Java static code analysis

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

Code Smell

"enum" fields should not be publicly mutable
Code Smell

Abstract methods should not be redundant
Code Smell

Arrays should not be copied using loops
Code Smell

Static non-final field names should comply with a naming convention
Code Smell

JUnit rules should be used
Code Smell

Nested "enum"s should not be declared static
Code Smell

"catch" clauses should do more than rethrow
Code Smell

Mutable fields should not be "public static"
Code Smell

The diamond operator ("<>") should be used
Code Smell

"finalize" should not set fields to "null"
Code Smell

Subclasses that add fields should override "equals"
Code Smell

"Thread.run()" should not be called directly

Analyze your code

Bug Major cwe multi-threading cert

The purpose of the `Thread.run()` method is to execute code in a separate, dedicated thread. Calling this method directly doesn't make sense because it causes its code to be executed in the current thread.

To get the expected behavior, call the `Thread.start()` method instead.

Noncompliant Code Example

```
Thread myThread = new Thread(runnable);
myThread.run(); // Noncompliant
```

Compliant Solution

```
Thread myThread = new Thread(runnable);
myThread.start(); // Compliant
```

See

- MITRE, CWE-572 - Call to Thread run() instead of start()
- CERT THI00-J. - Do not invoke Thread.run()

Available In:

sonarlint | sonarcloud | sonarqube

Catches should be combined

Code Smell

Methods of "Random" that return floating point values should not be used in random integer generation

Code Smell

Parsing should be used to convert "Strings" to primitives

Code Smell

Classes should not be empty

Code Smell