

# 1.checkStyle

## 1.1 설명

- CheckStyle은 코딩 표준에 맞게 소스코드를 작성하도록 도와주는 도구이다.
- 개발자들이 잘 지키지는 않지만 중요한 것들을 지적해준다 (ex : private 설정)
- 협업시에 코딩 스타일을 맞추 수 있다.

## 1.2 결과

### 1.2.1 총 결과

#### Warnings Trend

All Warnings	New Warnings	Fixed Warnings
2841	0	0

#### Summary

Total	High Priority	Normal Priority	Low Priority
2841	0	<a href="#">2841</a>	0

### 1.2.2 타입 별 Warnings

#### Details

Files	Categories	Types	Warnings	Details																																																															
<table border="1"><thead><tr><th>Type</th><th>Total</th><th>Distribution</th></tr></thead><tbody><tr><td><a href="#">ArrayTypeStyleCheck</a></td><td>2</td><td>■</td></tr><tr><td><a href="#">AvoidStarImportCheck</a></td><td>7</td><td>■</td></tr><tr><td><a href="#">CustomImportOrderCheck</a></td><td>1</td><td>■</td></tr><tr><td><a href="#">EmptyLineSeparatorCheck</a></td><td>33</td><td>■</td></tr><tr><td><a href="#">FileTabCharacterCheck</a></td><td>1163</td><td>■</td></tr><tr><td><a href="#">GenericWhitespaceCheck</a></td><td>1</td><td>■</td></tr><tr><td><a href="#">IndentationCheck</a></td><td>971</td><td>■</td></tr><tr><td><a href="#">JavadocMethodCheck</a></td><td>54</td><td>■</td></tr><tr><td><a href="#">LeftCurlyCheck</a></td><td>73</td><td>■</td></tr><tr><td><a href="#">LineLengthCheck</a></td><td>91</td><td>■</td></tr><tr><td><a href="#">LocalVariableNameCheck</a></td><td>20</td><td>■</td></tr><tr><td><a href="#">MemberNameCheck</a></td><td>1</td><td>■</td></tr><tr><td><a href="#">MethodNameCheck</a></td><td>2</td><td>■</td></tr><tr><td><a href="#">MultipleVariableDeclarationsCheck</a></td><td>7</td><td>■</td></tr><tr><td><a href="#">NeedBracesCheck</a></td><td>2</td><td>■</td></tr><tr><td><a href="#">ParameterNameCheck</a></td><td>15</td><td>■</td></tr><tr><td><a href="#">RightCurlyCheck</a></td><td>25</td><td>■</td></tr><tr><td><a href="#">VariableDeclarationUsageDistanceCheck</a></td><td>1</td><td>■</td></tr><tr><td><a href="#">WhitespaceAroundCheck</a></td><td>372</td><td>■</td></tr><tr><td>Total</td><td>2841</td><td></td></tr></tbody></table>					Type	Total	Distribution	<a href="#">ArrayTypeStyleCheck</a>	2	■	<a href="#">AvoidStarImportCheck</a>	7	■	<a href="#">CustomImportOrderCheck</a>	1	■	<a href="#">EmptyLineSeparatorCheck</a>	33	■	<a href="#">FileTabCharacterCheck</a>	1163	■	<a href="#">GenericWhitespaceCheck</a>	1	■	<a href="#">IndentationCheck</a>	971	■	<a href="#">JavadocMethodCheck</a>	54	■	<a href="#">LeftCurlyCheck</a>	73	■	<a href="#">LineLengthCheck</a>	91	■	<a href="#">LocalVariableNameCheck</a>	20	■	<a href="#">MemberNameCheck</a>	1	■	<a href="#">MethodNameCheck</a>	2	■	<a href="#">MultipleVariableDeclarationsCheck</a>	7	■	<a href="#">NeedBracesCheck</a>	2	■	<a href="#">ParameterNameCheck</a>	15	■	<a href="#">RightCurlyCheck</a>	25	■	<a href="#">VariableDeclarationUsageDistanceCheck</a>	1	■	<a href="#">WhitespaceAroundCheck</a>	372	■	Total	2841	
Type	Total	Distribution																																																																	
<a href="#">ArrayTypeStyleCheck</a>	2	■																																																																	
<a href="#">AvoidStarImportCheck</a>	7	■																																																																	
<a href="#">CustomImportOrderCheck</a>	1	■																																																																	
<a href="#">EmptyLineSeparatorCheck</a>	33	■																																																																	
<a href="#">FileTabCharacterCheck</a>	1163	■																																																																	
<a href="#">GenericWhitespaceCheck</a>	1	■																																																																	
<a href="#">IndentationCheck</a>	971	■																																																																	
<a href="#">JavadocMethodCheck</a>	54	■																																																																	
<a href="#">LeftCurlyCheck</a>	73	■																																																																	
<a href="#">LineLengthCheck</a>	91	■																																																																	
<a href="#">LocalVariableNameCheck</a>	20	■																																																																	
<a href="#">MemberNameCheck</a>	1	■																																																																	
<a href="#">MethodNameCheck</a>	2	■																																																																	
<a href="#">MultipleVariableDeclarationsCheck</a>	7	■																																																																	
<a href="#">NeedBracesCheck</a>	2	■																																																																	
<a href="#">ParameterNameCheck</a>	15	■																																																																	
<a href="#">RightCurlyCheck</a>	25	■																																																																	
<a href="#">VariableDeclarationUsageDistanceCheck</a>	1	■																																																																	
<a href="#">WhitespaceAroundCheck</a>	372	■																																																																	
Total	2841																																																																		

