Code:

```
import java.util.ArrayList;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

class Lexer{
   static final String NOUN = "[\\w]+";
   static final String KEYWORDS = "(if|then)";
   static final String VERB = "(hate|like)";

   static final Pattern PAT_NOUN = Pattern.compile(NOUN, Pattern.CASE_INSENSITIVE);
   static final Pattern PAT_KEYW = Pattern.compile(KEYWORDS,
Pattern.CASE_INSENSITIVE);
   static final Pattern PAT_VERB = Pattern.compile(VERB, Pattern.CASE_INSENSITIVE);

   List<String> keywords = new ArrayList<>();
   List<String> nouns    = new ArrayList<>();
   List<String> verbs    = new ArrayList<>();

   void parse_token(String token){

      // KEYWORDS
      Matcher m_keyw = PAT_KEYW.matcher(token);
      if(m_keyw.find()){
         //keyw found
         if(! keywords.contains(token)){
            keywords.add(token);
         }
         return;
      }

      // VERB
      Matcher m_verb = PAT_VERB.matcher(token);
      if(m_verb.find()){
         //verb found
         if(! verbs.contains(token)){
            verbs.add(token);
         }
```

```
            return;
        }

        // NOUN
        Matcher m_noun =  PAT_NOUN.matcher(token);
        if(m_noun.find()){
            // noun found
            if(! nouns.contains(token)) {
                nouns.add(token);
            }

            return;
        }

        throw new RuntimeException("invalid token: |" + token + "|");
    }


    String get_repr_token(String token){
        String ret = "-1";

        if(keywords.contains(token)){
            ret = "<k>";
        }
        else if(verbs.contains(token)){
            ret = "<V, " + verbs.indexOf(token) + ">";
        }
        else if(nouns.contains(token)){
            ret  = "<N, " + nouns.indexOf(token) + ">";
        }

        return ret;
    }
}
public class SPCC2 {
    public static void main(String[] args) {
        String input = "If dogs hate cats then they chase. " +
                "If cats like milk then they drink.";
```

```
        Lexer lexer = new Lexer();
        String[] tokens = input.split("([\\s]+|\\.[\\s]*)");
        // split on space and dot followed by zero or more space

        // Parsing
        for( String token : tokens){
           lexer.parse_token(token);
        }

        // Display
        List<String> output = new ArrayList<>();
        for(String token : tokens){
           output.add(lexer.get_repr_token(token));
        }

        System.out.println("Tokens: " + output.size());

        StringBuilder f_input = new StringBuilder();
        StringBuilder f_output = new StringBuilder();
        for(int i=0; i< tokens.length; i++){
           f_input.append(String.format(" %7s ", tokens[i]));
           f_output.append(String.format(" %7s ", output.get(i)));
        }

        System.out.println(f_input.toString());
        System.out.println(f_output.toString());

    }
}
```

Output:

Tokens: 14

If dogs hate cats then they chase If cats like milk then they drink

<k> <N, 0> <V, 0> <N, 1> <k> <N, 2> <N, 3> <k> <N, 1> <V, 1> <N, 4> <k> <N, 2> <N, 5>