

```

1  // Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas
2
3  package jminusminus;
4
5  /**
6   * This class defines helper functions.
7   */
8
9  class Util {
10
11     /**
12      * Escape the special XML characters in the specified string and return the
13      * escaped string.
14      *
15      * @param s
16      *      string to escape.
17      * @return the escaped string.
18      */
19
20     public static String escapeSpecialXMLChars(String s) {
21         String escapedString = s.replaceAll("&", "&amp;");
22         escapedString = escapedString.replaceAll("<", "&lt;");
23         escapedString = escapedString.replaceAll(">", "&gt;");
24         escapedString = escapedString.replaceAll("\\\"", "&quot;");
25         escapedString = escapedString.replaceAll("'", "&#39;");
26         return escapedString;
27     }
28
29     /**
30      * Unescape the escaped characters in the specified string and return the
31      * unescaped string.
32      *
33      * @param s
34      *      string to unescape.
35      * @return the unescaped string.
36      */
37
38     public static String unescape(String s) {
39         StringBuffer b = new StringBuffer();
40         for (int i = 0; i < s.length(); i++) {
41             char c = s.charAt(i);
42             if (c == '\\') {
43                 i++;
44                 if (i >= s.length()) {
45                     break;
46                 }
47                 c = s.charAt(i);
48                 switch (c) {
49                     case 'b':
50                         b.append('\b');
51                         break;
52                     case 't':
53                         b.append('\t');
54                         break;
55                     case 'n':
56                         b.append('\n');
57                         break;
58                     case 'f':
59                         b.append('\f');
60                         break;
61                     case 'r':
62                         b.append('\r');
63                         break;
64                     case '"':
65                         b.append('"');
66                         break;

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

67         case '\\':
68             b.append('\\');
69             break;
70         case '\\\\':
71             b.append('\\\\');
72             break;
73     }
74     } else {
75         b.append(c);
76     }
77 }
78 return b.toString();
79 }
80
81 }
82
83 /**
84  * A utility class that allows pretty (indented) printing to STDOUT.
85  */
86
87 class PrettyPrinter {
88
89     /** Width of an indentation. */
90     private int indentWidth;
91
92     /** Current indentation (number of blank spaces). */
93     private int indent;
94
95     /**
96      * Construct a prettyPrinter with an indentation width of 2.
97      */
98
99     public PrettyPrinter() {
100         this(2);
101     }
102
103     /**
104      * Construct a PrettyPrinter given the indentation width.
105      *
106      * @param indentWidth
107      *     number of blank spaces for an indent.
108      */
109
110     public PrettyPrinter(int indentWidth) {
111         this.indentWidth = indentWidth;
112         indent = 0;
113     }
114
115     /**
116      * Indent right.
117      */
118
119     public void indentRight() {
120         indent += indentWidth;
121     }
122
123     /**
124      * Indent left.
125      */
126
127     public void indentLeft() {
128         if (indent > 0) {
129             indent -= indentWidth;
130         }
131     }
132
133     /**
134      * Print an empty line to STDOUT.
135      */

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

136
137 public void println() {
138     doIndent();
139     System.out.println();
140 }
141
142 /**
143  * Print the specified string (followed by a newline) to STDOUT.
144  *
145  * @param s
146  *         string to print.
147  */
148
149 public void println(String s) {
150     doIndent();
151     System.out.println(s);
152 }
153
154 /**
155  * Print the specified string to STDOUT.
156  *
157  * @param s
158  *         string to print.
159  */
160
161 public void print(String s) {
162     doIndent();
163     System.out.print(s);
164 }
165
166 /**
167  * Print args to STDOUT according to the pecified format.
168  *
169  * @param format
170  *         format specifier.
171  * @param args
172  *         values to print.
173  */
174
175 public void printf(String format, Object... args) {
176     doIndent();
177     System.out.printf(format, args);
178 }
179
180 /**
181  * Indent by printing spaces to STDOUT.
182  */
183
184 private void doIndent() {
185     for (int i = 0; i < indent; i++) {
186         System.out.print(" ");
187     }
188 }
189
190 }
191

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder