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

```java
/***************************************************************************
*    FILE: FileIO.java
*    AUTHOR: Connor Beardsmore - 15504319
*    UNIT: DSA120 Assignment S2- 2015
*    PURPOSE: Handler Class For All FILE IO
*    LAST MOD: 22/10/15
*  REQUIRES: java.io, connorLib
***************************************************************************/
import java.io.*;
import connorLib.*;

public class FileIO
{
    private static final String FILE_SUFFIX = "-output";
//--------------------------------------------------------------------------
    //readDCFile
    //IMPORT: filename (String), roomList + itemList (DSALinkedList)
    //PURPOSE: Read DC File strings into a linked list.
    //ASSERTION: roomList contains strings before '%', itemList contains
    //             strings after '%', excluding empty lines and comments

    public static void readDCFile(String filename, DSALinkedList roomList,
                             DSALinkedList itemList) throws IOException
    {
        FileInputStream fs = null;
        InputStreamReader isr;
        BufferedReader br;

        try
        {
            //Open all streams
            fs = new FileInputStream(filename);
            isr = new InputStreamReader(fs);
            br = new BufferedReader(isr);

            //Called Twice. One For DC section, one for carton section
            //Easier than re-opening file and seeking to correct spot
            readFileToList( br, roomList );
            readFileToList( br, itemList );
            fs.close();
        }
        catch (IOException e)
        {
            if ( fs != null )
            {
                fs.close();
            }
            throw new IOException("error reading DC file");
        }
    }
//--------------------------------------------------------------------------
    //readTaskFile
    //IMPORT: filename (String), taskList (DSALinkedList)
    //PURPOSE: Read Task File strings into a linked list.
    //COMMENT: Code reuse from readDCFile is high, cannot avoid due to format

    public static void readTaskFile(String filename, DSALinkedList taskList)
                                            throws IOException
    {
        FileInputStream fs = null;
        InputStreamReader isr;
        BufferedReader br;


        try
        {
            //Open all streams
            fs = new FileInputStream(filename);
            isr = new InputStreamReader(fs);
            br = new BufferedReader(isr);

            readFileToList( br, taskList );
            fs.close();
        }
        catch (IOException e)
        {
            if ( fs != null )
            {
                fs.close();
            }
            throw new IOException("error reading Task file");
        }
    }
//--------------------------------------------------------------------------
    //readFileToList
    //IMPORT: br (BuffReader), list (DSALinkedList)
    //PURPOSE: Read strings from br into a linked list for later access

    private static void readFileToList(BufferedReader br, DSALinkedList list)
                                                         throws IOException
    {
        String newLine = new String ( br.readLine() );
        newLine = newLine.trim();
        //Stop at the specified seperator '%' line
        while ( (newLine != null) && (!newLine.equals("%")) )
        {
            //Cut all leading and trailing whitespace
            newLine = newLine.trim();
            //Ignore 'comment' lines, and empty lines
            if ( !newLine.startsWith("#") )
            {
                if ( newLine.length() > 0 )
                {
                    list.insertLast( newLine );
                }
            }
            newLine = br.readLine();
        }
    }
//--------------------------------------------------------------------------
    //writeOutput
    //IMPORT: dc (DistroCentre), filename (String)
    //PURPOSE: Write total contents of DC to output file

    public static void writeOutput(DistroCentre dc, String filename)
                                                throws IOException
    {
        String outputFile = filename + FILE_SUFFIX;
        FileOutputStream fs = null;
        PrintWriter pw;

        try
        {
            fs = new FileOutputStream(outputFile);
            pw = new PrintWriter(fs);
            //DC function does all the format work
            pw.write( dc.createOutString() );
            pw.close();
        }
        catch (IOException e)
        {
            if ( fs != null )
            {
                fs.close();
            }
            throw new IOException("error writing output file");
        }
    }
//--------------------------------------------------------------------------
}
```