Bug (154)

Security

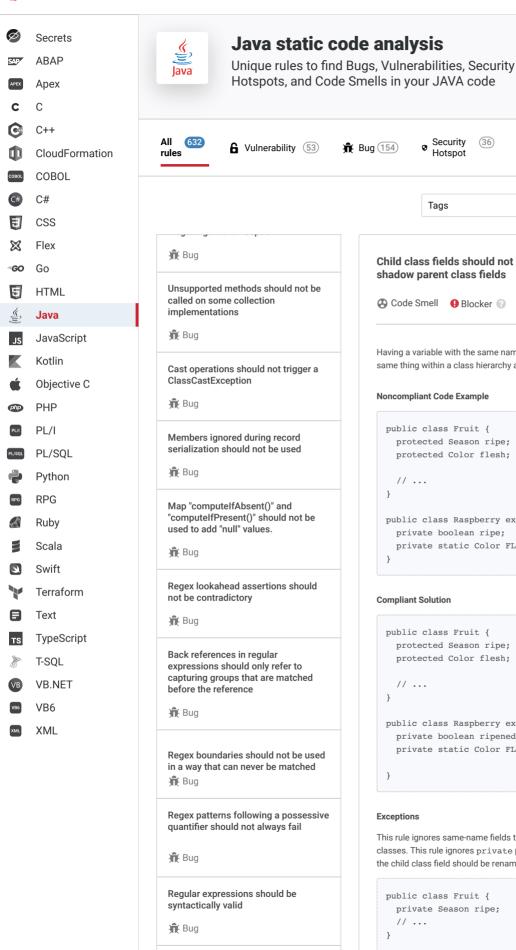
Hotspot

(36)



Products >

O Quick 42



```
Search by name.
           Tags
Child class fields should not
                                             Analyze your code
shadow parent class fields
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;
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'

⊗ Code

Smell

(389)

Assertions comparing incompatible

types should not be made

₩ Bud

JUnit5 inner test classes should be annotated with @Nested

IF Bug

Only one method invocation is expected when testing checked exceptions

IF Bug

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

IF Bug

Getters and setters should access the

expected fields

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