



# Category Partition Testing Tool

201414134 오세욱

201414136 임현유

201211375 임동현

201211387 하헌규

# Index

---

**Static Analysis Response**

**Demonstration**

**Epilogue**

# Static Analysis Response

## Test Result

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
Package	359	164.2	5.61	SoftwareMo...
UL.java	26	178.1	13.00	SoftwareMo...
ConstructorCallsOverridableMethod	3	20.5	1.50	SoftwareMo...
SystemPrintln	2	13.7	1.00	SoftwareMo...
VariableNamingConventions	21	143.8	10.50	SoftwareMo...
Test.java	62	76.5	6.20	SoftwareMo...
ConstructorCallsOverridableMethod	1	1.2	0.10	SoftwareMo...
SystemPrintln	5	6.2	0.50	SoftwareMo...
VariableNamingConventions	48	59.3	4.80	SoftwareMo...
MethodNamingConventions	8	9.9	0.80	SoftwareMo...
Test.java	62	76.5	6.20	SoftwareMo...
ConstructorCallsOverridableMethod	1	1.2	0.10	SoftwareMo...
SystemPrintln	5	6.2	0.50	SoftwareMo...
VariableNamingConventions	48	59.3	4.80	SoftwareMo...
MethodNamingConventions	8	9.9	0.80	SoftwareMo...
TC.java	12	226.4	6.00	SoftwareMo...
SystemPrintln	4	75.5	2.00	SoftwareMo...
VariableNamingConventions	6	113.2	3.00	SoftwareMo...
MethodNamingConventions	2	37.7	1.00	SoftwareMo...
RV.java	37	698.1	3.70	SoftwareMo...
VariableNamingConventions	26	490.6	2.60	SoftwareMo...
MethodNamingConventions	11	207.5	1.10	SoftwareMo...
Output_Manager.java	15	625.0	3.75	SoftwareMo...
VariableNamingConventions	8	333.3	2.00	SoftwareMo...
MethodNamingConventions	6	250.0	1.50	SoftwareMo...
SystemPrintln	1	41.7	0.25	SoftwareMo...
Input_Manager.java	99	811.5	5.50	SoftwareMo...
SystemPrintln	2	16.4	0.11	SoftwareMo...
VariableNamingConventions	65	532.8	3.61	SoftwareMo...
MethodNamingConventions	32	262.3	1.78	SoftwareMo...
Category.java	18	600.0	3.00	SoftwareMo...
VariableNamingConventions	14	466.7	2.33	SoftwareMo...
MethodNamingConventions	4	133.3	0.67	SoftwareMo...
Calculate.java	28	202.9	14.00	SoftwareMo...
SystemPrintln	4	29.0	2.00	SoftwareMo...
VariableNamingConventions	22	159.4	11.00	SoftwareMo...
MethodNamingConventions	2	14.5	1.00	SoftwareMo...

8조의 PMD rule에  
맞춰서 수정

# PMD

## Static Analysis Response

### Test Result

VariableNamingConventions - 258개

MethodNamingConventions - 73개

ConstructorCallsOverridableMethod - 5개

SystemPrintln - 23개

Error - 359개



**Class Diagram에 정의한 내용대로 유지하기 위해 사용**



**test class에서 발생한 warning -> test class 삭제**



**구현 중 기능확인을 위해 System.out.println을 사용하여 발생한warning-> 해당 code 삭제**

## Test Result

Found a 7 line (29 tokens) duplication in the following files:

Starting at line 61 of C:\Users\sunggeun\workspace\T7\_source\SoftwareModeling\src\unit\test\Input\_ManagerTest.java  
Starting at line 86 of C:\Users\sunggeun\workspace\T7\_source\SoftwareModeling\src\unit\test\Input\_ManagerTest.java  
Starting at line 139 of C:\Users\sunggeun\workspace\T7\_source\SoftwareModeling\src\unit\test\Input\_ManagerTest.java  
Starting at line 304 of C:\Users\sunggeun\workspace\T7\_source\SoftwareModeling\src\unit\test\Input\_ManagerTest.java  
Starting at line 323 of C:\Users\sunggeun\workspace\T7\_source\SoftwareModeling\src\unit\test\Input\_ManagerTest.java

```
public void testAdd_RV() {  
    Input_Manager im = new Input_Manager();  
  
    im.Add_Category();  
    im.Add_RV(0);  
    im.Add_RV(0);  
    im.Add_RV(0);
```



