package lexical;

import java.io.IOException;

import java.io.FileInputStream;

import java.io.PushbackInputStream;

import java.util.\*;

public class LexicalAnalysis implements AutoCloseable {

private int line;

private SymbolTable st;

private PushbackInputStream input;

public LexicalAnalysis(String filename) throws LexicalException {

try {

input = new PushbackInputStream(new FileInputStream(filename));

} catch (Exception e) {

throw new LexicalException("Unable to open file");

}

st = new SymbolTable();

line = 1;

}

public void close() throws IOException {

input.close();

}

public int getLine() {

return this.line;

}

public Lexeme nextToken() throws IOException {

int estado = 1;

Lexeme lex = new Lexeme("", TokenType.END\_OF\_FILE);

while (estado != 9 && estado != 10) {

int c = input.read();

switch (estado) {

case 1:

if (c == ' ' || c == '\t' || c == '\r') {

} else if (c == '\n') {

line++;

} else if (c == '/') {

lex.token += (char) c;

estado = 2;

} else if (Character.isDigit(c)) {

lex.token += (char) c;

estado = 5;

} else if (c == '>' || c == '<' || c == '!' || c == '=') {

lex.token += (char) c;

estado = 6;

} else if (Character.isLetter(c)) {

lex.token += (char) c;

estado = 7;

} else if (c == '\"') {

lex.type = TokenType.STRING;

estado = 8;

} else if (c == ';' || c == ',' || c == '(' || c == ')'

|| c == '{' || c == '}' || c == '+' || c == '-'

|| c == '\*' || c == '%' || c == '&' || c == '|') {

lex.token += (char) c;

estado = 9;

} else if (c == -1) {

lex.type = TokenType.END\_OF\_FILE;

estado = 10;

} else {

lex.type = TokenType.INVALID\_TOKEN;

lex.token += (char) c;

}

break;

case 2:

if (c == '\*') {

lex.token += (char) c;

estado = 3;

} else {

input.unread(c);

estado = 9;

}

break;

case 3:

if (c == '\*') {

lex.token += (char) c;

estado = 4;

} else {

lex.token += (char) c;

estado = 3;

}

break;

case 4:

if (c == '\*') {

lex.token += (char) c;

estado = 4;

}

else if (c == '/'){

lex.token += (char) c;

estado = 1;

}

else{

lex.token += (char) c;

estado = 3;

}

break;

case 5:

if (Character.isDigit(c)) {

lex.token += (char) c;

estado = 5;

} else {

if (c != -1) {

input.unread(c);

}

estado = 10;

}

break;

case 6:

if (c == '=') {

lex.token += (char) c;

} else {

if (c != -1) {

input.unread(c);

}

}

break;

case 7:

if (Character.isDigit(c) || Character.isLetter(c)) {

lex.token += (char) c;

estado = 7;

} else {

if (c != -1) {

input.unread(c);

}

}

break;

case 8:

if (c == '\"') {

estado = 10;

}

else {

lex.token += (char) c;

estado = 8;

}

break;

default:

break;

}

}

if (estado == 9) {

//consulta tabela

lex.type = TokenType.NAME;

}

if (estado == 10) {

}

return lex;

}

}