Static Analysis Report for Team 3

Project Team

Team 3

Latest update on:

2016-06-05

Team Information

컴퓨터공학부 201011320 김용현

컴퓨터공학부 201111360 손준익

컴퓨터공학부 201111384 정국빈

Table of Contents

1 Sonar Qube

1.1 URL

- http://54.199.228.51:9000

- 자세한 결과는 위 주소에서 확인 가능

1.2 Projects Name

- Check Style : checkstyle_google

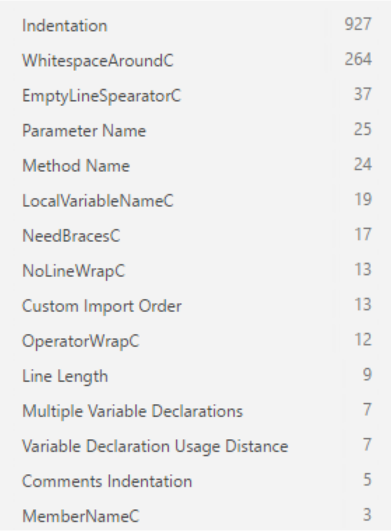
- Find Bug : bugfind

- PMD : pmd

2 Check Style

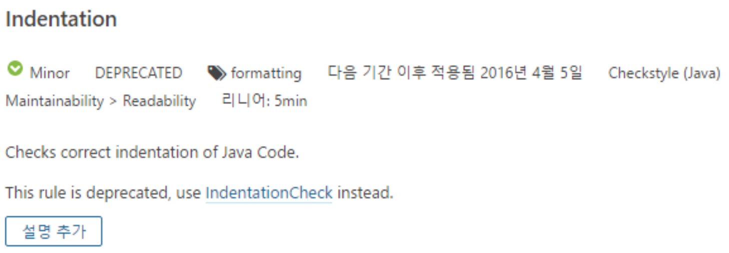
2.1 Rules

2.1.1 List



Indentation	927
WhitespaceAroundC	264
EmptyLineSpearatorC	37
Parameter Name	25
Method Name	24
LocalVariableNameC	19
NeedBracesC	17
NoLineWrapC	13
Custom Import Order	13
OperatorWrapC	12
Line Length	9
Multiple Variable Declarations	7
Variable Declaration Usage Distance	7
Comments Indentation	5
MemberNameC	3

2.1.2 Description



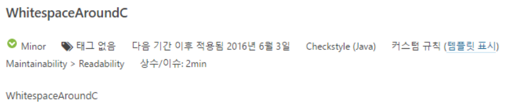
Indentation

Minor DEPRECATED formatting 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Readability 리니어: 5min

Checks correct indentation of Java Code.

This rule is deprecated, use IndentationCheck instead.

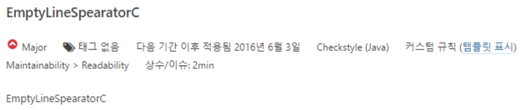
설명 추가



WhitespaceAroundC

Minor 태그 없음 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Maintainability > Readability 상수/이슈: 2min

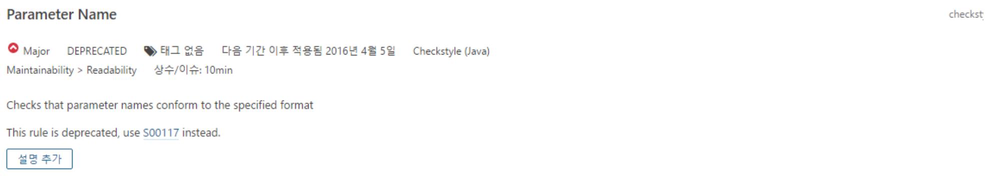
WhitespaceAroundC



EmptyLineSpearatorC

Major 태그 없음 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Maintainability > Readability 상수/이슈: 2min

EmptyLineSpearatorC



Parameter Name

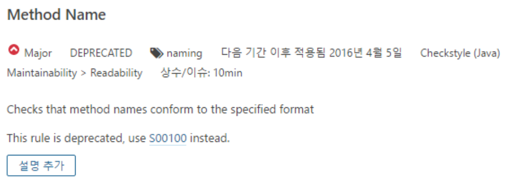
checkst

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Readability 상수/이슈: 10min

Checks that parameter names conform to the specified format

This rule is deprecated, use S00117 instead.

설명 추가



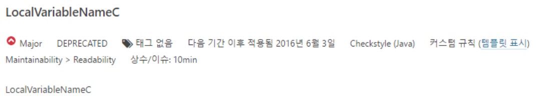
Method Name

Major DEPRECATED naming 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Readability 상수/이슈: 10min

Checks that method names conform to the specified format

This rule is deprecated, use S00100 instead.

