

<https://github.com/PeluzaVerde/FLCD>

(this one is the correct link)

Documentation for Scanner

1. Fields:

- 1.1. Reserved words –initialized as an arraylist of all the reserved words
- 1.2. Operators –initialized as an arraylist of all operators, including && and GE
- 1.3. Separators –initialized as an arraylist of all separators
- 1.4. `identifiersSymbolTable` – a symbol table responsible for memorizing identifiers with their hashed code aka position in bucket array
- 1.5. `ConstantsSymbolTable` – same as above, only that its used for constants only
- 1.6. Pif – a list of Pair of string and integer, string being the id/name aka 'a' 'read' '{' and the integer the position in the symbol table its from. If it's a token and not from a symbol table, its id is -1
- 1.7. File – String that shows what program is getting scanned. Parameter used to find the file
- 1.8. Index – used to go thru the line character by character, or by the length of the reserved word/operator
- 1.9. CurrentLine – The number of the line the scanner is currently at. Used for letting the user know at which line there is a lexical error
- 1.10. Line –The line is read in the scan function, being a line of code. After its done parsing thru it, it goes to the next line and resets the index.

2. Functions:

- 2.1. `SkipWhiteSpaces()` -Skips white spaces and advances the 'index' within the 'line' string
- 2.2. `StringConstantCase()`- Using Regex it makes sure it is a proper String. If it's a match to the regex(which is how all cases work) it adds it to the constants symbol table, adds it to the Pif using itself(the string) and its position from the symbol table, then returns true. If It doesn't match this returns false
- 2.3. `IntConstantscase()`- makes sure the expression is an int ie -4 or 543, then it checks if its an invalid integer (453MISTAKE5544). Adds it to pif like in the above function
- 2.4. `IdentifierCase()`- finds an identifier, which must start with a letter. IF this is in the reserved words list it returns false. if it already exists in the symbol table, then it just adds it to the pif with the same position index Otherwise it adds it to the IDENTIFIERS table and to pif.
- 2.5. `TokenCase()` - using a possibletoken string, it checks if it matches with operators, separators and reserved words, making sure to check for cases when there is a compound operator (ie GE, &&, = and ==)
- 2.6. `Next()` - we skip white spaces, if we are at the end of the line we also return. we check for the above 5 cases. If neither returns true, then it is a lexical error, throwing an error at the location of it.
- 2.7. `Scan()`- reading from a file, we parse the program , and for each line of code we call 'next'. After all the code was parsed, we print to two different files the symbol table contents and the PIF.