Java static code analysis

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

Return of boolean expressions should not be wrapped into an "if-then-else" statement

Analyze your code

Code Smell | Minor | clumsy

Return of boolean literal statements wrapped into `if-then-else` ones should be simplified.

Similarly, method invocations wrapped into `if-then-else` differing only from boolean literals should be simplified into a single invocation.

Noncompliant Code Example

```
boolean foo(Object param) {
  if (expression) { // Noncompliant
    bar(param, true, "qix");
  } else {
    bar(param, false, "qix");
  }

  if (expression) {  // Noncompliant
    return true;
  } else {
    return false;
  }
}
```

Compliant Solution

```
boolean foo(Object param) {
  bar(param, expression, "qix");

  return expression;
}
```

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