### 1.2.3 타입 별 Warnings 자세한 결과

### 1.2.3.1 ArrayTypeStyleCheck

File	Total	Distribution
<a href="#">Calculate.java</a>	1	
<a href="#">Controller.java</a>	1	
Total	2	

#### Array brackets at illegal position.

Checks the style of array type definitions. Some like Java style: `public static void main(String[] args)` and some like C style: `public static void main(String args[])`

### 1.2.3.2 AvoidStarImportCheck

File	Total	Distribution
<a href="#">Calculate.java</a>	1	
<a href="#">Controller.java</a>	5	
<a href="#">Files.java</a>	1	
Total	7	

#### Using the '\*' form of import should be avoided - java.util.\*.

Checks that there are no import statements that use the \* notation.

Rationale: Importing all classes from a package or static members from a class leads to tight coupling between packages or classes and might lead to problems when a new version of a library introduces name clashes.

### 1.2.3.3 CustomImportOrderCheck

[Controller.java:3](#), CustomImportOrderCheck, Priority: Normal

#### Wrong lexicographical order for 'java.awt.\*' import.

Checks that the groups of import declarations appear in the order specified by the user. If there is an import but its group is not specified in the configuration such an import should be placed at the end of the import list.

[Examples section](#) contains examples that work with default formatter configurations of Eclipse, IntelliJ IDEA and NetBeans

### 1.2.3.4 EmptyLineSeparatorCheck

File	Total	Distribution
<a href="#">Calculate.java</a>	13	
<a href="#">Controller.java</a>	3	
<a href="#">Files.java</a>	17	
Total	33	

#### 'METHOD\_DEF' should be separated from previous statement.

Checks for empty line separators after header, package, all import declarations, fields, constructors, methods, nested classes, static initializers and instance initializers.

### 1.2.3.5 FileTabCharacterCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	277	
<a href="#">Calculate.java</a>	318	
<a href="#">Controller.java</a>	476	
<a href="#">Files.java</a>	88	
<a href="#">Main.java</a>	4	
Total	1163	

Line contains a tab character.

Checks that there are no tab characters ( '\t ' ) in the source code.

Rationale:

- Developers should not need to configure the tab width of their text editors in order to be able to read source code.
- From the Apache jakarta coding standards: In a distributed development environment, when the commit messages get sent to a mailing list, they are almost impossible to read if you use tabs.

### 1.2.3.6 GenericWhitespaceCheck

[Calculate.java:11](#), GenericWhitespaceCheck, Priority: Normal

**GenericWhitespace '<' is preceded with whitespace.**

Checks that the whitespace around the Generic tokens (angle brackets) "<" and ">" are correct to the *typical* convention. The convention is not configurable.

Left angle bracket ("<"):

- should be preceded with whitespace only in generic methods definitions.
- should not be preceded with whitespace when it is precede method name or following type name.
- should not be followed with whitespace in all cases.

Right angle bracket (">"):

- should not be preceded with whitespace in all cases.
- should be followed with whitespace in almost all cases, except diamond operators and when preceding method name.