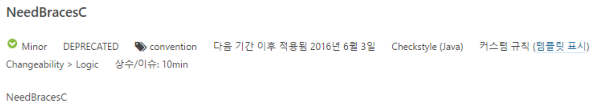
설명 추가



LocalVariableNameC

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Maintainability > Readability 상수/이슈: 10min

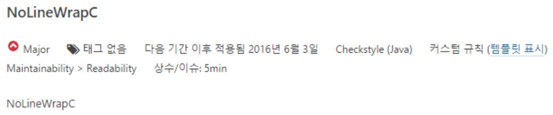
LocalVariableNameC



NeedBracesC

Minor DEPRECATED convention 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Changeability > Logic 상수/이슈: 10min

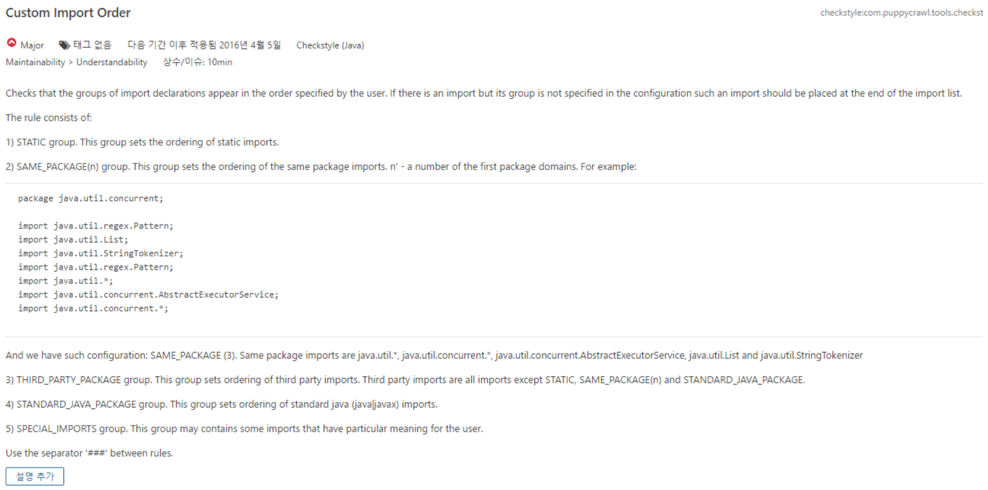
NeedBracesC



NoLineWrapC

Major 태그 없음 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Maintainability > Readability 상수/이슈: 5min

NoLineWrapC



Custom Import Order

checkstyle:com.puppycrawl.tools.checkst

Major 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Understandability 상수/이슈: 10min

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.

The rule consists of:

1) STATIC group. This group sets the ordering of static imports.

2) SAME_PACKAGE(n) group. This group sets the ordering of the same package imports. n' - a number of the first package domains. For example:

```
package java.util.concurrent;

import java.util.regex.Pattern;
import java.util.List;
import java.util.StringTokenizer;
import java.util.regex.Pattern;
import java.util.*;
import java.util.concurrent.AbstractExecutorService;
import java.util.concurrent.*;
```

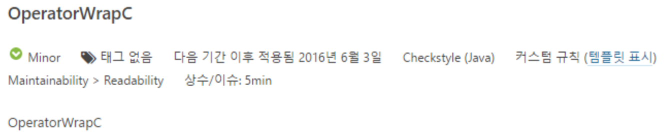
And we have such configuration: SAME_PACKAGE (3). Same package imports are java.util.*, java.util.concurrent.*, java.util.concurrent.AbstractExecutorService, java.util.List and java.util.StringTokenizer

3) THIRD_PARTY_PACKAGE group. This group sets ordering of third party imports. Third party imports are all imports except STATIC, SAME_PACKAGE(n) and STANDARD_JAVA_PACKAGE.

4) STANDARD_JAVA_PACKAGE group. This group sets ordering of standard java (java|javax) imports.

5) SPECIAL_IMPORTS group. This group may contains some imports that have particular meaning for the user.

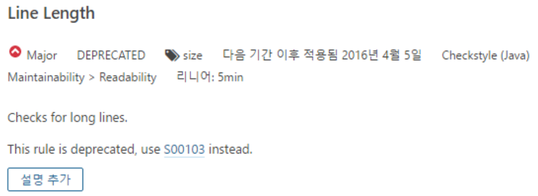
Use the separator '###' between rules.



OperatorWrapC

Minor 태그 없음 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Maintainability > Readability 상수/이슈: 5min

OperatorWrapC

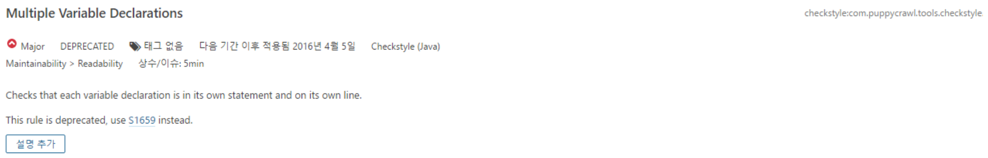


Line Length

Major DEPRECATED size 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Readability 리니어: 5min

Checks for long lines.

This rule is deprecated, use S00103 instead.



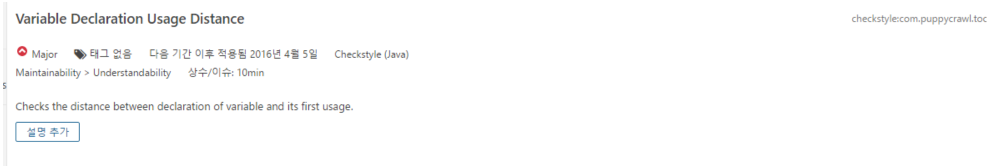
Multiple Variable Declarations

checkstyle:com.puppycrawl.tools.checkstyle.

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Readability 상수/이슈: 5min

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

This rule is deprecated, use S1659 instead.

