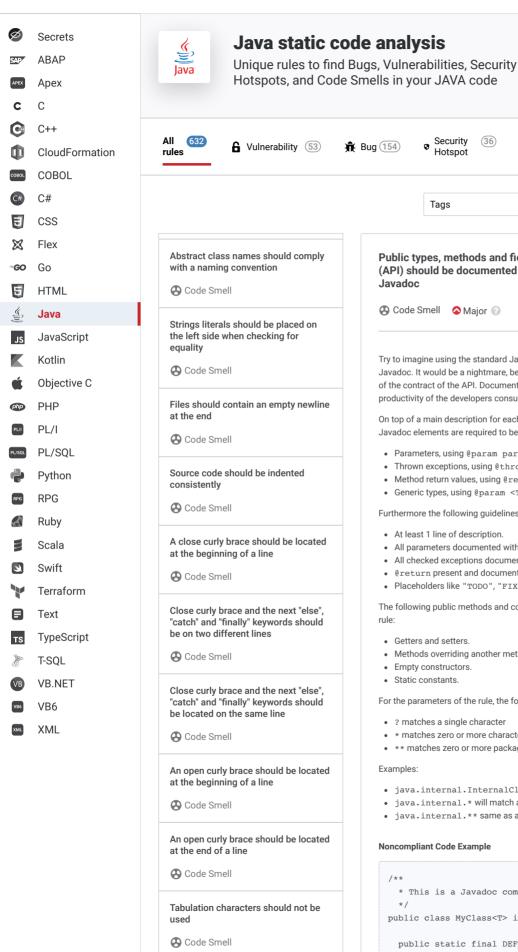
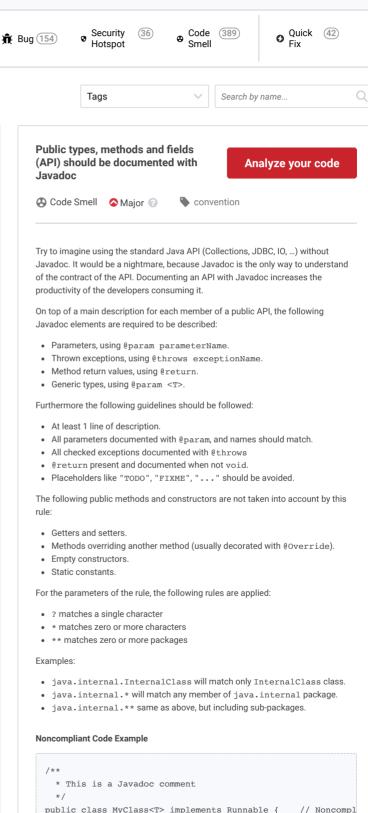


Products >





public static final DEFAULT STATUS = 0;

private int status;

public MyClass() {

public String message;

Functions should not be defined with

a variable number of arguments

Code Smell

// Compliant -

// Noncompliant

// Noncompliant

// Compliant

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

```
this.status = DEFAULT_STATUS;
}

public void setStatus(int status) { // Compliant - setter this.status = status;
}

@Override
public void run() { // Compliant - not public }

protected void doSomething() { // Compliant - not public }

public void doSomething2(int value) { // Noncompliant }

public int doSomething3(int value) { // Noncompliant return value;
}
}
```

## Compliant Solution

```
* This is a Javadoc comment
 * @param <T> the parameter of the class
public class MyClass<T> implements Runnable {
  public static final DEFAULT_STATUS = 0;
 private int status;
   * This is a Javadoc comment
  public String message;
  * Class comment...
 public MyClass() {
   this.status = DEFAULT_STATUS;
  public void setStatus(int status) {
  this.status = status;
  @Override
 public void run() {
  protected void doSomething() {
   * Will do something.
   * @param value the value to be used
  public void doSomething(int value) {
   * {@inheritDoc}
 public int doSomething(int value) {
   return value;
```

Available In:

sonarlint ⊕ | sonarcloud ↔ | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy