package lexical;

import java.util.Map;

import java.util.HashMap;

class SymbolTable {

private Map<String, TokenType> st;

public SymbolTable() {

st = new HashMap<String, TokenType>();

// symbols

st.put("{", TokenType.OPEN\_CUR);

st.put("}", TokenType.CLOSE\_CUR);

st.put(",", TokenType.COMMA);

st.put(".", TokenType.DOT);

// keywords

st.put("if", TokenType.IF);

st.put("else", TokenType.ELSE);

st.put("while", TokenType.WHILE);

// operators

st.put("!", TokenType.NOT);

st.put("&", TokenType.AND);

st.put("|", TokenType.OR);

st.put("==", TokenType.EQUAL);

st.put("!=", TokenType.DIFF);

st.put(">", TokenType.GREATER);

st.put("<", TokenType.LOWER);

st.put(">=", TokenType.GREATER\_EQUAL);

st.put("<=", TokenType.LOWER\_EQUAL);

st.put("+", TokenType.ADD);

st.put("-", TokenType.SUB);

st.put("system", TokenType.SYSTEM);

st.put("self", TokenType.SELF);

st.put("args", TokenType.ARGS);

}

public boolean contains(String token) {

return st.containsKey(token);

}

public TokenType find(String token) {

return this.contains(token) ?

st.get(token) : TokenType.INVALID\_TOKEN;

}

}