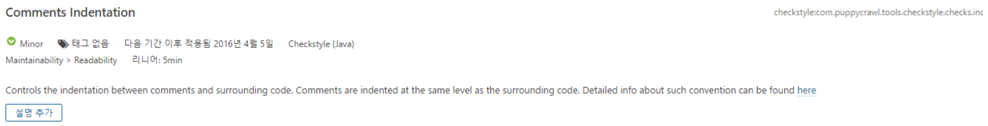


Variable Declaration Usage Distance

checkstyle:com.puppycrawl.toc

Major 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Understandability 상수/이슈: 10min

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

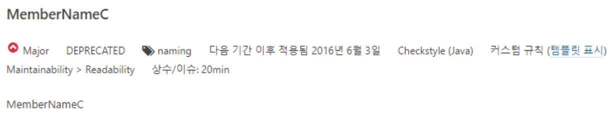


Comments Indentation

checkstyle:com.puppycrawl.tools.checkstyle.checks.inc

Minor 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 Checkstyle (Java)
Maintainability > Readability 리니어: 5min

Controls the indentation between comments and surrounding code. Comments are indented at the same level as the surrounding code. Detailed info about such convention can be found here

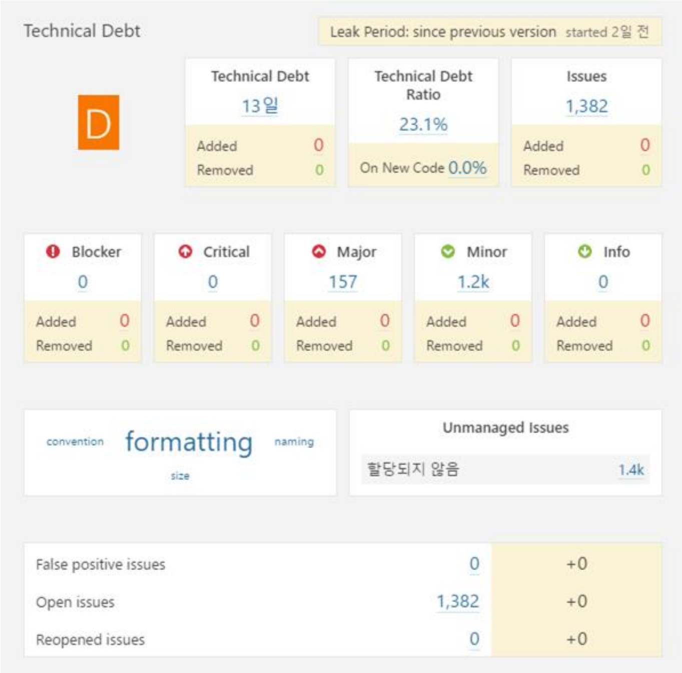


MemberNameC

Major DEPRECATED naming 다음 기간 이후 적용됨 2016년 6월 3일 Checkstyle (Java) 커스텀 규칙 (템플릿 표시)
Maintainability > Readability 상수/이슈: 20min

MemberNameC

2.2 Analysis Result



Technical Debt

Leak Period: since previous version started 2일 전

D

Technical Debt	Technical Debt Ratio	Issues
13일	23.1%	1,382
Added 0 Removed 0	On New Code 0.0%	Added 0 Removed 0

Blocker	Critical	Major	Minor	Info
0	0	157	1.2k	0
Added 0 Removed 0	Added 0 Removed 0	Added 0 Removed 0	Added 0 Removed 0	Added 0 Removed 0

