LEXICAL ANALYZER

**MAIN CODE**

import re

f = open("code.txt", "r")

count = 0

code = []

for i in f.read().split('\n'):

if i:

count += 1

code.append(i);

# print(code)

tokens = []

keywords = [

'abstract', 'continue', 'for', 'new', 'switch',

'assert', 'default', 'goto', 'package', 'synchronized',

'boolean', 'do', 'if', 'private', 'this',

'break', 'implements', 'protected', 'throw',

'byte', 'else', 'import', 'public', 'throws',

'case', 'enum', 'instanceof', 'return', 'transient',

'catch', 'extends', 'try',

'final', 'interface', 'static', 'void',

'class', 'finally', 'strictfp', 'volatile',

'const', 'native', 'super', 'while','System']

datatypes = ['double', 'int', 'short', 'char', 'long', 'float', 'bool', 'String']

lineno = 0

for line in code:

source\_code = str(line).split()

lineno += 1

# Loop through each source code word

for word in source\_code:

if word in keywords:

tokens.append([lineno, 'KEYWORD', word])

elif word in datatypes:

tokens.append([lineno,'DATATYPE', word])

elif re.match("[a-z]", word) or re.match("[A-Z]", word):

tokens.append([lineno,'IDENTIFIER', word])

elif word in '-/+%=\*':

tokens.append([lineno,'OPERATOR', word])

elif word in '(){}[]";':

tokens.append([lineno,'DELIMITER', word])

elif re.match(".[0-9]", word):

if word[len(word) - 1] == ';':

tokens.append([lineno,"INTEGER", word[:-1]])

else:

tokens.append([lineno,"INTEGER", word])

print('\n')

print('line no type\t \t name')

for token in tokens:

print(f'{token[0]} \t {token[1]} \t {token[2]}')

**CODE TEXT INPUT**

class Code {

public static void main ( String args [ ] ) {

System.out.println ( " Hello Java " ) ;

for(i=0; i<10; i++)

{

System.out.println ( " Experiment 1 " )

System.out.println ( " Lexical analyzer code " )

}

}

}

**OUTPUT**



