




 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... 🔍

should not be disclosed

Vulnerability

Reflection should not be vulnerable to injection attacks

Vulnerability

Authorizations should be based on strong decisions

Vulnerability

OpenSAML2 should be configured to prevent authentication bypass

Vulnerability

Server-side requests should not be vulnerable to forging attacks

Vulnerability

Collections should not be modified while they are iterated

Bug

Equals method should be overridden in records containing array fields

Bug

Reflection should not be used to increase accessibility of records' fields

Bug

AssertJ assertions with "Consumer" arguments should contain assertion inside consumers

Bug

The regex escape sequence \cX should only be used with characters in the @-\_ range

Bug

Regular expressions should not overflow the stack

Bug

Tests method should not be

"ScheduledThreadPoolExecutor" should not have 0 core threads

Analyze your code

BugCritical?

java.util.concurrent.ScheduledThreadPoolExecutor's pool is sized with corePoolSize, so setting corePoolSize to zero means the executor will have no threads and run nothing.

This rule detects instances where corePoolSize is set to zero, via either its setter or the object constructor.

Noncompliant Code Example

```
public void do(){  
  
    ScheduledThreadPoolExecutor stpe1 = new ScheduledThreadPoo  
  
    ScheduledThreadPoolExecutor stpe2 = new ScheduledThreadPoo  
    stpe2.setCorePoolSize(0); // Noncompliant  
  
    ...  
}
```





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-2122

1/2

<div>Tests method should not be annotated with competing annotations</div> <div> Bug</div>
<div>Assertions should not be used in production code</div> <div> Bug</div>
<div>DateTimeFormatters should not use mismatched year and week numbers</div> <div> Bug</div>
<div>Unicode Grapheme Clusters should be avoided inside regex character classes</div> <div> Bug</div>