Question 1 - Q1Listener

The Q1Listener class is designed to track imported and used classes in a Java file. It identifies classes imported in the file and distinguishes them from classes that are actually used. This listener is particularly useful for detecting unused imports, helping in code optimization and cleanup. The 'enterImportDeclaration' method captures all imported classes, while 'enterUnannClassType_lfno_unannClassOrInterfaceType' and corresponding enter/exit methods for variable declarations help in identifying used classes.

Question 2 - Q2Listener

Q2Listener focuses on class inheritance in Java code. It creates a map of classes and their extended classes, providing a clear view of the inheritance hierarchy. This listener is particularly useful for analyzing and visualizing class relationships and dependencies. It makes use of the 'enterNormalClassDeclaration' and 'enterSuperclass' methods to track the current class and its superclass, respectively, and then organizes this information efficiently.

Question 3 - Q3Listener

The Q3Listener is designed to analyze methods within Java classes. It identifies empty methods, methods throwing UnsupportedOperationException, and normal methods. This listener is essential for code quality checks, helping to spot methods that might need further implementation or modification. By inspecting method declarations and their bodies, Q3Listener can efficiently categorize methods based on their implementation details.

Question 4 - Q4Listener

Q4Listener is tailored to filter out comments from Java code, specifically targeting

comments containing control structures like 'if', 'switch', and 'for'. This listener is useful in scenarios where comments with potential code snippets need to be cleaned up. It examines comments and excludes those containing the specified keywords, thereby streamlining the code documentation process and enhancing code readability.