Cameron Watson Wiki

1. UnusedVariableChecker

The UnusedVariableChecker identifies unused class fields and local variables within Java classes, helping to clean up the code and improve its maintainability.

Inputs:
ClassNode: Represents the Java class file being analyzed.
Outputs:
String: A report listing all the unused variables found in the class.
How It Works:
Analyzes class fields and local variables in each method.
Determines whether each variable is referenced in the code.
Generates a report on variables that are declared but not used.

2. SRPChecker (Single Responsibility Principle Checker)

The SRPChecker ensures that each class in a Java project adheres to the Single Responsibility Principle, a key concept in object-oriented design which states that a class should have only one reason to change.

Inputs:
ClassNode: The Java class to analyze for adherence to SRP.
Outputs:
String: Assessment results indicating whether the class follows the Single Responsibility Principle.
How It Works:
Evaluates the responsibilities of a class based on its methods and properties.
Checks if the class is performing multiple unrelated tasks.
Reports classes that appear to have more than one responsibility.

3. FacadePatternChecker

This check verifies the implementation of the Facade Design Pattern in classes. The Facade Pattern provides a simplified interface to a complex subsystem.

Inputs:
ClassNode: The Java class to be checked for compliance with the Facade Pattern.
Outputs:
String: The result indicating whether the class correctly implements the Facade Pattern.
How It Works:
Identifies classes that aim to provide a simplified interface to other classes.
Analyzes the methods and interactions of the class with other components.
Determines if the class effectively encapsulates the complexities of the subsystem and provides a straightforward interface.