Troy Cope

Concepts of Programming Languages

CS4308, Section 03

Deliverable 1, Scanner

March 6th, 2022

Sharon Perry

Project Deliverable is 100% completed and working as designed.

I choose to use Java for my interpreter implementation and approached it from the parallel list style. To begin, my program calls a helper class called FileManager. As the name might suggest, the whole point of the FileManager is to ensure that I do not have to worry about file issues in the main interpreter class. The FileManager has standard functions like setup and WriteInto which are the IO setup functions. It uses splitFileByLines to split the file, and is also used to handle tokens via splitLineByTokens. For input it simply reads each part of the file line by line. Splitting by the token is done through importing the target String line and splitting it via String functions.

In the interpreter, the input file is recognized or generated if not present. This is important to note because it ensures no errors occur and show where the file should be. The input file then gets parsed into lines and then into tokens via FileManager, then stored into an ArrayList. Once more these values are sectioned into an ArrayList to ensure formatting is consistent and more flexibility if necessary. In-between these functions, putSpaceAroundSelection and accountForFunctions are implemented in order to ensure that the formatting of spaces are perfect so that parsing by spaces is interpreted properly. Those function then bring us to the assignToken function.

The assignToken function is the table-lookup function of this, and therefore the backbone of the interpreter. Essentially, the function works in three stages:

1. Direct Value Lookup (Matching)
2. Boolean Flag Lookup (Conditionals)
3. Specific Outlier(s) (Conditionals)
4. Remaining Unknowns (Default)

The direct lookup functions exactly as expected, comparing the direct token value and pushing back the identifier code. The Boolean flag lookup is for specific orderings and/or the case of hard to distinguish sections. The specific outlier(s) is reserved for the functional declaration, because it can be any multitude of ID’s attached to parenthesis but can be purposed for any necessary edition. Lastly, the default is that the program cannot identify the token. This is arguably the most important because it accounts for any token that is slightly misplaced, and therefore is efficient for compile error detection.

The file then implements FileManager one more time to consider the output file. This class gets setup, as discussed earlier, and then writes the output lines directly to the file via WriteInto.

Input and Output Capture

Test1, input:

Graphical user interface, text

Description automatically generated

Test 1, output:

Text

Description automatically generated

Test2, input:

Text

Description automatically generated

Test2, output:

Text

Description automatically generated

Test3, input:

Text

Description automatically generated

Test3, output:

Text

Description automatically generated