**A comprehensive test plan for the "Indents Class" include the following:**

Test for correct indentation of a single file: This will test the basic functionality of the method by verifying if a single file's indentation is properly checked.

Test for multiple files: This will test the method's ability to handle and check indentation of multiple files in a directory.

Test for the handling of an empty directory: This will test the method's ability to handle and return an error message when no files are present in the directory.

Test the return values of all the methods: This will ensure that the methods return the expected values.

Test the edge cases: This will test the methods under extreme or unexpected conditions, such as large numbers of files or unexpected file types.

Test for compatibility: This will test the method's compatibility with different file types and operating systems.

Test for error handling: This will test the method's ability to handle and report errors, such as invalid file paths or incorrect indentation.

**As for the coverage criterion, the following elements:**

1.Test the return values of all the methods: This will ensure that the methods return the expected values.

2.Test the different conditions and branches in the code: This will ensure that all the code paths are executed and tested, including the handling of errors.

3.Test the edge cases: This will test the methods under extreme or unexpected conditions, such as large numbers of files or unexpected file types.

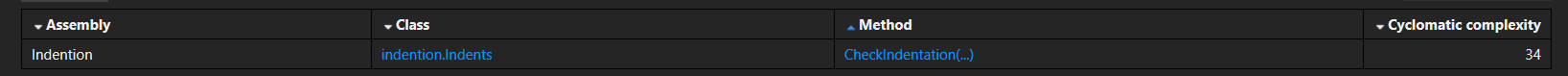
4.Test the performance: This will test the methods' performance under different scenarios, such as large directories with many files.

**Abnormality check:**

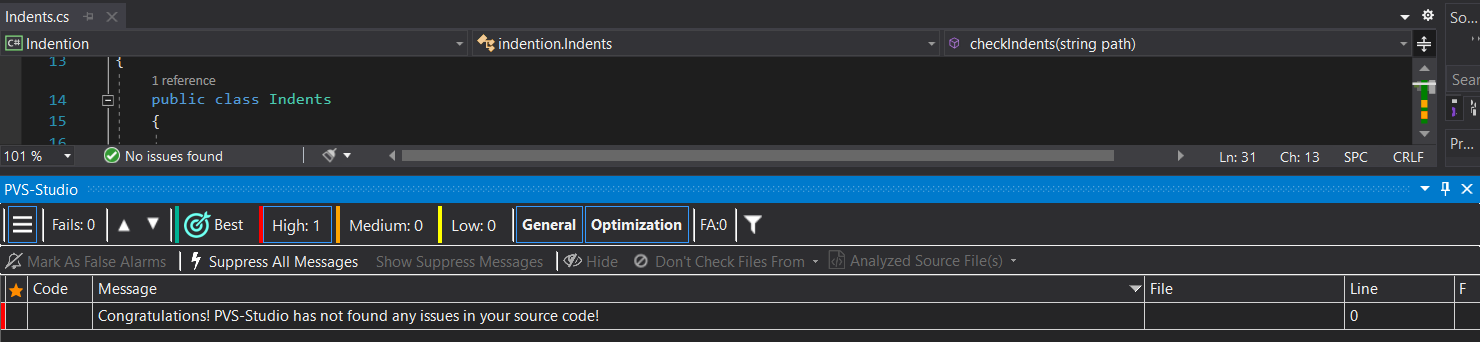
Run-time Errors: These are errors that occur when the code is executed. For example, a null reference exception or a division by zero error or DirectoryNotFoundException.

file format error: that occurs when a file is not correctly labeled with its proper extension, leading to issues when attempting to open or use the file leading to incorrect results or data loss.

### One resk is High Cyclomatic Complexity (34)which means that the function will be difficult to understand, and more difficult to test.(tool fine code coverage)



But over all there was no issues or risks in code.(tool PVS- studio)



The test percentage of the code that did we cover is 94.7% (tool fine code coverage)

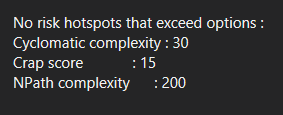




After improvements on the code 100% line coverage :

The changes: replace the nested if statement with robust function this reduced the cyclomatic complexity





איתרתנו שגיאות בקוד באמצעות הבדיקות שביצעתנו כמו אם הקובץ הוא בפורמאט לא נכון אז לא קורא את הקובץ דוגמה:

Test.12.c

לא תכין .