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...

Rules list (sidebar)

Abstract class names should comply with a naming convention
⊙ Code Smell

Strings literals should be placed on the left side when checking for equality
⊙ Code Smell

Files should contain an empty newline at the end
⊙ Code Smell

Source code should be indented consistently
⊙ Code Smell

A close curly brace should be located at the beginning of a line
⊙ Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines
⊙ Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line
⊙ Code Smell

An open curly brace should be located at the beginning of a line
⊙ Code Smell

An open curly brace should be located at the end of a line
⊙ Code Smell

Tabulation characters should not be used
⊙ Code Smell

Functions should not be defined with a variable number of arguments
⊙ Code Smell

Arrays should not be copied using loops

Analyze your code

⊙ Code Smell   ⊙ Minor ?   🏷 clumsy

Using a loop to copy an array or a subset of an array is simply wasted code when there are built-in functions to do it for you. Instead, use `Arrays.copyOf` to copy an entire array into another array, use `System.arraycopy` to copy only a subset of an array into another array, and use `Arrays.asList` to feed the constructor of a new list with an array.

Note that `Arrays.asList` simply puts a `Collections` wrapper around the original array, so further steps are required if a non-fixed-size `List` is desired.

Noncompliant Code Example

```
public void makeCopies(String[] source) {

  this.array = new String[source.length];
  this.list = new ArrayList(source.length);

  for (int i = 0; i < source.length; i++) {
    this.array[i] = source[i]; // Noncompliant
  }

  for (String s : source) {
    this.list.add(s); // Noncompliant
  }
}
```

Compliant Solution

```
public void makeCopies(String[] source) {
  this.array = Arrays.copyOf(source, source.length);
  Collections.addAll(this.list, source);
}
```

Exceptions

Rule detects only the most idiomatic patterns, it will not consider loops with non-trivial control flow. For example, array elements that are copied conditionally are ignored.

```
public int[] getCopy(int[] source) {
  int[] dest = new int[source.length];
  for (int i = 0; i < source.length; i++) {
    if (source[i] > 10) {
      dest[i] = source[i];  // Compliant
    }
  }
  return dest;
}
```

Local-Variable Type Inference should be used

⊗ Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

⊗ Code Smell

Track uses of disallowed classes

⊗ Code Smell

Track uses of "@SuppressWarnings" annotations

⊗ Code Smell

Available In:

sonarlint ⊖ | sonarcloud ♻ | sonarqube ⟩⟩⟩