Examples with correct spacing:

```
public void <K, V extends Number> boolean foo(K, V) {} // Generic methods definitions
class name<T1, T2, ..., Tn> {} // Generic type definition
OrderedPair<String, Box<Integer>> p; // Generic type reference
boolean same = Util.<Integer, String>compare(p1, p2); // Generic preceded
method name Pair<Integer, String> p1 = new Pair<>(1, "apple"); // Diamond operator
List<T> list = ImmutableList.Builder<T>::new; // Method reference
sort(list, Comparable::<String>compareTo); // Method reference
```

### 1.2.3.7 IndentationCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	262	
<a href="#">Calculate.java</a>	193	
<a href="#">Controller.java</a>	428	
<a href="#">Files.java</a>	85	
<a href="#">Main.java</a>	3	
Total	971	

'member def type' have incorrect indentation level 8, expected level should be 2.

Checks correct indentation of Java code.

The idea behind this is that while pretty printers are sometimes convenient for bulk reformat of legacy code, they often either aren't configurable enough or just can't anticipate how format should be done. Sometimes this is personal preference, other times it is practical experience. In any case, this check should just ensure that a minimal set of indentation rules is followed.

### 1.2.3.8 JavadocMethodCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	2	
<a href="#">Calculate.java</a>	14	
<a href="#">Controller.java</a>	18	
<a href="#">Files.java</a>	19	
<a href="#">Main.java</a>	1	
Total	54	

#### Missing a Javadoc comment.

Checks the Javadoc of a method or constructor. By default, does not check for unused throws. To allow documented `java.lang.RuntimeExceptions` that are not declared, set property `allowUndeclaredRTE` to true. The scope to verify is specified using the `Scope` class and defaults to `Scope.PRIVATE`. To verify another scope, set property `scope` to a different [scope](#).

Error messages about parameters and type parameters for which no param tags are present can be suppressed by defining property `allowMissingParamTags`. Error messages about exceptions which are declared to be thrown, but for which no throws tag is present can be suppressed by defining property `allowMissingThrowsTags`. Error messages about methods which return non-void but for which no return tag is present can be suppressed by defining property `allowMissingReturnTag`.

Javadoc is not required on a method that is tagged with the `@Override` annotation. However under Java 5 it is not possible to mark a method required for an interface (this was corrected under Java 6). Hence Checkstyle supports using the convention of using a single `{@inheritDoc}` tag instead of all the other tags.

Note that only inheritable items will allow the `{@inheritDoc}` tag to be used in place of comments. Static methods at all visibilities, private non-static methods and constructors are not inheritable.

For example, if the following method is implementing a method required by an interface, then the Javadoc could be done as:

```
/** {@inheritDoc} */
public int checkReturnTag(final int aTagIndex,
                        JavadocTag[] aTags,
                        int aLineNo)
```

The classpath may need to be configured to locate the class information. The classpath configuration is dependent on the mechanism used to invoke Checkstyle.

#### 1.2.3.9 LeftCurlyCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	5	
<a href="#">Calculate.java</a>	28	
<a href="#">Controller.java</a>	20	
<a href="#">Files.java</a>	20	
Total	73	

#### '{' should be on the previous line.

Checks for the placement of left curly braces ( '{ ' ) for code blocks. The policy to verify is specified using the property `option`. Policies `eo1` and `nlw` take into account the property `maxLineLength`.

#### 1.2.3.10 LineLengthCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	23	
<a href="#">Calculate.java</a>	41	
<a href="#">Controller.java</a>	27	



[Analyze.java:61](#). LineLengthCheck, Priority: Normal

#### Line is longer than 100 characters (found 143).

Checks for long lines.

Rationale: Long lines are hard to read in printouts or if developers have limited screen space for the source code, e.g. if the IDE displays additional information like project tree, class hierarchy, etc.

### 1.2.3.11 LocalVariableNameCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	12	
<a href="#">Calculate.java</a>	8	
<b>Total</b>	<b>20</b>	

<a href="#">Analyze.java:56</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:56</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:73</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:73</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:123</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:136</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:146</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:177</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:201</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:242</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:253</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Analyze.java:253</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.
<a href="#">Calculate.java:70</a> , LocalVariableNameCheck, Priority: Normal
Local variable name 'Y' must match pattern "[a-z][a-z0-9]*[A-Z0-9]".
Checks that local, non-final variable names conform to a format specified by the format property.

