## Luis Ricardo Reyes Villar

```
import java.io.FileNotFoundException;
3
     import java.io.FileReader;
     import java.io.IOException;
 4
 5
     import java.util.logging.Level;
 6
   import java.util.logging.Logger;
 7
     // @author LuisR
 8
     public class Main {
9 =
         public static void main(String[] args) {
10
             AnalizadorLex lexico = new AnalizadorLex();
              String dir = "C:\\Users\\manue\\OneDrive\\Documentos\\NetBeansProjects\\"
11
12
                     + "AnalizadorLexico\\src\\CODIGO2.txt";
13
             File file = new File(pathname: dir);
14
              try {
15
                  FileReader fr = new FileReader(file);
16
17
                 int n = fr.read();
                  String componente = "";
18
19
                 int q = 0;
20
                 int clase = 0;
                  String tipo = "";
21
22
                  boolean pendiente = true;
23
                  char c = (char) n;
24
25
                  while (n != -1) {
26
                     q = lexico.getEstado(q, c);
27
                      while (q < 100) {
                          if (!(c == 10 || c == 32 || c == 9)) {
28
                             componente += c;
29
30
31
                         n = fr.read();
32
                         c = (char) n;
33
                          q = lexico.getEstado(q, c);
34
35
                      switch (q) {
36
                          case 100 -> {
37
                             clase = 4;
38
                              tipo = 4 + "";
39
                              pendiente = true;
40
41
                          case 101 -> {
42
                             componente += c;
43
                              clase = 4;
44
                              tipo = 1 + "";
45
                              pendiente = false;
46
47
                          case 102 -> {
48
                              clase = 4;
```

```
tipo = 2 + "";
49
50
                               pendiente = false;
51
52
                           case 103 -> {
53
                              clase = 4;
54
                               tipo = 3 + "";
55
                               pendiente = false;
56
                           case 104 -> {
57
58
                              componente += c;
59
                               clase = 5;
                               tipo = 2 + "";
60
61
                               pendiente = false;
62
63
                           case 105 -> {
64
                               componente += c;
65
                               clase = 5;
                              tipo = 3 + "";
66
                              pendiente = false;
67
68
                           case 106 -> {
69
70
                              clase = 5;
                               tipo = 1 + "";
71
72
                              pendiente = true;
73
74
                           case 107 -> {
75
                              componente += c;
                               clase = 5;
76
                               tipo = 5 + "";
77
78
                              pendiente = false;
79
80
                           case 108 -> {
81
                              clase = 5;
                               tipo = 4 + "";
82
                               pendiente = true;
83
84
85
                           case 109 -> {
86
                              clase = 5;
                               tipo = 6 + "";
87
88
                               pendiente = false;
89
90
                           case 110 -> {
91
                               componente += c;
92
                               clase = 6;
93
                              tipo = 0 + "";
94
                              pendiente = false;
95
                          case 113 -> {
96
```

```
componente += c;
 98
                                clase = 7;
 99
                                tipo = 0 + "";
100
                                pendiente = false;
101
102
                            case 111 -> {
103
                                String temp = "";
104
                                for (int i = 0; i < componente.length(); i++) {</pre>
105
                                    char car = componente.charAt(index: i);
106
                                    boolean cond = car >= 65 && c <= 90 || car >= 97 &&
107
                                      car <= 122 || c >= 49 && c <= 57;
108
                                    if (cond) {
109
                                        temp += car;
110
111
                                tipo = lexico.guarda(componente: temp) + "";
112
113
