




 Secrets


 ABAP


 Apex


 C


 C++


 CloudFormation


 COBOL


 C#


 CSS


 Flex


 Go


 HTML


 **Java**


 JavaScript


 Kotlin


 Objective C


 PHP


 PL/I


 PL/SQL


 Python


 RPG


 Ruby


 Scala


 Swift


 Terraform


 Text


 TypeScript

 T-SQL

 VB.NET

 VB6

 XML



Java static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

All rules632

Vulnerability53

Bug154

Security Hotspot36

Code Smell389

Quick Fix42

Tags ▾

Search by name... 🔍

Silly String operations should not be made

Bug

Non-serializable classes should not be written

Bug

"hashCode" and "toString" should not be called on array instances

Bug

Collections should not be passed as arguments to their own methods

Bug

"BigDecimal(double)" should not be used

Bug

Invalid "Date" values should not be used

Bug

Reflection should not be used to check non-runtime annotations

Bug

Custom serialization method signatures should meet requirements

Bug

"Externalizable" classes should have no-arguments constructors

Bug

Classes should not be compared by name

Bug

Related "if/else if" statements should not have the same condition

Bug

IllegalMonitorStateException should not be caught

Analyze your code

Code SmellCritical?error-handling multi-threading

According to Oracle Javadoc:

IllegalMonitorStateException is thrown when a thread has attempted to wait on an object's monitor or to notify other threads waiting on an object's monitor without owning the specified monitor.

In other words, this exception can be thrown only in case of bad design because `Object.wait(...)`, `Object.notify()` and `Object.notifyAll()` methods should never be called on an object whose monitor is not held.

Noncompliant Code Example

```
public void doSomething(){
    ...
    try {
        ...
        anObject.notify();
        ...
    } catch(IllegalMonitorStateException e) {
        ...
    }
}
```

Compliant Solution

```
public void doSomething(){
    ...
    synchronized(anObject) {
        ...
        anObject.notify();
        ...
    }
}
```

Available In:





sonarlint | sonarcloud | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

[Privacy Policy](#)

https://rules.sonarsource.com/java/RSPEC-2235

1/2

| |
|---|
| <div>Synchronization should not be done on instances of value-based classes</div> <div> Bug</div> |
| <div>"Iterator.hasNext()" should not call "Iterator.next()"</div> <div> Bug</div> |
| <div>Identical expressions should not be used on both sides of a binary operator</div> <div> Bug</div> |
| <div>Loops with at most one iteration should be refactored</div> <div> Bug</div> |