




 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 rules 632

Vulnerability 53

Bug 154

Security Hotspot 36


Code Smell 389

Quick Fix 42


Tags ▾

Search by name... 🔍


"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


Synchronization should not be done on instances of value-based classes

 Bug


"Iterator.hasNext()" should not call "Iterator.next()"

 Bug


Identical expressions should not be used on both sides of a binary operator

 Bug


Loops with at most one iteration should be refactored

 Bug


Variables should not be self-assigned

 Bug


"StringBuilder" and "StringBuffer" should not be instantiated with a character

 Bug


Methods should not be named "toString", "hashCode" or "equal"

 Bug

"Thread.run()" should not be called directly

 Bug

"equals" method overrides should accept "Object" parameters

 Bug

"readResolve" methods should be inheritable

Analyze your code

Code Smell

Critical

pitfall

The `readResolve()` method allows final tweaks to the state of an object during deserialization. Non-final classes which implement `readResolve()`, should not set its visibility to `private` since it will then be unavailable to child classes.

Noncompliant Code Example

```
public class Fruit implements Serializable {
    private static final long serialVersionUID = 1;

    private Object readResolve() throws ObjectStreamException
    {...}

    //...
}

public class Raspberry extends Fruit implements Serializable
//...
}
```

Compliant Solution

```
public class Fruit implements Serializable {
    private static final long serialVersionUID = 1;

    protected Object readResolve() throws ObjectStreamException
    {...}

    //...
}





public class Raspberry extends Fruit implements Serializable
//...
}
```

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-2062

1/2

<div>accept object parameters</div> <div> Bug</div>
<div>The Object.finalize() method should not be called</div> <div> Bug</div>
<div>Enabling file access for WebViews is security-sensitive</div> <div> Security Hotspot</div>
<div>Enabling JavaScript support for WebViews is security-sensitive</div> <div> Security Hotspot</div>
<div>Constructing arguments of system commands from user input is</div>