




 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... 🔍


 Bug


Unsupported methods should not be called on some collection implementations




 Bug


Cast operations should not trigger a ClassCastException




 Bug


Members ignored during record serialization should not be used




 Bug


Map "computeIfAbsent()" and "computeIfPresent()" should not be used to add "null" values.




 Bug


Regex lookahead assertions should not be contradictory




 Bug


Back references in regular expressions should only refer to capturing groups that are matched before the reference



 Bug

Regex boundaries should not be used in a way that can never be matched



 Bug

Regex patterns following a possessive quantifier should not always fail

 Bug


Regular expressions should be syntactically valid


 Bug


Assertions comparing incompatible types should not be made


Child class fields should not shadow parent class fields

Analyze your code

 Code Smell

 Blocker

 ?

 confusing

Having a variable with the same name in two unrelated classes is fine, but do the same thing within a class hierarchy and you'll get confusion at best, chaos at worst.

Noncompliant Code Example

```
public class Fruit {
    protected Season ripe;
    protected Color flesh;

    // ...
}

public class Raspberry extends Fruit {
    private boolean ripe; // Noncompliant
    private static Color FLESH; // Noncompliant
}
```

Compliant Solution

```
public class Fruit {
    protected Season ripe;
    protected Color flesh;

    // ...
}

public class Raspberry extends Fruit {
    private boolean ripened;
    private static Color FLESH_COLOR;
}
```

Exceptions

This rule ignores same-name fields that are static in both the parent and child classes. This rule ignores private parent class fields, but in all other such cases, the child class field should be renamed.


```
public class Fruit {
    private Season ripe;
    // ...
}

public class Raspberry extends Fruit {
    private Season ripe; // Compliant as parent field 'ripe'
    // ...
}
```


https://rules.sonarsource.com/java/RSPEC-2387

1/2


JUnit5 inner test classes should be annotated with @Nested

 Bug


Only one method invocation is expected when testing checked exceptions

 Bug




Assertion methods should not be used within the try block of a try-catch catching an Error

 Bug

Getters and setters should access the expected fields



Available In:

 |  | 

© 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](#)