**Unit Testing을 위해 사용한  
Input\_ManagerTest.java에서  
초기값을 입력하기 위해 중복코드 사용**

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

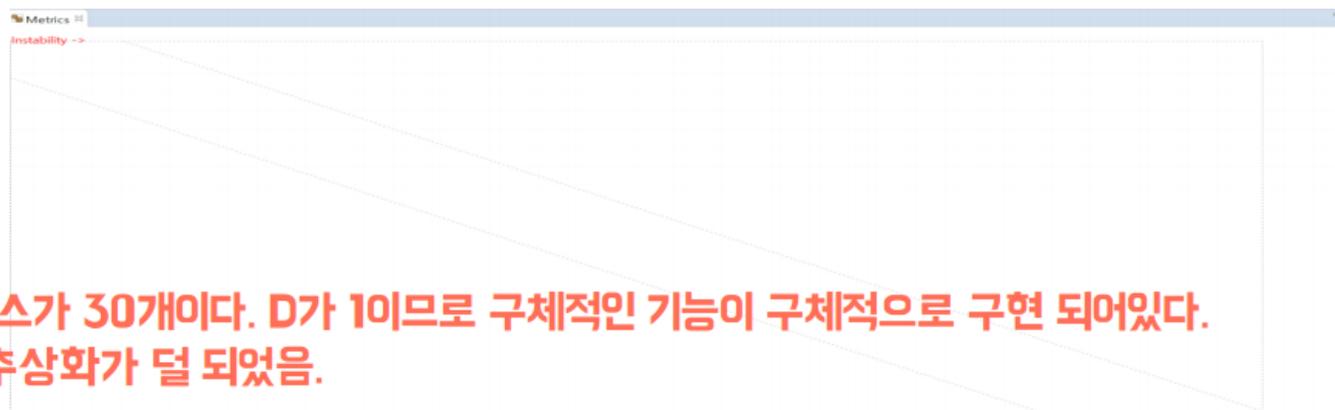
A: 0  
I: 0  
D: 1

### Depends Upon:

Not dependent on any packages.

### Used By:

Not used by any packages.



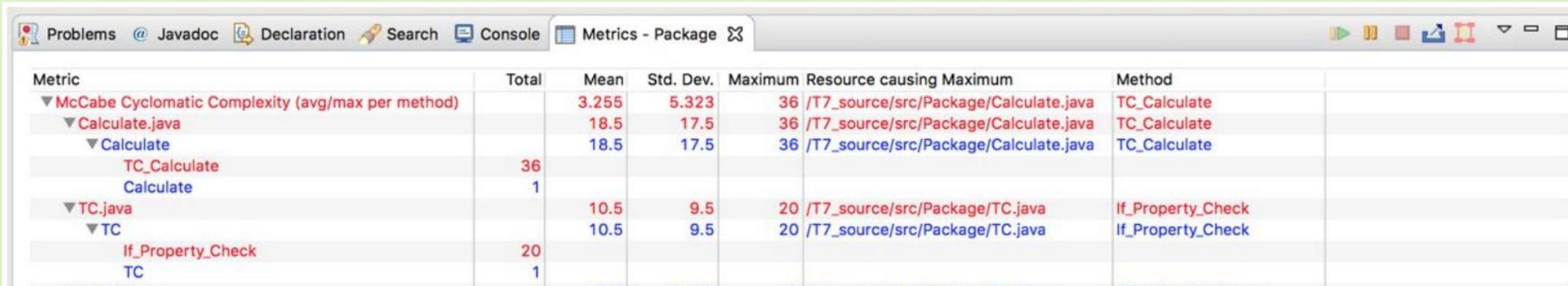
생성되는 클래스가 30개이다. D가 1이므로 구체적인 기능이 구체적으로 구현 되어있다.  
안정적이면서 추상화가 덜 되었음.

규모가 작은 프로젝트이기 때문에 추상클래스를 사용하면  
오히려 더 복잡해질 수 있다고 생각하여 추상클래스를 사용하지 않았다.

