Java static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

Methods should not have identical implementations

Analyze your code

Code Smell | Major | confusing duplicate suspicious

When two methods have the same implementation, either it was a mistake - something else was intended - or the duplication was intentional, but may be confusing to maintainers. In the latter case, one implementation should invoke the other. Numerical and string literals are not taken into account.

Noncompliant Code Example

```
private final static String CODE = "bounteous";

public String calculateCode() {
  doTheThing();
  return CODE;
}

public String getName() {  // Noncompliant
  doTheThing();
  return CODE;
}
```

Compliant Solution

```
private final static String CODE = "bounteous";

public String getCode() {
  doTheThing();
  return CODE;
}

public String getName() {
  return getCode();
}
```

Exceptions

Methods that are not accessors (getters and setters), with fewer than 2 statements are ignored.

Available In:
sonarlint sonarcloud sonarqube

on operator precedence

Code Smell

Custom getter method should not be used to override record's getter behavior

Code Smell

Tests should use fixed data instead of randomized data

Code Smell

Spring's ModelAndViewAssert assertions should be used instead of other assertions

Code Smell