[Calculate.java-71](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'j' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

[Calculate.java-132](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'i' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

[Calculate.java-133](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'j' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

[Calculate.java-203](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'PreDiff' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

[Calculate.java-204](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'PreDiff1' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

[Calculate.java-206](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'i' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

[Calculate.java-207](#), LocalVariableNameCheck, Priority: Normal

**Local variable name 'j' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Checks that local, non-`final` variable names conform to a format specified by the format property.

### 1.2.3.12 MemberNameCheck

[Calculate.java:Z](#), MemberNameCheck, Priority: Normal

**Member name 'SyncRate' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9]*$`.**

Validates identifiers for non-`static` fields.

### 1.2.3.13 MethodNameCheck

File	Line	Priority	Type	Category
<a href="#">Controller.java:117</a>	117	Normal	MethodNameCheck	Naming
<a href="#">Controller.java:211</a>	211	Normal	MethodNameCheck	Naming

Method name 'DisplayMain' must match pattern `^[a-z][a-z0-9][a-zA-Z0-9_]*$`.

Validates identifiers for methods.

### 1.2.3.14 MultipleVariableDeclarationsCheck



File	Total	Distribution
<a href="#">Analyze.java</a>	4	
<a href="#">Controller.java</a>	3	
Total	7	

Each variable declaration must be in its own statement.

Checks that each variable declaration is in its own statement and on its own line.

Rationale: [the Java code conventions chapter 6.1](#) recommends that declarations should be one per line/statement.

### 1.2.3.15 NeedBracesCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	1	
<a href="#">Controller.java</a>	1	
Total	2	

'if' construct must use '{}'s.

Checks for braces around code blocks.

### 1.2.3.16 ParameterNameCheck

File	Line	Priority	Type	Category
<a href="#">Controller.java:155</a>	155	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:171</a>	171	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:255</a>	255	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:264</a>	264	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:412</a>	412	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:426</a>	426	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:440</a>	440	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:501</a>	501	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:501</a>	501	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:506</a>	506	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:506</a>	506	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:506</a>	506	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:506</a>	506	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:513</a>	513	Normal	ParameterNameCheck	Naming
<a href="#">Controller.java:517</a>	517	Normal	ParameterNameCheck	Naming

<p><a href="#">Controller.java-1</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-171</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-255</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-264</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-412</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-426</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-440</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-501</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-501</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'g' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-506</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'e' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-506</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'g' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-506</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'x' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-506</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'y' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-513</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'c' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p> <p><a href="#">Controller.java-517</a>. ParameterNameCheck, Priority: Normal</p> <p>Parameter name 'c' must match pattern <code>^[a-z][a-z0-9][a-zA-Z0-9]*\$</code>.</p> <p>Checks that method and catch parameter names conform to a format specified by the format property.</p>
--

### 1.2.3.17 RightCurlyCheck

File	Total	Distribution
<a href="#">Analysis.java</a>	7	

} should be on the same line.

Checks the placement of right curly braces ( } ) for if-else, try-catch-finally blocks, while-loops, for-loops, method definitions, class definitions, constructor definitions, instance and static initialization blocks. The policy to verify is specified using the property `opt.Lon`.






### 1.2.3.18 VariableDeclarationUsageDistanceCheck

[Controller.java:369](#), VariableDeclarationUsageDistanceCheck, Priority: Normal

Distance between variable 'lbTempPercent' declaration and its first usage is 4, 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).

Checks the distance between declaration of variable and its first usage.

### 1.2.3.19 WhitespaceAroundCheck

File	Total	Distribution
<a href="#">Analyze.java</a>	260	
<a href="#">Calculate.java</a>	74	
<a href="#">Controller.java</a>	38	
Total	372	

**WhitespaceAround: '<' is not preceded with whitespace.**

Checks that a token is surrounded by whitespace. Empty constructor, method, class, enum, interface, loop bodies (blocks), lambdas of the form

```
public MyClass() {} // empty constructor
public void func() {} // empty method
public interface Foo {} // empty interface
public class Foo {} // empty class
public enum Foo {} // empty enum
MyClass c = new MyClass() {}; // empty anonymous class
while (i = 1) {} // empty while loop
for (int i = 1; i > 1; i++) {} // empty for loop
do {} while (i = 1); // empty do-while loop
Runnable noop = () -> {}; // empty lambda
@interface Beta {} // empty annotation type
```

may optionally be exempted from the policy using the `allowEmptyMethods`, `allowEmptyConstructors`, `allowEmptyTypes`, `allowEmptyLoops` and `allowEmptyLambdas` properties.

