https://github.com/cs-ubbcluj-ro/lab-work-computer-science-2024-SzilagyiBotond/tree/main/1-Mini-Language-And-Scanner/Scanner/Lab3

Scanner

In my implementation I choose to store in separate lists my operators, separators and my reserved words.

I also store a pattern, based on which I tokenize my input program. (The pattern is based on the operators and separators)

I also store in my scanner 3 instances of ST, one for the identifiers, one for the string constants and one for the integer constants.

I also have a PIFList instace in my scanner, where I store the program internal form and additional information about the data type of the token (necessary to identify the corresponding ST)

writePIFtoFile()
Writes the content of the PIFList (Program Internal Form list) to a file named PIF.OUT.

writeIdentifierSTtoFile()
Writes the content of the identifier symbol table to a file named IdST.OUT.

writeIntegersSTtoFile()
Writes the content of the integer constants symbol table to a file named IntST.OUT.

writeStringsSTtoFile()
Writes the content of the string constants symbol table to a file named StringST.OUT.

readFile()
Reads the source file specified in programPath and returns its contents as a string. Throws FileNotFoundException if the file is not found.

tokenize(String content)
Takes the file content as input and breaks it into tokens based on a regex pattern for delimiters (operators, separators, and string literals). Each token is stored with its line number as a Pair.

scan()
Processes each token to classify it as a reserved word, operator, separator, string constant, integer constant, or identifier. Adds each classified token to the PIF and the appropriate symbol table. Throws a LexicalError if an invalid token is encountered.

isStringConstant(String token)
Checks if the token is a string constant by verifying if it is enclosed in double or single quotes.

isIntegerConstant(String token)
Checks if the token represents a valid integer constant using a regex pattern.

isIdentifier(String token)
Checks if the token is a valid identifier using a regex pattern (starts with a letter and contains only alphanumeric characters).

PIFList

This contains a list of program internal forms as a Pair of String and Position. This class also contains the types of each symbol that ensures the matching of a symbol to a ST.

add(Pair<String,Position> symbol, Types type)
Adds a token and its type to the PIFList. Each token is represented by a Pair containing the token string and its position, and the type is added to the types list to maintain alignment with the tokens.