FLCD Scanner Documentation

Link to GitHub repository:

https://github.com/dianaaadumitru/FLCD/tree/main/lab3

Problem Requirements:

Statement: Implement a scanner (lexical analyzer): Implement the scanning algorithm and use ST from lab 2 for the symbol table.

Program Scanner:

The scanner receives the program as String and the list of tokens from token.in.

The class also contains a field index that keeps track of the position where the scanner currently is, a field currentLine, the symbol table and a list of pifs.

The scanner also checks for errors and raises a ScannerException if an error is found.

In the pif tokens correspond to value -1, identifiers to the position in the symbol table, int constants to -2 and string constants to -3.

Methods:

skipWhiteSpace() – skips characters that are considered " ". Returns true if any blank space was skipped, false otherwise.

isIntConstant() – uses regex to check if the current token is an integer constant. Returns true if it is a match, false otherwise.

isStringConstant() – uses reqex to check if the pattern matches (letters, digits, space and _ between quotes). If the pattern matches it is added to sym table and to the pif, otherwise an exception is thrown.

isTokenFromList() – checks if the current token is part of the tokens list and add it to the pif. Returns true if it is part of the list, false otherwise

isIdentifier() – identifiers are a sequence of letters and digits, such that the first character is a letter and they can also contain "_". Regex is used to check if the current token matches identifiers. Returns true if a match was found, false otherwise.

nextToken() – gets to the next token. If the current index is on a whites pace it skips it. If the token is an identifier, a token from the token.in file, a int constant or a string constant, it is added to the pif list. If the token is not found an exception is thrown having as message the error token and the line it was found on.

scan() – scans the whole program

writeToSTFile() – write the symbol table to a file (ST.out)

writeToPIFFile() – write the pif to a file (PIF.out)