This check does not flag as violation double brace initialization like:

```
new Properties() {{
    setProperty("key", "value");
}};
```

## 2.PMD

### 2.1설명

- Program May Dependable의 약자
- 정해진 규칙에 따라 소스코드를 검사해서 위반사항을 찾아낼 수 있다.
- Eclipse -plugin으로 사용할 수 있으며, 별도로 커맨드 라인 에서 사용 가능하다.
- 분석 대상은 java, javaScript, XML 등이 있다.
- PMD에서 제공하는 카테고리로부터, 분석 대상 프로그램 언어인 Java에 맞는카테고리 선정

### 2.2 결과

## 2.2.1 총 결과

### Warnings Trend

All Warnings	New Warnings	Fixed Warnings
16	0	0

### Summary

Total	High Priority	Normal Priority	Low Priority
16	0	16	0

## 2.2.2 타입 별 Warnings

Type	Total	Distribution
<a href="#">CollapsibleIfStatements</a>	3	
<a href="#">JumbledIncrementer</a>	2	
<a href="#">UselessParentheses</a>	11	
Total	16	

## 2.2.3 타입별 Warnings 자세한 설명

### 2.2.3.1 CollapsibleIfStatements

File	Total	Distribution
<a href="#">Analyze.java</a>	2	
<a href="#">Controller.java</a>	1	
Total	3	

**These nested if statements could be combined.**

Sometimes two consecutive 'if' statements can be consolidated by separating their conditions with a boolean short-circuit operator.

```
void bar() {
    if (x) {
        if (y) {
            // do stuff
        }
    }
}

void bar() {
    if (x && y) {
        // do stuff
    }
}
```

### 2.2.3.2 JumbledIncrementer

File	Line	Priority	Type	Category
<a href="#">Controller.java:414</a>	414	Normal	JumbledIncrementer	Basic
<a href="#">Controller.java:427</a>	427	Normal	JumbledIncrementer	Basic

Avoid modifying an outer loop incremter in an inner loop for update expression.

Avoid jumbled loop incremterers - its usually a mistake, and is confusing even if intentional.

```
public class JumbledIncrementerRule1 {
    public void foo() {
        for (int i = 0; i < 10; i++) {
            for (int k = 0; k < 20; i++) { // only references 'i'
                System.out.println("Hello"); // references both 'i' and 'k'
            }
        }
    }
}
```

### 2.2.3.3 UselessParentheses

File	Total	Distribution
<a href="#">Analyze.java</a>	1	
<a href="#">Calculate.java</a>	6	
<a href="#">Controller.java</a>	4	
Total	11	

```
public class Foo {
    private int _bar1;
    private Integer _bar2;

    public void setBar(int n) {
        _bar1 = Integer.valueOf((n)); // here
        _bar2 = (n); // and here
    }
}
```

## 3. FindBugs

### 3.1 설명

- 자바프로그램 분석도구로, 메릴랜드 대학에서 개발.
- Data flow, Control flow 분석 등을 이용한다.
- 버그 패턴을 자동으로 찾아서 알려 준다.

## 3.2 결과

### 3.2.1 총 결과

#### Warnings Trend

All Warnings	New this build	Fixed Warnings
10	0	0

#### Summary

Total	High Priority	Normal Priority	Low Priority
10	1	2	0

### 3.2.2 Type 별 결과

Type	Total	Distribution
<a href="#">DM_DEFAULT_ENCODING</a>	1	
<a href="#">DM_STRING_CTOR</a>	3	
<a href="#">NM_METHOD_NAMING_CONVENTION</a>	2	
<a href="#">NP_NULL_ON_SOME_PATH_FROM_RETURN_VALUE</a>	1	
<a href="#">SIC_INNER_SHOULD_BE_STATIC</a>	1	
<a href="#">UUF_UNUSED_FIELD</a>	1	
<a href="#">UWF_UNWRITTEN_FIELD</a>	1	
Total	10	

### 3.2.3 Type 별 자세한 결과

#### 3.2.3.1 DM\_DEFAULT\_ENCODING

Controller.java:97, DM_DEFAULT_ENCODING, Priority: High
<b>Dm:</b> Found reliance on default encoding in Controller.mkFileInstance(): new java.io.FileReader(String)
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.

