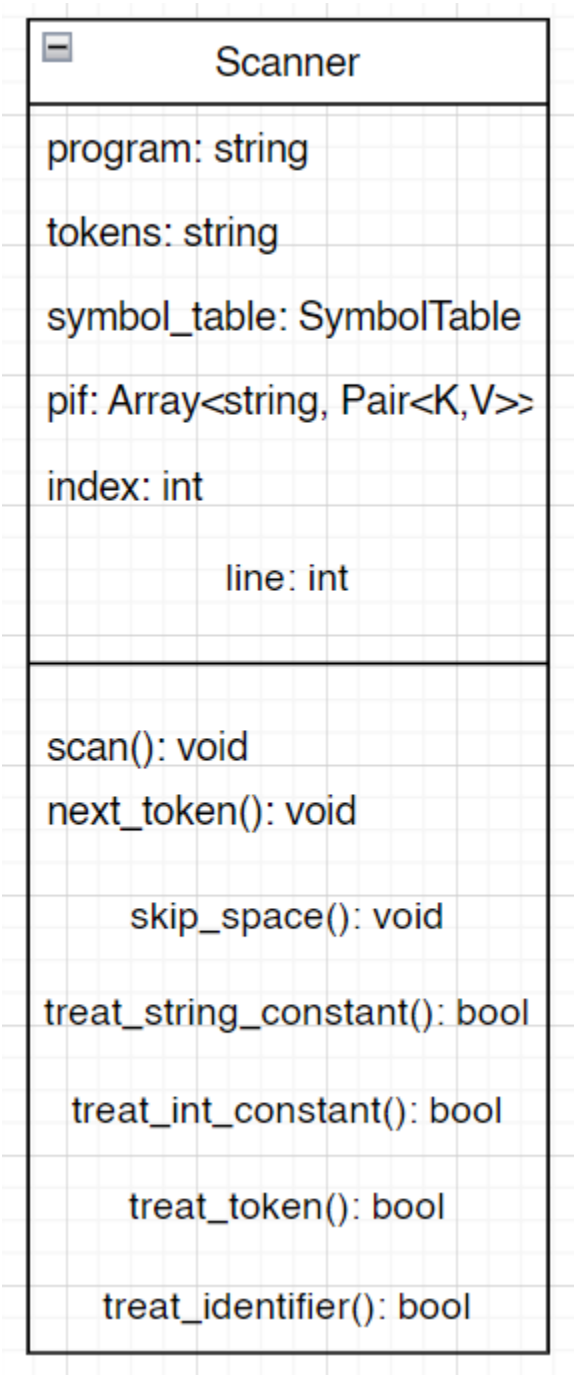
https://github.com/911-Berescu-Adrian/flcd

For my implementation of the scanner I chose to use an index, to keep track of the program given as input, a line, which is mainly used in case of a lexical error to print the line where there's an unknown token and the PIF is an array that contains the detected input and its position in the symbol table (for identifiers or constants) or -1 (token).

This is the class diagram for the Scanner:



To start the scanner the method scan() is called. It checks each token, as long as the length of the program is not exceeded. It eliminates spaces and newlines via the skip_space() method and then checks the detected input for all the possible cases: token, identifier, string/integer constant.

This classification is done using regex. If the detected input doesn't match any category then an error is raised. Otherwise, it adds the input to the symbol table (in case of an identifier or an integer/string constant) along with its position in the corresponding hash table and then it also gets added to the PIF with the position from the symbol table or -1 if token.