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...

- Code Smell
- Tests should be kept in a dedicated source directory — Code Smell
- "this" should not be exposed from constructors — Code Smell
- Classes should not have too many "static" imports — Code Smell
- Escaped Unicode characters should not be used — Code Smell
- Inner classes should not have too many lines of code — Code Smell
- Inner classes which do not reference their owning classes should be "static" — Code Smell
- "deleteOnExit" should not be used — Code Smell
- Public methods should not contain selector arguments — Code Smell
- Java parser failure — Code Smell
- Track uses of disallowed methods — Code Smell
- Types should be used in lambdas — Code Smell
- "java.time" classes should be used for dates and times

"Thread.sleep" should not be used in tests

Analyze your code

Code Smell Major tests bad-practice

Using `Thread.sleep` in a test is just generally a bad idea. It creates brittle tests that can fail unpredictably depending on environment ("Passes on my machine!") or load. Don't rely on timing (use mocks) or use libraries such as `Awaitility` for asynchroneous testing.

Noncompliant Code Example

```
@Test
public void testDoTheThing(){

  MyClass myClass = new MyClass();
  myClass.doTheThing();

  Thread.sleep(500);  // Noncompliant
  // assertions...
}
```

Compliant Solution

```
@Test
public void testDoTheThing(){

  MyClass myClass = new MyClass();
  myClass.doTheThing();

  await().atMost(2, Duration.SECONDS).until(didTheThing());
  // assertions...
}

private Callable<Boolean> didTheThing() {
  return new Callable<Boolean>() {
    public Boolean call() throws Exception {
      // check the condition that must be fulfilled...
    }
  };
}
```

Available In:

sonarlint sonarcloud sonarqube

Code Smell

The names of methods with boolean return values should start with "is" or "has"

Code Smell

Files should contain only one top-level class or interface each

Code Smell

Classes should not have too many fields

Code Smell

The ternary operator should not be