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...

Nested code blocks should not be used

Analyze your code

Code Smell | Minor | bad-practice

Nested code blocks can be used to create a new scope and restrict the visibility of the variables defined inside it. Using this feature in a method typically indicates that the method has too many responsibilities, and should be refactored into smaller methods.

Noncompliant Code Example

```
public void evaluate(int operator) {
  // Do some computation...
  {
    int a = stack.pop();
    int b = stack.pop();
    int result = a + b;
    stack.push(result);
  }
}
```

Compliant Solution

```
public void evaluate(int operator) {
  // Do some computation...
  evaluateAdd();
}

private void evaluateAdd() {
  int a = stack.pop();
  int b = stack.pop();
  int result = a + b;
  stack.push(result);
}
```

Available In:

sonarlint | sonarcloud | sonarqube

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