# Eclipse Metrics Plug-in

## Static Analysis Response

### Test Result



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
▼ McCabe Cyclomatic Complexity (avg/max per method)		3.255	5.323	36	/T7_source/src/Package/Calculate.java	TC_Calculate
▼ Calculate.java		18.5	17.5	36	/T7_source/src/Package/Calculate.java	TC_Calculate
▼ Calculate		18.5	17.5	36	/T7_source/src/Package/Calculate.java	TC_Calculate
TC_Calculate	36					
Calculate	1					
▼ TC.java		10.5	9.5	20	/T7_source/src/Package/TC.java	If_Property_Check
▼ TC		10.5	9.5	20	/T7_source/src/Package/TC.java	If_Property_Check
If_Property_Check	20					
TC	1					

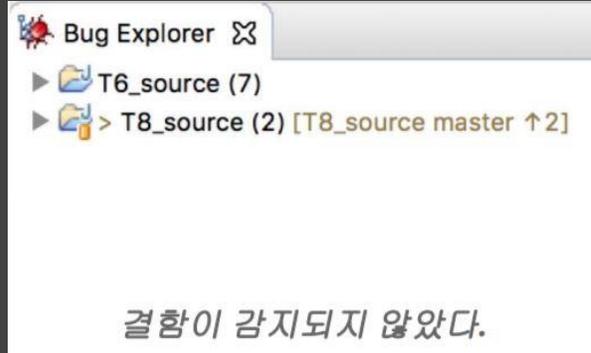
```
for(int i =Category_List.size() -1 ; i>=0 ;i--){  
  
    Category category = Category_List.get(i);  
    if(category.RV_List.size() != 0){  
        for(int l=0;l<(Whole_TC_Count/(tmp_TC_num*category.RV_List.size())); l++){  
  
            for(int j=0;j<category.RV_List.size();j++){  
  
                RV value = category.RV_List.get(j);  
  
                for(int k=0;k<tmp_TC_num;k++){  
                    All_TC.get((1*tmp_TC_num*category.RV_List.size()+j*tmp_TC_num+k).Data.add(0,value);  
                }  
            }  
        }  
    }  
}
```

```
for(Object tc : All_TC){  
    TC testcase = (TC)tc;  
  
    for(RV selected : Selected_RV_List){  
        Boolean b =false;  
        for(int i=0;i<testcase.Data.size();i++){  
            if(testcase.Data.get(i).RV_Id==selected.RV_Id &&testcase.Data.get(i).category.Category_Id==selected.category.Category_Id){  
                Selected_TC.add(testcase);  
                b=true;  
                break;  
            }  
        }  
        if(b)break;  
    }  
}
```

문서에 하나의 Method로 계산을 하도록 명시되어 있기 때문에 더 이상 Method를 추가하지 않고 구현  
+ 추가 예제와 관련된 기능을 추가 -> Cyclomatic Complexity증가

# Find Bugs

## Static Analysis Response



### 7조 분석 결과

FindBug를 사용하여 분석한 결과 아무 warning도 발생하지 않아서인지, 버그리포팅 파일이 생성되지 않았다.

# Find Bugs

# Static Analysis Response

 Bug Explorer 

-  T6\_source (7)
-  > T8\_source (2) [T8\_source master ↑2]

결함이 감지되지 않았다.

