

10/26/15 02:01:12 /home/15504319/DSA120/DSAAssignment/DCHandler.java

```

1  /*****
2  *   FILE: DCHandler
3  *   AUTHOR: Connor Beardsmore - 15504319
4  *   UNIT: DSA120 Assignment S2- 2015
5  *   PURPOSE: Handles DC Tasks
6  *   LAST MOD: 19/10/15
7  *   REQUIRES: java.util.Iterator, java.io.IOException, connorLib
8  *****/
9  import java.io.IOException;
10 import java.util.Iterator;
11 import connorLib.*;
12
13 public class DCHandler
14 {
15     //createDC
16     //IMPORT: filename (String)
17     //EXPORT: dc (DistroCentre)
18     //PURPOSE: Creates and populates a new DC, based on date in 'filename'
19
20     public static DistroCentre createDC( String filename )
21         throws IOException
22     {
23         DSALinkedList roomList = new DSALinkedList();
24         DSALinkedList itemList = new DSALinkedList();
25
26         //Reads file into two linked lists
27         FileIO.readDCFile( filename, roomList, itemList );
28         //Constructs DC now number rooms is known
29         DistroCentre dc = new DistroCentre( roomList.getLength() );
30         //Populate Index Array with items, and DC with new empty rooms
31         populateArray( dc, itemList );
32         populateDC( dc, roomList );
33
34         //Ensure top and bottom Cartons match up
35         if ( itemList.getLength() != dc.totalItems() )
36         {
37             throw new IllegalArgumentException("mismatch item/s in DC file");
38         }
39
40         return dc;
41     }
42     //-----
43     //populateArray
44     //IMPORT: dc (DistroCentre), itemList (DSALinkedList)
45     //PURPOSE: Creates and populates DC's index array, based on itemList
46
47     private static void populateArray( DistroCentre dc,
48                                       DSALinkedList itemList )
49     {
50         String newLine;
51         Carton newItem;
52         Iterator iter = itemList.iterator();
53
54         //Iterates over itemList, creating Cartons and adding to array
55         while ( iter.hasNext() )
56         {
57             newLine = (String)iter.next();
58             newItem = new Carton( newLine );
59             dc.setCartonIndex(newItem);
60         }
61     }
62     //-----
63     //createDC
64
65     //IMPORT: dc (DistroCentre), roomList (DSALinkedList)
66     //PURPOSE: Creates and populates a new DC, based on roomList
67
68     private static void populateDC(DistroCentre dc, DSALinkedList roomList)
69     {
70         String newLine, roomType;
71         Iterator iter = roomList.iterator();
72         int items, total, spaces, ii = 0;
73
74         while ( iter.hasNext() )
75         {
76             //Tokenize line
77             newLine = (String)iter.next();
78             newLine = newLine.trim();
79             String[] tokens = newLine.split(":", -1);
80
81             //Create Room based on first token, i.e. D/R/Y
82             roomType = tokens[0];
83             createRoom(roomType, tokens.length - 1, dc);
84
85             //Populate the new room with Cartons from the indexArray in DC

```

```

85         IStockRoom room = dc.getStockRoom(ii);
86         populateRoom(room, dc, tokens, ii);
87         ii++;
88     }
89 }
90 //-----
91 //createRoom
92 //IMPORT: type (String), length (int), dc (DistroCentre)
93 //PURPOSE: Creates and populates a new DC, based on date in 'filename'
94
95 private static void createRoom(String type, int length, DistroCentre dc)
96 {
97     //Creates Room depending on type declared in the start of line
98     if ( type.equals("D") )
99     {
100         dc.createDeadEnd(length);
101     }
102     else if ( type.equals("R") )
103     {
104         dc.createRolling(length);
105     }
106     else if ( type.equals("Y") )
107     {
108         dc.createYard(length);
109     }
110     else
111     {
112         String errMessage = ("Invalid room type in DC Geometry");
113         throw new IllegalArgumentException( errMessage );
114     }
115 }
116 //-----
117 //populateRoom
118 //IMPORT: room (IStockRoom), dc (DistroCentre, tokens (String), dIndex (int)
119 //PURPOSE: Populates a stockroom with a new carton
120
121 private static void populateRoom(IStockRoom room, DistroCentre dc,
122                                 String[] tokens, int dIndex)
123 {
124     for ( int ii = 1; ii < tokens.length; ii++ )
125     {
126         String nextCart = tokens[ii];
127         if ( !nextCart.equals("") )
128         {
129             int newNote;
130             //Catch possible erros in the room description section
131             try
132             {
133                 newNote = Integer.parseInt(nextCart);
134             }
135             catch (NumberFormatException e)
136             {
137                 throw new NumberFormatException("Con Note Must Be Numbers Only");
138             }
139
140             Carton newCarton = dc.getCartonIndex(newNote);
141
142             if (newCarton == null)
143             {
144                 String errMessage = ("Carton doesn't exist in DC file");
145                 throw new IllegalArgumentException( errMessage );
146             }
147             //If it's a yard, add to specific index, to allow empty slots
148             if ( room instanceof Yard )
149             {
150                 ((Yard)room).addCarton(newCarton, ii - 1);
151             }
152             //If not a yard, we place in whatever we are allowed to
153             else
154             {
155                 room.addCarton(newCarton);
156             }
157             newCarton.setDIndex(dIndex);
158         }
159     }
160 }
161 //-----
162 }

```