Unmanaged Issues

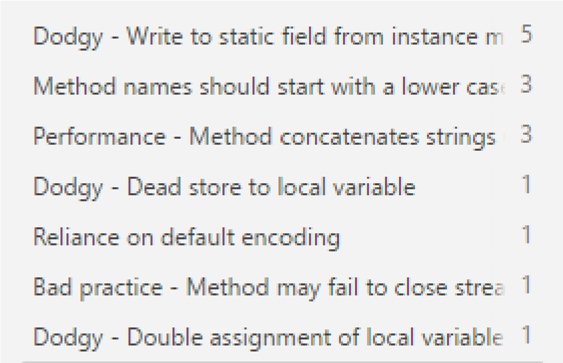
할당되지 않음 1.4k

False positive issues	0	+0
Open issues	1,382	+0
Reopened issues	0	+0

3 Find Bug

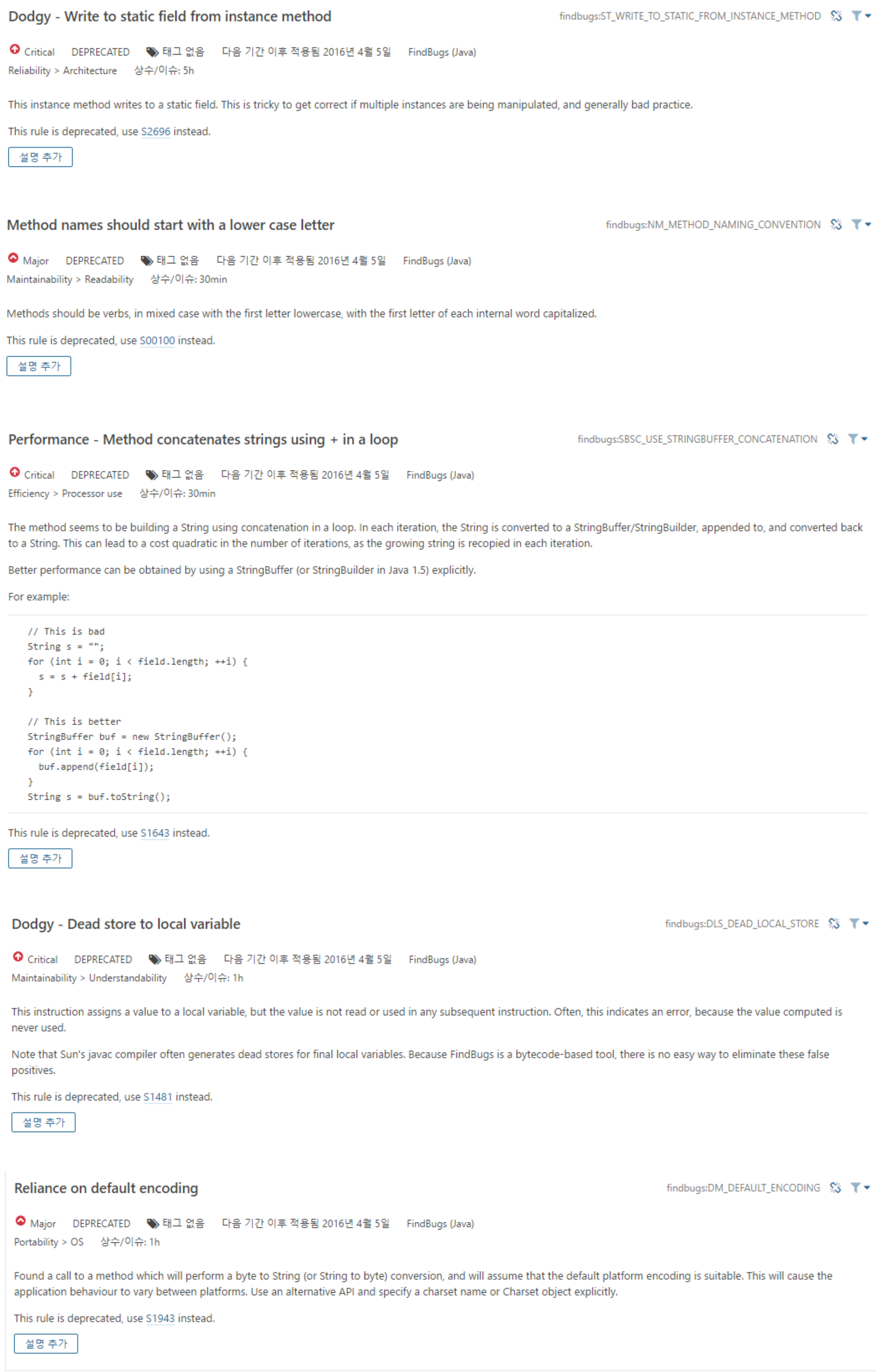
3.1 Rules

3.1.1 List



Dodgy - Write to static field from instance m	5
Method names should start with a lower cas	3
Performance - Method concatenates strings	3
Dodgy - Dead store to local variable	1
Reliance on default encoding	1
Bad practice - Method may fail to close strea	1
Dodgy - Double assignment of local variable	1

3.1.2 Description



Dodgy - Write to static field from instance method

findbugs:ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD

Critical DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Reliability > Architecture 상수/이슈: 5h

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

This rule is deprecated, use S2696 instead.

설명 추가

Method names should start with a lower case letter

findbugs:NM_METHOD_NAMING_CONVENTION

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Maintainability > Readability 상수/이슈: 30min

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

This rule is deprecated, use S00100 instead.

설명 추가

Performance - Method concatenates strings using + in a loop

findbugs:SBSC_USE_STRINGBUFFER_CONCATENATION

Critical DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Efficiency > Processor use 상수/이슈: 30min

The method seems to be building a String using concatenation 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 the number of iterations, as the growing string is recopied in each iteration.

Better performance can be obtained by using a StringBuffer (or StringBuilder in Java 1.5) explicitly.