## 7조 분석 결과

FindBug를 사용하여 분석한 결과 아무 warning도 발생하지 않아서인지, 버그리포트링 파일이 생성되지 않았다.

```
<BugInstance type="NM_METHOD_NAMING_CONVENTION" priority="2" rank="16" abbrev="Nm" category="BAD_PRACTICE" first="1">
  <Class classname="Package.Input_Manager">
    <SourceLine classname="Package.Input_Manager" sourcefile="Input_Manager.java" sourcepath="Package/Input_Manager.java"/>
  </Class>
  <Method classname="Package.Input_Manager" name="Calculate" signature="()V" isStatic="false">
    <SourceLine classname="Package.Input_Manager" start="159" end="161" startBytecode="0" endBytecode="76" sourcefile="Input_Manager.java" sourcepath="Package/Input_Manager.java"/>
  </Method>
</BugInstance>
<BugInstance type="NM_METHOD_NAMING_CONVENTION" priority="2" rank="16" abbrev="Nm" category="BAD_PRACTICE" first="1">
  <Class classname="Package.Input_Manager">
    <SourceLine classname="Package.Input_Manager" sourcefile="Input_Manager.java" sourcepath="Package/Input_Manager.java"/>
  </Class>
  <Method classname="Package.Input_Manager" name="Clear" signature="()V" isStatic="false">
    <SourceLine classname="Package.Input_Manager" start="165" end="169" startBytecode="0" endBytecode="95" sourcefile="Input_Manager.java" sourcepath="Package/Input_Manager.java"/>
  </Method>
</BugInstance>
<BugInstance type="DM_DEFAULT_ENCODING" priority="1" rank="19" abbrev="Dm" category="I18N" first="1">
  <Class classname="Package.Output_Manager">
    <SourceLine classname="Package.Output_Manager" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  </Class>
  <Method classname="Package.Output_Manager" name="Output_File" signature="(Ljava/util/List;)V" isStatic="false">
    <SourceLine classname="Package.Output_Manager" start="27" end="41" startBytecode="0" endBytecode="405" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  </Method>
  <Method classname="java.io.FileWriter" name="&lt;init&gt;" signature="(Ljava/lang/String;)V" isStatic="false" role="METHOD_CALLED">
    <SourceLine classname="java.io.FileWriter"/>
  </Method>
  <SourceLine classname="Package.Output_Manager" start="27" end="27" startBytecode="10" endBytecode="10" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  <SourceLine classname="Package.Output_Manager" start="27" end="27" startBytecode="10" endBytecode="10" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
</BugInstance>
<BugInstance type="DM_EXIT" priority="2" rank="16" abbrev="Dm" category="BAD_PRACTICE" first="1">
  <Class classname="Package.Output_Manager">
    <SourceLine classname="Package.Output_Manager" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  </Class>
  <Method classname="Package.Output_Manager" name="Output_File" signature="(Ljava/util/List;)V" isStatic="false">
    <SourceLine classname="Package.Output_Manager" start="27" end="41" startBytecode="0" endBytecode="405" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  </Method>
  <SourceLine classname="Package.Output_Manager" start="39" end="39" startBytecode="148" endBytecode="148" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  <SourceLine classname="Package.Output_Manager" start="39" end="39" startBytecode="148" endBytecode="148" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
</BugInstance>
<BugInstance type="URF_UNREAD_PUBLIC_OR_PROTECTED_FIELD" priority="2" rank="18" abbrev="Urf" category="STYLE" first="1">
  <Class classname="Package.Output_Manager">
    <SourceLine classname="Package.Output_Manager" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  </Class>
  <Field classname="Package.Output_Manager" name="All_TC" signature="Ljava/util/List;" isStatic="false">
    <SourceLine classname="Package.Output_Manager" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
  </Field>
  <SourceLine classname="Package.Output_Manager" start="16" end="16" startBytecode="23" endBytecode="23" sourcefile="Output_Manager.java" sourcepath="Package/Output_Manager.java"/>
</BugInstance>
<BugInstance type="NM_FIELD_NAMING_CONVENTION" priority="2" rank="16" abbrev="Nm" category="BAD_PRACTICE" first="1">
  <Class classname="Package.TC">
    <SourceLine classname="Package.TC" sourcefile="TC.java" sourcepath="Package/TC.java"/>
  </Class>
  <Field classname="Package.TC" name="Data" signature="Ljava/util/List;" isStatic="false">
    <SourceLine classname="Package.TC" sourcefile="TC.java" sourcepath="Package/TC.java"/>
  </Field>
</BugInstance>
```

# Find Bugs

## Static Analysis Response

```
<FindBugsSummary timestamp="Mon, 5 Jun 2017 17:26:44 +0900" total_classes="31" referenced_classes="317" total_bugs="6" total_size="1353"
```



```
<FindBugsSummary timestamp="Mon, 5 Jun 2017 18:38:17 +0900" total_classes="31" referenced_classes="320" total_bugs="3" total_size="1350"
```

```
public List<TC> All_TC;
```

UrF: Unread field (URF\_UNREAD\_FIELD)

사용하지 않는 필드 제거

This field is never read. Consider removing it from the class.

```
BufferedWriter out = new BufferedWriter(new FileWriter("out.txt"));
```

Dm: Reliance on default encoding (DM\_DEFAULT\_ENCODING)

File encoding 방식 수정

Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assume that the default platform encoding is suitable. This will cause the application behaviour to vary between platforms. Use an alternative API and specify a charset name or Charset object explicitly.

```
System.exit(1);
```

Dm: Method invokes System.exit(...) (DM\_EXIT)

exit 명령어제거

Invoking System.exit shuts down the entire Java virtual machine. This should only be done when it is appropriate. Such calls make it hard or impossible for your code to be invoked by other code. Consider throwing a RuntimeException instead.

