sonar RULES

Products

- 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...

- "NullPointerException" should not be explicitly thrown — Code Smell
- Classes should not have too many methods — Code Smell
- Methods should not have too many lines — Code Smell
- Track uses of "NOSONAR" comments — Code Smell
- Classes and enums with private members should have a constructor — Code Smell
- Track comments matching a regular expression — Code Smell
- Statements should be on separate lines — Code Smell
- Classes should not be coupled to too many other classes (Single Responsibility Principle) — Code Smell
- "java.lang.Error" should not be extended — Code Smell
- Anonymous classes should not have too many lines — Code Smell
- Public types, methods and fields (API) should be documented with Javadoc — Code Smell
- Exception handlers should preserve

Inner class calls to super class methods should be unambiguous

Analyze your code

Code Smell | Major | pitfall

When an inner class extends another class, and both its outer class and its parent class have a method with the same name, calls to that method can be confusing. The compiler will resolve the call to the superclass method, but maintainers may be confused, so the superclass method should be called explicitly, using `super.`.

Noncompliant Code Example

```
public class Parent {
  public void foo() { ... }
}

public class Outer {

  public void foo() { ... }

  public class Inner extends Parent {

    public void doTheThing() {
      foo();  // Noncompliant; was Outer.this.foo() intended
      // ...
    }
  }
}
```

Compliant Solution

```
public class Parent {
  public void foo() { ... }
}

public class Outer {

  public void foo() { ... }

  public class Inner extends Parent {

    public void doTheThing() {
      super.foo();
      // ...
    }
  }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

the original exceptions

Code Smell

Checked exceptions should not be thrown

Code Smell

Public methods should throw at most one checked exception

Code Smell

"switch case" clauses should not have too many lines of code

Code Smell

Methods should not have too many

© 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