




 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... 🔍


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

 Bug

JUnit5 inner test classes should be annotated with @Nested


 Bug


Only one method invocation is expected when testing checked exceptions


 Bug


TestCases should contain tests

Analyze your code

 Code Smell

 Blocker



 junit tests unused confusing




There's no point in having a JUnit `TestCase` without any test methods. Similarly, you shouldn't have a file in the tests directory named `*Test`, `*Tests`, or `*TestCase`, but no tests in the file. Doing either of these things may lead someone to think that uncovered classes have been tested.

This rule raises an issue when files in the test directory are named `*Test`, `*Tests`, or `*TestCase` or implement `TestCase` but don't contain any tests.

Supported frameworks:

- JUnit3
- JUnit4
- JUnit5
- TestNG
- Zohhak
- ArchUnit
- Pact

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](#)

https://rules.sonarsource.com/java/RSPEC-2187

1/2


<p>Assertion methods should not be used within the try block of a try-catch catching an Error</p> <p> Bug</p>
<p>Getters and setters should access the expected fields</p> <p> Bug</p>
<p>Zero should not be a possible denominator</p> <p> Bug</p>
<p>Locks should be released</p> <p> Bug</p>