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();
```

This rule is deprecated, use S1643 instead.

설명 추가

Dodgy - Dead store to local variable

findbugs:DLS_DEAD_LOCAL_STORE

Critical DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Maintainability > Understandability 상수/이슈: 1h

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.

This rule is deprecated, use S1481 instead.

설명 추가

Reliance on default encoding

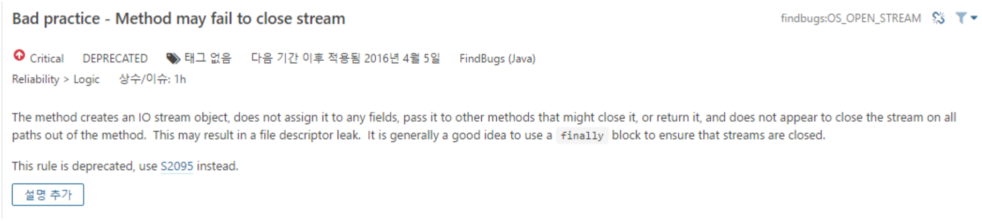
findbugs:DM_DEFAULT_ENCODING

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Portability > OS 상수/이슈: 1h

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.

This rule is deprecated, use S1943 instead.

설명 추가

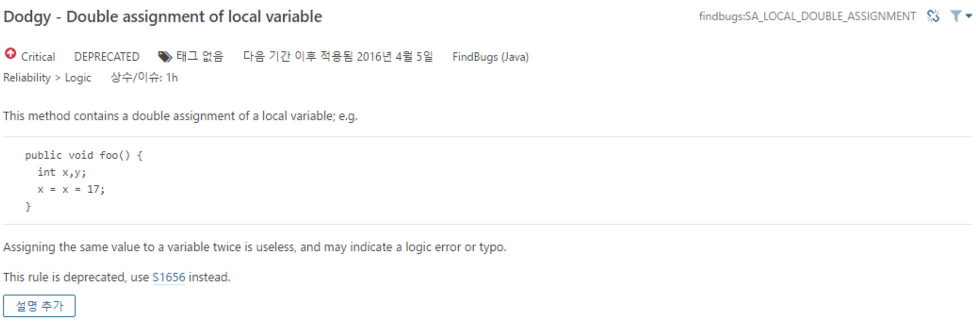


Bad practice - Method may fail to close stream findbugs:OS_OPEN_STREAM

Critical DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Reliability > Logic 상수/이슈: 1h

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.

This rule is deprecated, use S2095 instead.



Dodgy - Double assignment of local variable findbugs:SA_LOCAL_DOUBLE_ASSIGNMENT

Critical DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 FindBugs (Java)
Reliability > Logic 상수/이슈: 1h

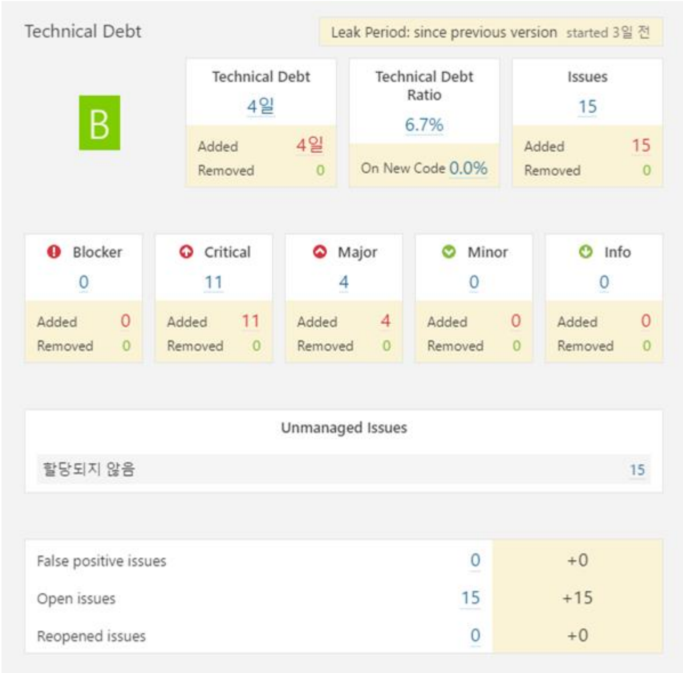
This method contains a double assignment of a local variable; e.g.