#### 3.2.3.2 DM\_STRING\_CTOR

File	Line	Priority	Rank	Type	Category
<a href="#">Calculate.java:85</a>	85	Normal	18	DM_STRING_CTOR	PERFORMANCE
<a href="#">Calculate.java:147</a>	147	Normal	18	DM_STRING_CTOR	PERFORMANCE
<a href="#">Calculate.java:221</a>	221	Normal	18	DM_STRING_CTOR	PERFORMANCE

#### Dm: Calculate.calFunction() invokes inefficient new String(String) constructor

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.

### 3.2.3.3 NM\_METHOD\_NAMING\_CONVENTION

File	Line	Priority	Rank	Type	Category
<a href="#">Controller.java:119</a>	119	Normal	16	NM_METHOD_NAMING_CONVENTION	BAD_PRACTICE
<a href="#">Controller.java:213</a>	213	Normal	16	NM_METHOD_NAMING_CONVENTION	BAD_PRACTICE

**Nm:** The method name `Controller.DisplayMain()` doesn't start with a lower case letter

Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

### 3.2.3.4 NP\_NULL\_ON\_SOME\_PATH\_FROM\_RETURN\_VALUE

[Controller.java:48](#), NP\_NULL\_ON\_SOME\_PATH\_FROM\_RETURN\_VALUE, Priority: Normal

**NP:** Possible null pointer dereference in `Controller.mkFileInstance()` due to return value of called method

The return value from a method is dereferenced without a null check, and the return value of that method is one that should generally be checked for null. This may lead to a `NullPointerException` when the code is executed.

### 3.2.3.5 SIC\_INNER\_SHOULD\_BE\_STATIC

[Controller.java:501](#), SIC\_INNER\_SHOULD\_BE\_STATIC, Priority: Normal

**SIC:** Should `Controller$RoundedBorder` be a `_static_ inner class`?

This class is an inner class, but does not use its embedded reference to the object which created it. This reference makes the instances of the class larger, and may keep the reference to the creator object alive longer than necessary. If possible, the class should be made static.

### 3.2.3.6 UUF\_UNUSED\_FIELD

[Analyze.java:1](#), UUF\_UNUSED\_FIELD, Priority: Normal

**UuF:** Unused field: `Analyze.tempFolderPath`

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

### 3.2.3.7 UWF\_UNWRITTEN\_FIELD

[Controller.java:437](#), UWF\_UNWRITTEN\_FIELD, Priority: Normal

**UwF:** Unwritten field: `Controller$5.IbTempTotalSyncRate`

This field is never written. All reads of it will return the default value. Check for errors (should it have been initialized?), or remove it if it is useless.

## 4.Code Coverage

## 4.1 Code Coverage Tool: JaCoCo (Java Code Coverage Library)

### 4.1.1 JaCoCo 설명

- 자바를 위한 무료 코드 커버리지 라이브러리
- 코드 상에서 Green line, Yellow line, Red line으로 표시가 되며 Green line은 직접 코드를 통과한 경우, Yellow line은 분기에서 모든 조건을 체크하지 않고 지나가는 경우, Red line은 아예 코드를 통과하지 않은 경우이다.

## 4.2 결과

### 4.2.1 총결과

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
default		38%		51%	119	200	382	602	43	77	8	11

### 4.2.2 사용 Class별 Coverage

#### default

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Files		92%		n/a	2	19	3	33	2	19	0	1
Main		0%		n/a	2	2	3	3	2	2	1	1
Controller.new ActionListener()		0%		n/a	2	2	3	3	2	2	1	1
Controller.new ActionListener()		0%		n/a	2	2	7	7	2	2	1	1
Controller.new MouseAdapter()		0%		0%	4	4	5	5	2	2	1	1
Controller.new KeyAdapter()		0%		0%	4	4	6	6	2	2	1	1
Controller.RoundedBorder		0%		n/a	5	5	13	13	5	5	1	1
Controller.new MouseAdapter()	■	0%	■	0%	9	9	23	23	4	4	1	1
Analyze		72%		75%	24	73	36	160	1	12	0	1
Calculate		63%		66%	24	39	28	94	8	14	0	1
Controller		0%		0%	41	41	260	260	13	13	1	1
Total	2,665 of 4,269	38%	121 of 246	51%	119	200	382	602	43	77	8	11

### 4.2.3 결과

JUnit Code가 Analyze 코도와 Calculate Class에 관한 것만 작성이 되어서 해당 클래스 커버리지 계산만 의미가 있다. 또한 다른 Class들의 JUnit Test가 되지 않은 상황이기 때문에 JUnit 테스트 결과에 대해서 신뢰하기 어렵다 판단된다.