Antonio Scalfaro

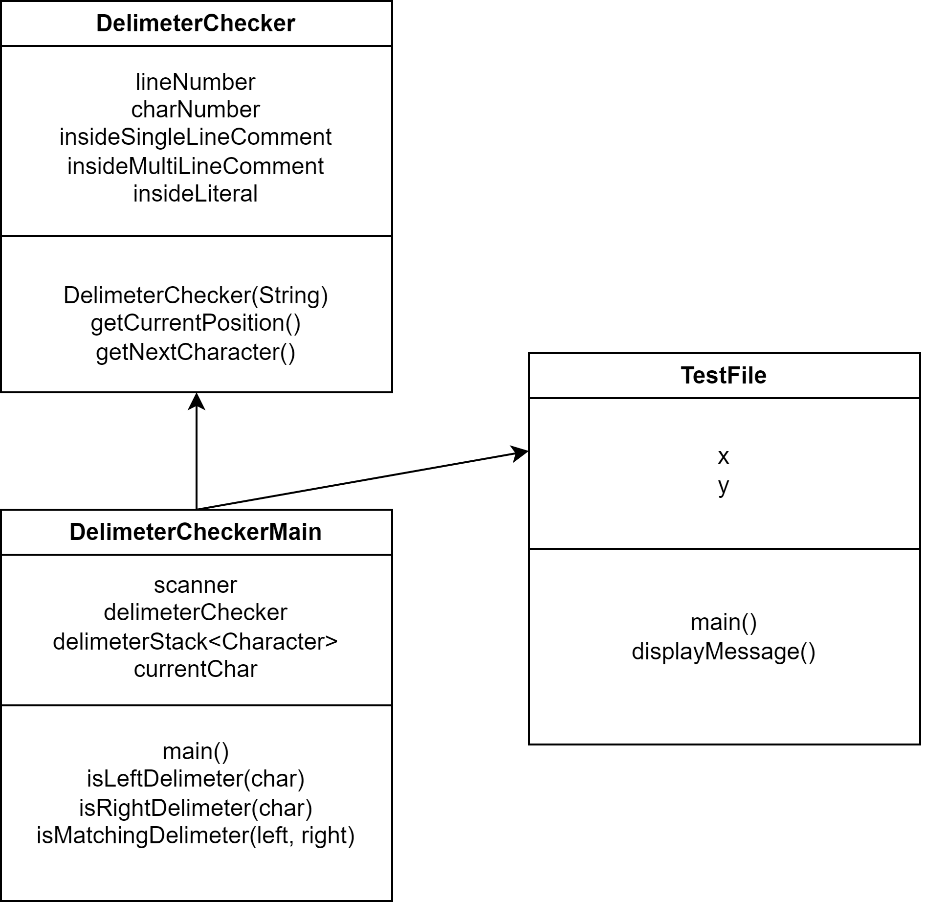
Project 1 – Delimiter Checker

CMSC 315

1/23/2024

This is the Delimiter Checker Program, it consists of three files, a TestFile.java, DelimeterChecker.java class, and a DelimeterCheckerMain.java class. It utilizes the native IOException Handling, FileReader, BufferedReader, Scanner, and Stack classes. It is designed for the user to input the name of a file, if the file name is invalid it will continue to ask until a valid name is given. Once a valid file is given, the program will read every line character by character from the file. The DelimeterCheckerMain class will check to see if each character is a delimiter and if so is it a right or left delimiter. If it is a left delimiter, it will be pushed onto the delimiterStack. If it is a right delimiter, it will pop the most recent delimiter from the stack and check to see if they are matching delimiters. If they are not, it will notify the user via a console message with the line number and character number of the last delimiter read in. Admittedly, this is slightly confusing because it will return the delimiter last read instead of the delimiter that was last unpaired. Accordingly, it will print out the line number and character number of the last delimiter and not the unpaired delimiter. To test this program, I used a TestFile.java file. For each test, I changed the file to adjust to test a new aspect of the program instead of creating separate test files. It passes all tests and behaves as designed.

**Class Diagram:**

****

**Test Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Input** | **Output** | **Expected Output** | **Pass** |
| **1** | “TestFile”, “TestFile.java” | “Enter the file name”, “Error opening File”, “Mismatched delimiter at position Line number: 30 Character number: 9 - Character: }” | Mismatched delimiter at position Line number: 30 Character number: 9 - Character: } | Pass |
| **2** | “TestFile.java” | “Enter the file name”, “Mismatched delimiter at position Line number: 29 Character number: 48 - Character: )” | Mismatched delimiter at position Line number: 29 Character number: 48 - Character: ) | Pass |
| **3** | “Afile.java”, “Testfile”, “TestFile.java” | “Enter the file name”, “Error opening file”, “Error opening file”, “Delimiters are balanced” | Delimiters are balanced | Pass |
| **4** | “TestFile.java” | “Enter the file name”, “Mismatched delimiter at position Line number: 16 Character number: 12 - Character: ]” | Mismatched delimiter at position Line number: 16 Character number: 12 - Character: ] | Pass |

**Test 1:**

A computer screen shot of a program

Description automatically generated

**Test 2:**

A computer screen shot of a program

Description automatically generated

**Test 3:**

A computer screen shot of a program

Description automatically generated

**Test 4:**

A computer screen with a black screen

Description automatically generated

**What was Learned?**

I did not have a ton to learn from this exercise, I am fairly proficient with the java language, file reading, and data structures. I did, however, find it difficult to get the perfect behavior out of the program. I noticed that if a right delimiter was missing, it would locate the next right delimiter and then spot the issue. This caused the program to get the location of the actual missing delimiter incorrect. If a left delimiter was missing, however, it would more accurately locate the issue at hand. This is an interesting issue and deserves more thought from me. It was a fun challenge and I look forward to future assignments pushing my knowledge even further.