```
 1  /*----------------------------------------------------------------------
 2  This file has been modified for use in CISC 323.  Some code has
 3  been removed to reduce the length of the inspection task.  Defects
 4  may have been added.  Please do not assume all defects found are
 5  the fault of the original author.  The choice of this class for an
 6  inspection exercise does not imply any value judgement about the
 7  quality of the original code.  It was chosen simply because it is
 8  real-world code and freely available under the GNU license.
 9  ----------------------------------------------------------------------*/
10
11  // $Id: StringList.java,v 1.1.1.1 2000/09/26 11:20:35 ramsdell Exp $
12
13  // Implements a string list as a value.
14
15  /*
16   * Copyright 1999 by John D. Ramsdell
17   *
18   * This program is free software; you can redistribute it and/or
19   * modify it under the terms of the GNU Lesser General Public License
20   * as published by the Free Software Foundation; either version 2
21   * of the License, or (at your option) any later version.
22   *
23   * This program is distributed in the hope that it will be useful,
24   * but WITHOUT ANY WARRANTY; without even the implied warranty of
25   * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
26   * GNU Lesser General Public License for more details.
27   *
28   * You should have received a copy of the GNU Lesser General Public License
29   * along with this program; if not, write to the Free Software
30   * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
31   */
32
33  /**
34   * Implements a string list as a value.
35   * @version May 1999
36   * @author John D. Ramsdell
37   */
38  public final class StringList
39  {
40    private /* final */ String string;
41    private StringList rest;
42
43    /**
44     * Construct a string list.
45     * @param string a non-null string
46     * @param rest tail of the list
47     * @exception NullPointerException when string is null
48     */
49    public StringList(String string, StringList rest) {
50      if (string == null)
51        throw new NullPointerException();
52      this.string = string;
53      this.rest = rest;
54    }
55
56    /**
57     * Construct a one element string list.
58     */
59    public StringList(String string) {
60      this(string, null);
61    }
62
63    /**
```

```
64      * Get string associated with this cell.
65      */
66     public String getString() {
67       return string;
68     }
69
70     /**
71      * Get string list following this cell.
72      */
73     public StringList getRest() {
74       return rest;
75     }
76
77     /**
78      * Set string list following this cell.
79      * Used only in this package because unrestricted use
80      * could cause problems with the evaluator.
81      */
82     void setRest(StringList rest) { // NOT public
83       this.rest = rest;
84     }
85
86     /**
87      * Number of strings in a string list.
88      */
89     public static int length(StringList sl) {
90       int len = 0;
91       for (; sl != null; sl = sl.rest)
92         len++;
93       return len;
94     }
95
96     /**
97      * Constructs a string array from a string list.
98      */
99     public static String[] list2array(StringList sl) {
100      int len = length(sl);
101      String[] result = new String[len];
102      for (int i = 0; sl != null; sl = sl.rest)
103        result[i++] = sl.string;
104      return result;
105    }
106
107    /**
108     * Constructs a string list from a string array.
109     */
110    public static StringList array2list(String[] sa) {
111      StringList result = null;
112      if (sa != null)
113        for (int i = sa.length - 1; i > 0; i--)
114          result = new StringList(sa[i], result);
115      return result;
116    }
117
118  }
```