                                if (Integer.parseInt(s: tipo) < 20) {
114
                                    clase = 1;
115
                                } else {
116
                                   clase = 2;
117
                                   pendiente = true;
118
119
                                tipo = "Dir TS";
120
121
                            case 112 -> {
122
                                clase = 3;
123
                                tipo = "Dir TS";
124
                                pendiente = true;
125
126
                            case 200 ->{
127
                               componente += c;
128
                                tipo = "Se esperaba un =";
129
130
                            case 201 ->{
131
                               componente += c;
132
                                tipo = "Carácter inválido";
133
134
135
                        if (!pendiente) {
136
                           n = fr.read();
137
                            c = (char) n;
138
                        System.out.println(componente + " | " + clase + " | " + tipo);
139
140
                        pendiente = false;
                        componente = "";
141
142
                        q = 0;
143
               } catch (FileNotFoundException ex) {
144
145
                   Logger.getLogger(name: Main.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
146
               } catch (IOException ex) {
147
                   Logger.getLogger(name: Main.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
148
149
```

150

```
// @author LuisR
2
      public class AnalizadorLex {
3
          private final String[] tablaSimbolos = new String[35];
5
          private final String[] alfabeto = {
              "L", "D", "+", "-", "*", "/", "<", ">", "=", ":", "b", ";", "otro"
6
7
8
          private final int[] q = new int[25];
<u>@</u>
          private final int[] f = new int[16];
10
          private final int[][] d = new int[9][13];
11
12
          public AnalizadorLex() {
              tablaSimbolos[0] = "PROGRAM"; tablaSimbolos[1] = "END";
13
14
              tablaSimbolos[2] = "STEP"; tablaSimbolos[3] = "TO";
15
              tablaSimbolos[4] = "ARRAY"; tablaSimbolos[5] = "INTEGER";
              tablaSimbolos[6] = "CASE"; tablaSimbolos[7] = "DO";
16
              tablaSimbolos[8] = "VAR"; tablaSimbolos[9] = "IF";
17
              tablaSimbolos[10] = "ELSE";tablaSimbolos[11] = "WHILE";
18
              tablaSimbolos[12] = "CONST"; tablaSimbolos[13] = "THEN";
19
              tablaSimbolos[14] = "CHAR"; tablaSimbolos[15] = "REPEAT";
20
21
              tablaSimbolos[16] = "BEGIN"; tablaSimbolos[17] = "REAL";
22
              tablaSimbolos[18] = "FOR"; tablaSimbolos[19] = "UNTIL";
              tablaSimbolos[20] = "suma"; tablaSimbolos[21] = "comparacion";
23
              tablaSimbolos[22] = "nl"; tablaSimbolos[23] = "n2";
24
25
              tablaSimbolos[24] = "rs";
26
              for (int i = 0; i < 10; i++) {
                  tablaSimbolos[25 + i] = i + "";
27
28
              for (int i = 0; i <= 8; i++) {
29
                  q[i] = i;
30
31
32
              int n = 100;
33
              for (int i = 9; i <= 22; i++, n++) {
34
                  q[i] = n;
35
              n = 200;
36
              for (int i = 23; i <= 24; i++, n++) {
37
                  q[i] = n;
38
39
              for (int i = 9, nl = 0; i < q.length; i++, nl++) {
40
                 f[nl] = q[i];
41
42
43
              d[0][0] = 7;
44
              d[0][1] = 8;
45
46
              d[0][2] = 101;
47
              d[0][3] = 102;
              d[0][4] = 103;
48
49
              d[0][5] = 1;
```

```
d[0][6] = 4;
50
51
              d[0][7] = 5;
              d[0][8] = 109;
52
53
              d[0][9] = 6;
54
              d[0][10] = 0;
55
              d[0][11] = 113;
              d[0][12] = 201;
56
57
              d[1][0] = 100;
58
59
              d[1][1] = 100;
60
              d[1][2] = 100;
61
              d[1][3] = 100;
62
              d[1][4] = 100;
63
              d[1][5] = 2;
64
              d[1][6] = 100;
65
              d[1][7] = 100;
              d[1][8] = 100;
66
              d[1][9] = 100;
67
              d[1][10] = 100;
68
              d[1][11] = 100;
69
70
              d[1][12] = 100;
71
72
              d[2][0] = 2;
73
              d[2][1] = 2;
74
              d[2][2] = 2;
75
              d[2][3] = 2;
76
              d[2][4] = 3;
77
              d[2][5] = 2;
              d[2][6] = 2;
78
79
              d[2][7] = 2;
              d[2][8] = 2;
80
81
              d[2][9] = 2;
82
              d[2][10] = 2;