나머지 3개의 bug는 naming관련문제로 PMD에서 이미 확인했다.

**Demonstration**

# Epilogue

# OOPT\_Pro & Cons

팀명	팀원	개인 숙제 #1	개인 숙제 #2	팀발표 #1	팀발표 #2	팀발표 #3	팀발표 #4
		Category Partitioning Test 실습	Category Partitioning Test 이론	OOPT Stage 1000	OOPT Stage 2030	OOPT Stage 2040	OOPT Stage 2050/2060
		CPT 실습결과 보고서 (~03.11 23:00)	<a href="#">CACM 1988</a> 논문 정리 (~03.18 23:00)	Planning	Analysis	Design	Implementation & Unit Test
T7	201414134 오세욱 201414136 임현유 201211375 임동현 201211387 하현규 	<a href="#">오세욱</a> <a href="#">임현유</a> <a href="#">임동현</a> <a href="#">하현규</a>	<a href="#">오세욱</a> <a href="#">임현유</a> <a href="#">임동현</a> <a href="#">하현규</a>	<a href="#">발표자료</a> <a href="#">보고서 v2 v3 v4</a> <a href="#">wmv</a>	<a href="#">발표자료</a> <a href="#">보고서 v2 v3</a> wmv	<a href="#">발표자료</a> <a href="#">보고서 v2</a> <a href="#">wmv</a>	<a href="#">발표자료</a> <a href="#">보고서</a> <a href="#">program</a> <a href="#">wmv</a>

1. 문서 량 多 -> 구현을 시작하기까지 오래 걸림
2. 문서 수정에 대한 부담
3. 구현 시 문서의 틀에 맞춰서 구현 -> detail한 문서 작성이 필요
4. 소규모 프로젝트에서의 비효율성

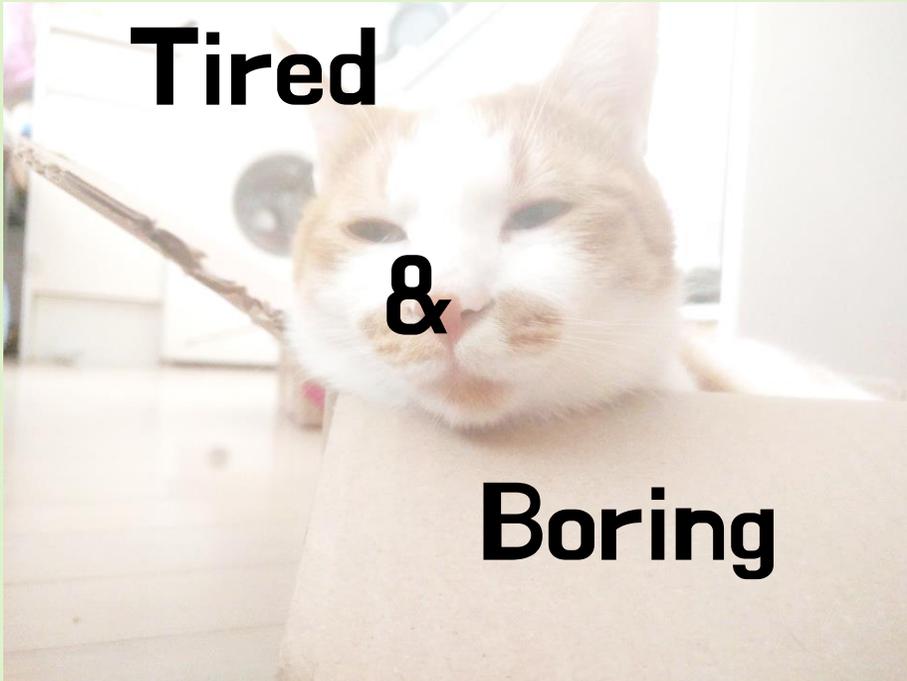


1. 1000단계에서 risk analysis를 통해 프로젝트 진행 시 위험 요소에 대한 대비 가능 (ex. 고양이가 아픔, 아르바이트 때문에 시간이 부족함)
2. 체계적인 문서를 통해서 단계별 진행상황을 파악하기 용이
3. 대형 프로젝트라면 효율적

# OOPT\_Profs & Cons

Epilogue

**Before**



**After**



Thank you ~!