```
public void foo() {
  int x,y;
  x = x = 17;
}
```

Assigning the same value to a variable twice is useless, and may indicate a logic error or typo.

This rule is deprecated, use S1656 instead.

3.2 Analysis Result



Technical Debt

Leak Period: since previous version started 3일 전

Technical Debt	Technical Debt Ratio	Issues
4일	6.7%	15
Added 4일	On New Code 0.0%	Added 15
Removed 0		Removed 0

Blocker	Critical	Major	Minor	Info
0	11	4	0	0
Added 0	Added 11	Added 4	Added 0	Added 0
Removed 0	Removed 0	Removed 0	Removed 0	Removed 0

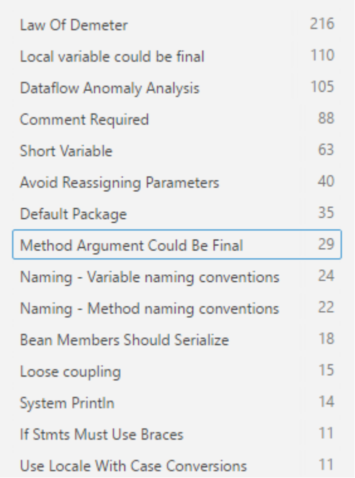
Unmanaged Issues	
할당되지 않음	15

False positive issues	0	+0
Open issues	15	+15
Reopened issues	0	+0

4 PMD

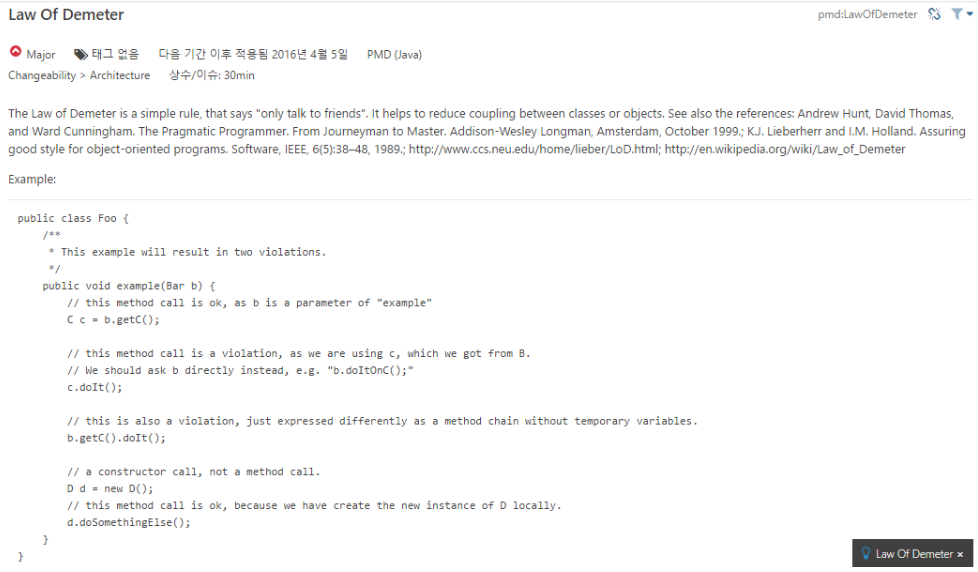
4.1 Rules

4.1.1 List



Law Of Demeter	216
Local variable could be final	110
Dataflow Anomaly Analysis	105
Comment Required	88
Short Variable	63
Avoid Reassigning Parameters	40
Default Package	35
Method Argument Could Be Final	29
Naming - Variable naming conventions	24
Naming - Method naming conventions	22
Bean Members Should Serialize	18
Loose coupling	15
System Println	14
If Stmts Must Use Braces	11
Use Locale With Case Conversions	11

4.1.2 Description



Law Of Demeter

pmd:LawOfDemeter

Major 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Changeability > Architecture 상수/이슈: 30min

The Law of Demeter is a simple rule, that says "only talk to friends". It helps to reduce coupling between classes or objects. See also the references: Andrew Hunt, David Thomas, and Ward Cunningham. The Pragmatic Programmer. From Journeyman to Master. Addison-Wesley Longman, Amsterdam, October 1999.; K.J. Lieberherr and I.M. Holland. Assuring good style for object-oriented programs. Software, IEEE, 6(5):38–48, 1989.; http://www.ccs.neu.edu/home/lieber/LoD.html; http://en.wikipedia.org/wiki/Law_of_Demeter

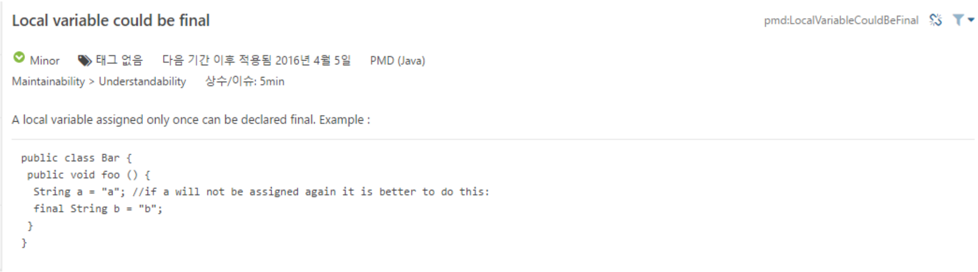
Example:

```
public class Foo {
    /**
     * This example will result in two violations.
     */
    public void example(Bar b) {
        // this method call is ok, as b is a parameter of "example"
        C c = b.getC();

        // this method call is a violation, as we are using c, which we got from B.
        // We should ask b directly instead, e.g. "b.doItOnC();"
        c.doIt();

        // this is also a violation, just expressed differently as a method chain without temporary variables.
        b.getC().doIt();