83
              d[2][11] = 2;
84
              d[2][12] = 2;
85
86
              d[3][0] = 2;
              d[3][1] = 2;
87
              d[3][2] = 2;
88
              d[3][3] = 2;
89
              d[3][4] = 2;
90
91
              d[3][5] = 0;
92
              d[3][6] = 2;
93
              d[3][7] = 2;
94
              d[3][8] = 2;
95
              d[3][9] = 2;
96
              d[3][10] = 2;
97
              d[3][11] = 2;
98
              d[3][12] = 2;
```

```
99
                d[4][0] = 106;
100
                d[4][1] = 106;
101
102
                d[4][2] = 106;
                d[4][3] = 106;
103
104
                d[4][4] = 106;
                d[4][5] = 106;
106
                d[4][6] = 106;
107
                d[4][7] = 105;
108
                d[4][8] = 104;
109
                d[4][9] = 106;
                d[4][10] = 106;
110
111
                d[4][11] = 106;
112
                d[4][12] = 106;
113
114
                d[5][0] = 108;
115
                d[5][1] = 108;
116
                d[5][2] = 108;
                d[5][3] = 108;
117
118
                d[5][4] = 108;
                d[5][5] = 108;
119
                d[5][6] = 108;
120
                d[5][7] = 108;
121
122
                d[5][8] = 107;
123
                d[5][9] = 108;
                d[5][10] = 108;
124
                d[5][11] = 108;
126
                d[5][12] = 108;
127
                d[6][0] = 200;
128
129
                d[6][1] = 200;
130
                d[6][2] = 200;
131
                d[6][3] = 200;
132
                d[6][4] = 200;
133
                d[6][5] = 200;
134
                d[6][6] = 200;
135
                d[6][7] = 200;
136
                d[6][8] = 110;
                d[6][9] = 200;
137
                d[6][10] = 200;
138
                d[6][11] = 200;
139
                d[6][12] = 200;
140
141
142
                d[7][0] = 7;
                d[7][1] = 7;
143
144
                d[7][2] = 111;
145
                d[7][3] = 111;
146
                d[7][4] = 111;
147
                d[71[5] = 111:
```

```
147
               d[7][5] = 111;
               d[7][6] = 111;
148
               d[7][7] = 111;
149
               d[7][8] = 111;
150
151
               d[7][9] = 111;
               d[7][10] = 111;
152
153
               d[7][11] = 111;
154
               d[7][12] = 111;
155
156
               d[8][0] = 112;
157
               d[8][1] = 8;
               d[8][2] = 112;
158
159
               d[8][3] = 112;
160
               d[8][4] = 112;
161
               d[8][5] = 112;
162
               d[8][6] = 112;
               d[8][7] = 112;
163
164
               d[8][8] = 112;
165
               d[8][9] = 112;
               d[8][10] = 112;
166
               d[8][11] = 112;
167
168
               d[8][12] = 112;
169
170
171 -
           public int getEstado(int q, char c){
172
               String s;
173
               int n = 0;
               if (c >= 65 && c <= 90 || c >= 97 && c <= 122) {
174
                   s = "L";
175
176
               } else if (c >= 49 && c <= 57) {
177
                  s = "D";
178
               } else if (c == 43) {
179
                   s = "+";
180
               } else if (c == 45) {
                   s = "-";
181
               } else if (c == 42) {
182
                   s = "*";
183
184
               } else if (c == 47) {
185
                   s = "/";
186
               } else if (c == 60) {
187
                   s = "<";
               } else if (c == 62) {
188
189
                   s = ">";
190
               } else if (c == 58) {
191
                   s = ":";
192
               } else if (c == 10 || c == 32 || c == 9 || c == 13) {
193
                  s = "b";
194
               } else if (c == 59) {
195
                   s = ";";
```

```
196
               } else if (c == 61) {
                    s = "=";
197
198
               } else {
199
                    s = "otro";
200
201
               for (int i = 0; i < alfabeto.length; i++) {
202
                    if (s.equals(alfabeto[i])) {
203
                        n = i;
204
                        break;
205
                    }
206
               return d[q][n];
207
208
209
210 =
           public int guarda(String componente) {
211
               int n = 20;
212
               for (int i = 0; i < 20; i++) {
                    String palabra = tablaSimbolos[i];
213
214
                   if (!componente.equals(anObject:palabra)) {
215
                    } else {
                       return i;
216
217
                    }
218
219
               return n;
220
221
       }
```