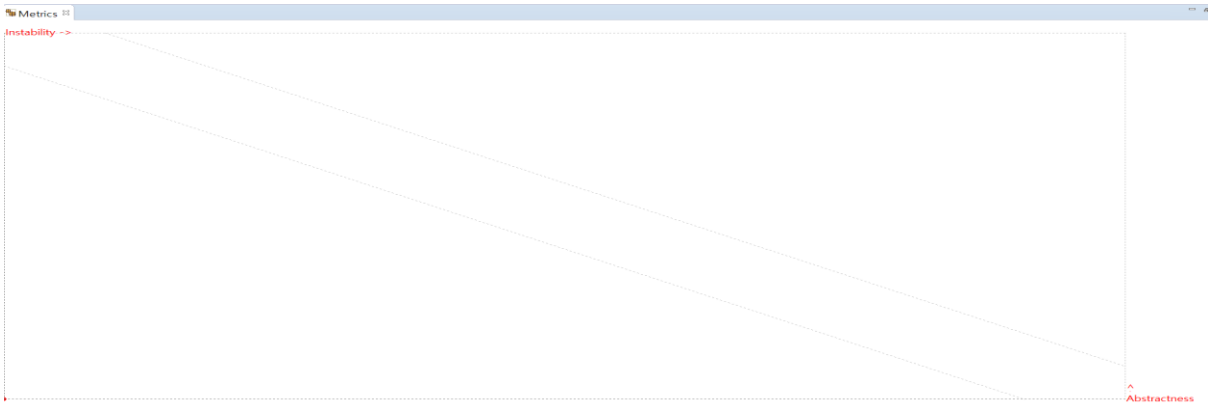
2.1 T7

총평 : 생성되는 클래스가 30개이다. D가 1이므로 구체적인 기능이 구체적으로 구현 되어있다

Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
Package	30	0	0	0	0.00	0.00	1.00	

```
Stats:
    Total Classes: 30
    Concrete Classes: 30
    Abstract Classes: 0

    Ca: 0
    Ce: 0

    A: 0
    I: 0
    D: 1
```

```
Depends Upon:
    Not dependent on any packages.

Used By:
    Not used by any packages.
```



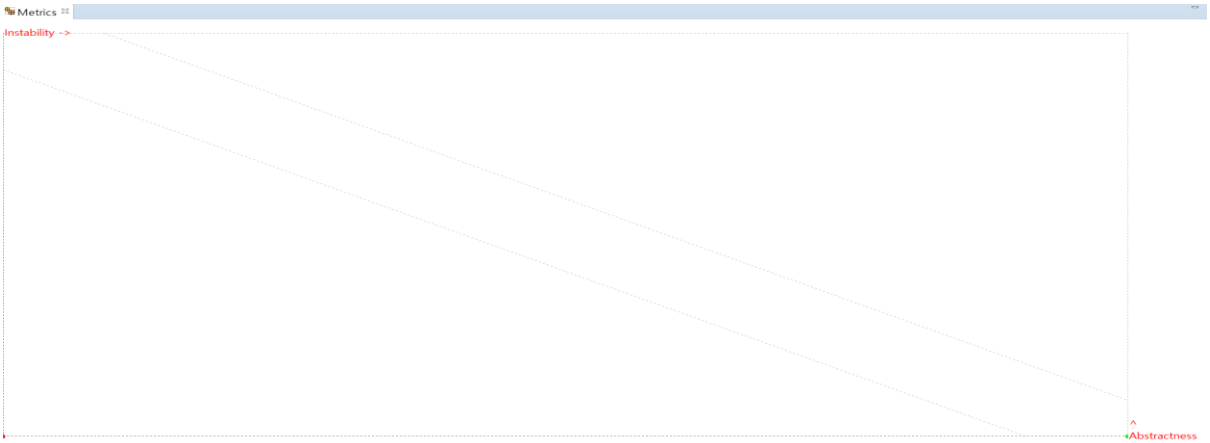
안정적이면서 추상화가 덜 되었음.

2.2 T8

총평 : org.junit은 8조팀이 테스트하면서 포함시켜서 있는거고 그것을 제외하고 보면 Ca랑 Ce는 0이다. (Ce가 1인거는 org.junit을 Default가 참고하고 있는거고 Ca가 1인거는 org.junit가 Default에 의해 사용되고 있음을 나타냄. 즉 Ca는 사용되어지는 것, Ce는 사용하는 것) 따라서 I가 1이므로 불안정한 패키지란 뜻인데, 아마도 org.junit 때문일 것

Package	CC(concr.cl.)	AC(abstr.cl.)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
Default	25	0	0	1	0.00	1.00	0.00	
org.junit	0	0	1	0	0.00	0.00	1.00	

```
JDepend
----------------------------
- Package: Default
----------------------------

Stats:
    Total Classes: 25
    Concrete Classes: 25
    Abstract Classes: 0

    Ca: 0
    Ce: 1

    A: 0
    I: 1
    D: 0

Depends Upon:
    org.junit

Used By:
    Not used by any packages.
```



안정적이면서 추상화가 덜 되었음.