        // a constructor call, not a method call.
        D d = new D();
        // this method call is ok, because we have create the new instance of D locally.
        d.doSomethingElse();
    }
}
```



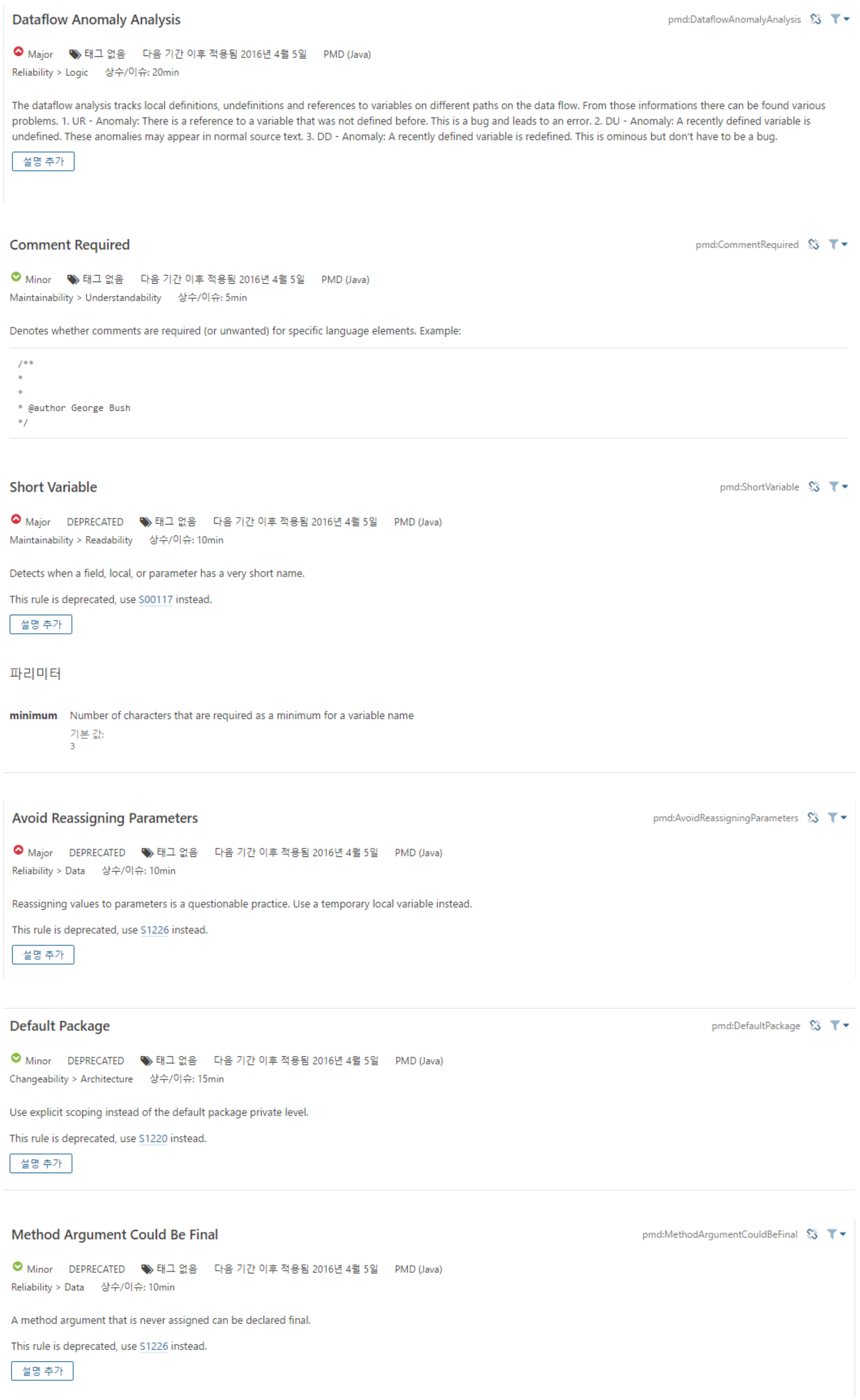
Local variable could be final

pmd:LocalVariableCouldBeFinal

Minor 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Maintainability > Understandability 상수/이슈: 5min

A local variable assigned only once can be declared final. Example :

```
public class Bar {
 public void foo () {
  String a = "a"; //if a will not be assigned again it is better to do this:
  final String b = "b";
 }
}
```



Dataflow Anomaly Analysis

pmd:DataflowAnomalyAnalysis

Major 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Reliability > Logic 상수/이슈: 20min

The dataflow analysis tracks local definitions, undefinitions and references to variables on different paths on the data flow. From those informations there can be found various problems. 1. UR - Anomaly: There is a reference to a variable that was not defined before. This is a bug and leads to an error. 2. DU - Anomaly: A recently defined variable is undefined. These anomalies may appear in normal source text. 3. DD - Anomaly: A recently defined variable is redefined. This is ominous but don't have to be a bug.

Comment Required

pmd:CommentRequired

Minor 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Maintainability > Understandability 상수/이슈: 5min

Denotes whether comments are required (or unwanted) for specific language elements. Example:

