

10/26/15 02:00:57 /home/15504319/DSA120/DSAAssignment/FileIO.java

```

1  /*****
2  *   FILE: FileIO.java
3  *   AUTHOR: Connor Beardsmore - 15504319
4  *   UNIT: DSA120 Assignment S2- 2015
5  *   PURPOSE: Handler Class For All FILE IO
6  *   LAST MOD: 22/10/15
7  *   REQUIRES: java.io, connorLib
8  *****/
9  import java.io.*;
10 import connorLib.*;
11
12 public class FileIO
13 {
14     private static final String FILE_SUFFIX = "-output";
15     //-----
16     //readDCFile
17     //IMPORT: filename (String), roomList + itemList (DSALinkedList)
18     //PURPOSE: Read DC File strings into a linked list.
19     //ASSERTION: roomList contains strings before '%', itemList contains
20     //             strings after '%', excluding empty lines and comments
21
22     public static void readDCFile(String filename, DSALinkedList roomList,
23                                   DSALinkedList itemList) throws IOException
24     {
25         FileInputStream fs = null;
26         InputStreamReader isr;
27         BufferedReader br;
28
29         try
30         {
31             //Open all streams
32             fs = new FileInputStream(filename);
33             isr = new InputStreamReader(fs);
34             br = new BufferedReader(isr);
35
36             //Called Twice. One For DC section, one for carton section
37             //Easier than re-opening file and seeking to correct spot
38             readFileToList( br, roomList );
39             readFileToList( br, itemList );
40             fs.close();
41         }
42         catch (IOException e)
43         {
44             if ( fs != null )
45             {
46                 fs.close();
47             }
48             throw new IOException("error reading DC file");
49         }
50     }
51     //-----
52     //readTaskFile
53     //IMPORT: filename (String), taskList (DSALinkedList)
54     //PURPOSE: Read Task File strings into a linked list.
55     //COMMENT: Code reuse from readDCFile is high, cannot avoid due to format
56
57     public static void readTaskFile(String filename, DSALinkedList taskList)
58                                   throws IOException
59     {
60         FileInputStream fs = null;
61         InputStreamReader isr;
62         BufferedReader br;
63
64         try
65         {
66             //Open all streams
67             fs = new FileInputStream(filename);
68             isr = new InputStreamReader(fs);
69             br = new BufferedReader(isr);
70
71             readFileToList( br, taskList );
72             fs.close();
73         }
74         catch (IOException e)
75         {
76             if ( fs != null )
77             {
78                 fs.close();
79             }
80             throw new IOException("error reading Task file");
81         }
82     }
83     //-----
84     //readFileToList

```

```

85 //IMPORT: br (BufferedReader), list (DSALinkedList)
86 //PURPOSE: Read strings from br into a linked list for later access
87
88 private static void readFileToList(BufferedReader br, DSALinkedList list)
89                                     throws IOException
90 {
91     String newLine = new String ( br.readLine() );
92     newLine = newLine.trim();
93     //Stop at the specified seperator '%' line
94     while ( ( newLine != null) && (!newLine.equals("%")) )
95     {
96         //Cut all leading and trailing whitespace
97         newLine = newLine.trim();
98         //Ignore 'comment' lines, and empty lines
99         if ( !newLine.startsWith("#") )
100         {
101             if ( newLine.length() > 0 )
102             {
103                 list.insertLast( newLine );
104             }
105         }
106         newLine = br.readLine();
107     }
108 }
109 //-----
110 //writeOutput
111 //IMPORT: dc (DistroCentre), filename (String)
112 //PURPOSE: Write total contents of DC to output file
113
114 public static void writeOutput(DistroCentre dc, String filename)
115                                 throws IOException
116 {
117     String outputFile = filename + FILE_SUFFIX;
118     FileOutputStream fs = null;
119     PrintWriter pw;
120
121     try
122     {
123         fs = new FileOutputStream(outputFile);
124         pw = new PrintWriter(fs);
125         //DC function does all the format work
126         pw.write( dc.createOutString() );
127         pw.close();
128     }
129     catch (IOException e)
130     {
131         if ( fs != null )
132         {
133             fs.close();
134         }
135         throw new IOException("error writing output file");
136     }
137 }
138 //-----
139 }

```