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...

Abstract class names should comply with a naming convention
Code Smell

Strings literals should be placed on the left side when checking for equality
Code Smell

Files should contain an empty newline at the end
Code Smell

Source code should be indented consistently
Code Smell

A close curly brace should be located at the beginning of a line
Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines
Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line
Code Smell

An open curly brace should be located at the beginning of a line
Code Smell

An open curly brace should be located at the end of a line
Code Smell

Tabulation characters should not be used
Code Smell

Functions should not be defined with a variable number of arguments
Code Smell

"@NonNull" values should not be set to null

Analyze your code

Bug | Minor | cwe cert

Fields, parameters and return values marked `@NotNull`, `@NonNull`, or `@Nonnull` are assumed to have non-null values and are not typically null-checked before use. Therefore setting one of these values to `null`, or failing to set such a class field in a constructor, could cause `NullPointerExceptions` at runtime.

Noncompliant Code Example

```
public class MainClass {

  @Nonnull
  private String primary;
  private String secondary;

  public MainClass(String color) {
    if (color != null) {
      secondary = null;
    }
    primary = color;  // Noncompliant; "primary" is Nonnull
  }

  public MainClass() { // Noncompliant; "primary" is Nonnull
  }

  @Nonnull
  public String indirectMix() {
    String mix = null;
    return mix;  // Noncompliant; return value is Nonnull, b
  }
```

See

- MITRE, CWE-476 - NULL Pointer Dereference
- CERT, EXP01-J. - Do not use a null in a case where an object is required

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