```
/**
*
*
* @author George Bush
*/
```

Short Variable

pmd:ShortVariable

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Maintainability > Readability 상수/이슈: 10min

Detects when a field, local, or parameter has a very short name.

This rule is deprecated, use S00117 instead.

파라미터

minimum Number of characters that are required as a minimum for a variable name
기본 값:
3

Avoid Reassigning Parameters

pmd:AvoidReassigningParameters

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Reliability > Data 상수/이슈: 10min

Reassigning values to parameters is a questionable practice. Use a temporary local variable instead.

This rule is deprecated, use S1226 instead.

Default Package

pmd:DefaultPackage

Minor DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Changeability > Architecture 상수/이슈: 15min

Use explicit scoping instead of the default package private level.

This rule is deprecated, use S1220 instead.

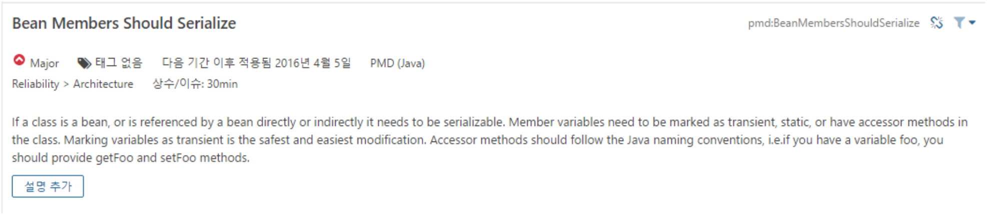
Method Argument Could Be Final

pmd:MethodArgumentCouldBeFinal

Minor DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Reliability > Data 상수/이슈: 10min

A method argument that is never assigned can be declared final.

This rule is deprecated, use S1226 instead.

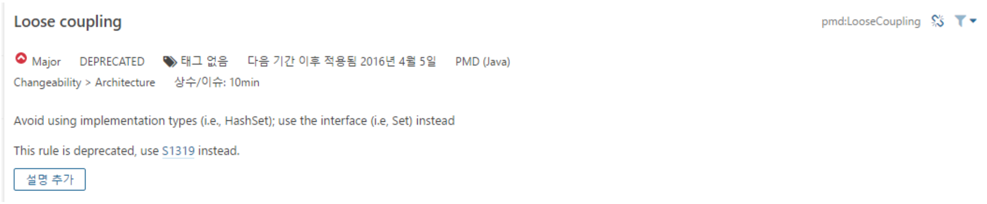


Bean Members Should Serialize

pmd:BeanMembersShouldSerialize

Major 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Reliability > Architecture 상수/이슈: 30min

If a class is a bean, or is referenced by a bean directly or indirectly it needs to be serializable. Member variables need to be marked as transient, static, or have accessor methods in the class. Marking variables as transient is the safest and easiest modification. Accessor methods should follow the Java naming conventions, i.e.if you have a variable foo, you should provide getFoo and setFoo methods.



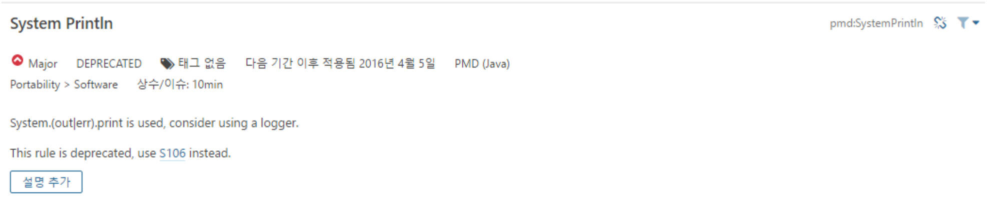
Loose coupling

pmd:LooseCoupling

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Changeability > Architecture 상수/이슈: 10min

Avoid using implementation types (i.e., HashSet); use the interface (i.e, Set) instead

This rule is deprecated, use S1319 instead.



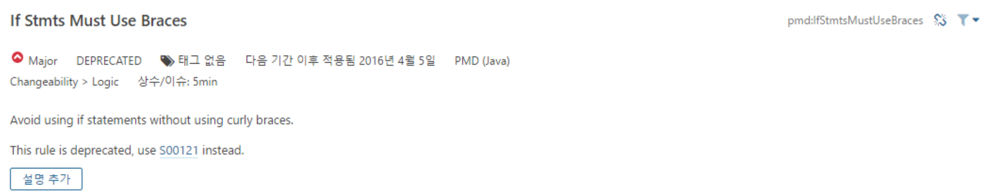
System Println

pmd:SystemPrintln

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Portability > Software 상수/이슈: 10min

System.(out|err).print is used, consider using a logger.

This rule is deprecated, use S106 instead.



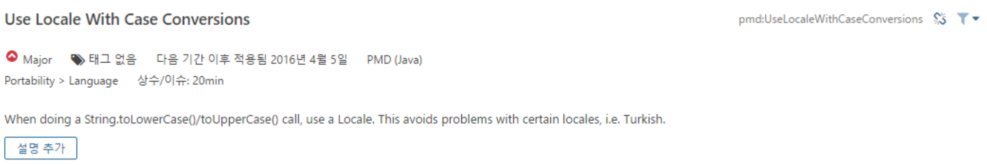
If Stmts Must Use Braces

pmd:IfStmtsMustUseBraces

Major DEPRECATED 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Changeability > Logic 상수/이슈: 5min

Avoid using if statements without using curly braces.

This rule is deprecated, use S00121 instead.



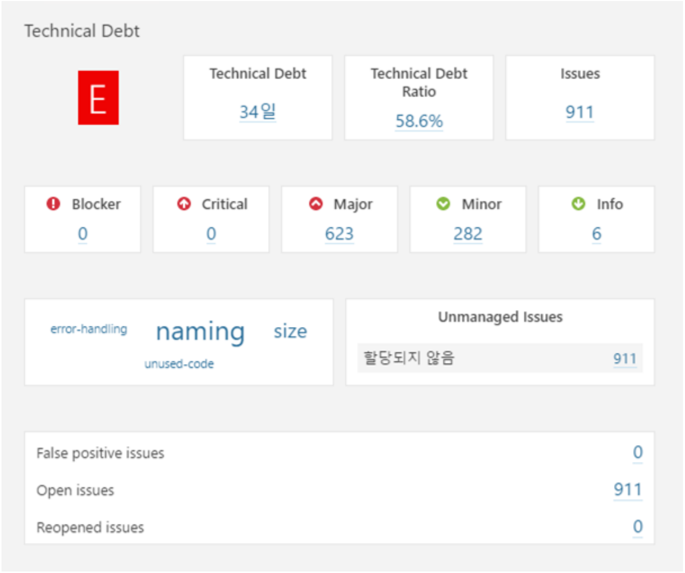
Use Locale With Case Conversions

pmd:UseLocaleWithCaseConversions

Major 태그 없음 다음 기간 이후 적용됨 2016년 4월 5일 PMD (Java)
Portability > Language 상수/이슈: 20min

When doing a String.toLowerCase()/toUpperCase() call, use a Locale. This avoids problems with certain locales, i.e. Turkish.

4.2 Analysis Result



Technical Debt

E

Technical Debt	Technical Debt Ratio	Issues
34일	58.6%	911

Blocker	Critical	Major	Minor	Info
0	0	623	282	6

Unmanaged Issues

할당되지 않음 911

False positive issues	0
Open issues	911
Reopened issues	0