**Documentation**

**Github Link**:

<https://github.com/raresica1234/LFTC/tree/master/Lab3>

The Symbol Table uses the same representation as last time, a HashTable.

The Program Internal Form is just an Array List of PIFEntries which contains: token (the token that has been parsed), position (either -1/null in case the token is predefined, or the position in the symbol table in case it’s an identifier or a constant), currentLine (the line where the token has been found).

The LexicalAnalyzer works by reading the file line by line, then it loops until the file has ended or a lexical error has been found. Firstly it removes all of the beginning white spaces: spaces, tabs, or new lines. Then tries to match the beginning of the line with all of the token the program contains. If it does not find a token that matches it throws a lexical error. If it does find a token, and the token is not an identifier or a constant, it simply adds it. If the token is, in fact, an identifier or a constant, then it looks in the symbol table to check if it already exists. If it does, then it just creates a PIFEntry with that position and appends it to the PIF. If it doesn’t exist, then it adds it to the SymbolTable, then finds its position and adds it to the PIF.

