

```

1 package LexicalAnalyzer;
2 /**
3  * This is the main driver class for the
4  * Lexical analyzer package
5  *
6  * @author Derek Trom
7  * @author Elena Corpus
8  * @version 1.0
9  * @since 2020-09-26
10 */
11 public class Driver {
12     private static String fileName = "";
13     /**
14      * This is Main and where the start
15      * of the package is for the Lexical
16      * analyzer
17      * @param args get fileName from command line
18      * @throws Exception no input file provided
19      */
20     public static void main (String[] args) throws
Exception {
21
22         /**
23          * Start of program drives the package
24          * Try getting filename from args[]
25          * Throws Exception if no program provided
26          */
27         try {
28             fileName = args[0];
29         } catch (Exception e) {
30             System.err.println("Please provide an input
program\n");
31             System.exit(1);
32         }
33
34         /**
35          * Create a new IOModule and give it the file to
read
36          */
37
38         IOModule io = new LexicalAnalyzer.IOModule(fileName
);
39         /**
40          * Create a new Lexer and give it the programText
read in by io
41          */
42         Lexer lexer = new LexicalAnalyzer.Lexer(io.
getProgramText());
43
44         /**

```

```
45      * Try reading symbols from the lexer
46      */
47      try {
48          /**
49           * While the Lexer is ready to read symbols
50           * get symbols using Lexer.getsym()
51           * @see Lexer
52           */
53          while (lexer.isReady()) {
54              // Get the next symbol from the Lexer
55              String symbol = lexer.getsym();
56              // If the Lexer didn't find anything, and
57              if it didn't find a COMMENT
58              if (!"".equals(symbol) && !LexicalAnalyzer.
59              Language.TOK_LP_COMMENT.equals(lexer.getDetectedToken())) {
60                  // Add a new match to IOModule's
61                  matches List
62                  io.addMatch(new LexicalAnalyzer.Match(
63                  lexer.getDetectedToken(), symbol));
64              }
65          }
66      } catch (Exception e) {
67          System.err.println(e.getMessage());
68          System.exit(1);
69      }
70      /**
71      * Print all matches from the list generated
72      */
73      io.printMatches();
74  }
75
```