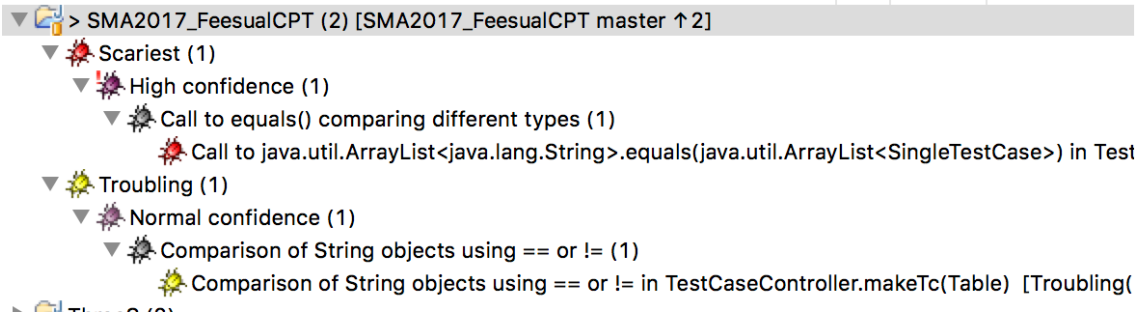
3. FindBug

7조 분석 결과

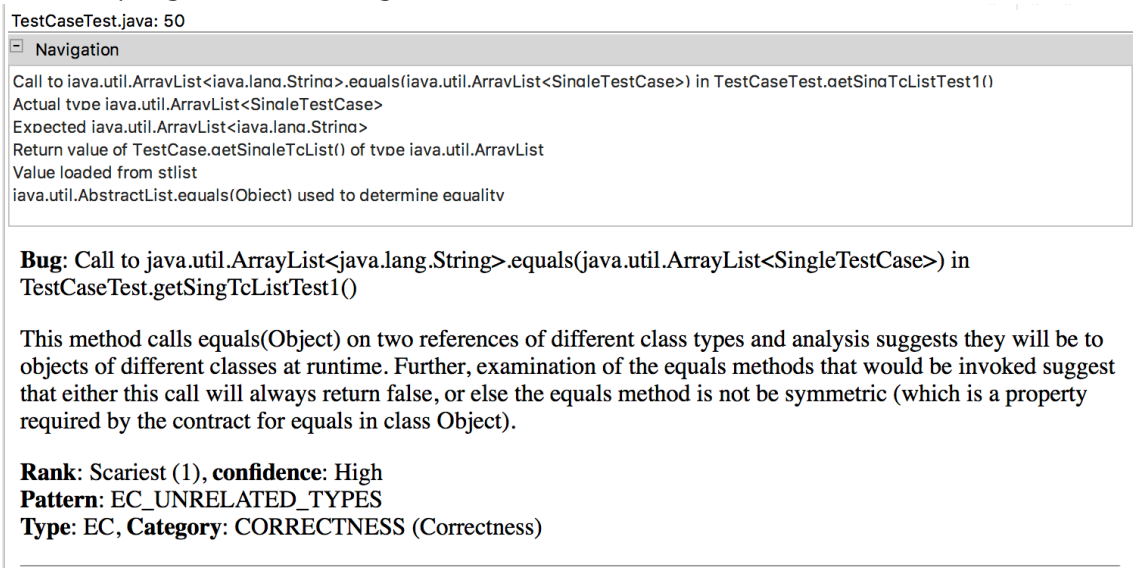
- FindBug 를 사용하여 분석한 결과 아무 warning 도 발생하지 않아서인지, 버그리포팅 파일이 생성되지 않았다.

8조 분석결과

- FindBug 를 사용하여 분석한 결과, 총 2 개의 warning 이 발견 되었다.



Scariest, High 등급 warning



Bug: Call to java.util.ArrayList<java.lang.String>.equals(java.util.ArrayList<SingleTestCase>) in TestCaseTest.getSingTcListTest1()

This method calls equals(Object) on two references of different class types and analysis suggests they will be to objects of different classes at runtime. Further, examination of the equals methods that would be invoked suggest that either this call will always return false, or else the equals method is not be symmetric (which is a property required by the contract for equals in class Object).

Rank: Scariest (1), **confidence**: High
Pattern: EC_UNRELATED_TYPES
Type: EC, **Category**: CORRECTNESS (Correctness)

Troubling, Normal등급 warning

TestCaseController.java: 162

Navigation

Comparison of String objects using == or != in TestCaseController.makeTc(Table)
Actual type String
Value loaded from sub

Bug: Comparison of String objects using == or != in TestCaseController.makeTc(Table)

This code compares `java.lang.String` objects for reference equality using the == or != operators. Unless both strings are either constants in a source file, or have been interned using the `String.intern()` method, the same string value may be represented by two different String objects. Consider using the `equals(Object)` method instead.

Rank: Troubling (11), **confidence**: Normal
Pattern: ES_COMPARING_STRINGS_WITH_EQ
Type: ES, **Category**: BAD_PRACTICE (Bad practice)
