20.8 Program 4a: WriteCSV

Objectives

* Practice reading and writing from file with wrapper file/stream classes.
* Use try/catch clauses to gracefully handle error conditions.
* Convert a file type by renaming the extension.

Background Reading

ZyBooks Chapter 9 Exceptions, and Chapter 10 Streams

Instructions

Write a program WriteCSV.java that reads in a text file line by line, replaces all spaces with commas, and writes new lines to a new CSV file.

This program contains a template from which you will begin coding. There is not a lot of code to write, but many things to practice.

* Develop the code in an IDE (Eclipse or your favorite).
* Periodically submit your program for grading. This will help keep you from going too far off track.
* When you are satisfied with the program (or are out of time and want some partial credit), upload the source file WriteCSV.java.
  + See the "Uploading source file to ZyBooks lab" video on our Blackboard course web site for help.
* Your program will be graded automatically against the requirements.
* You may submit as many times as necessary.
* The automatic grading program is very specific. If you feel you have the correct solution but are not receiving full credit, please
  + Carefully review the output -- you might need to scroll all the way to the right to find what is wrong with a particular output.
  + Verify you have the correct names for the program itself and all methods.
  + Check your calculations by hand: was there a logic error?
  + Review the requirements: did you miss a step? misinterpret a requirement?
  + If all these check out, contact the T.A. for assistance.

Program Requirements

(0) Write a method getIdentificationString() that returns a string containing the programming assignment and your name. DO NOT call this method. Just make sure it is there and working correctly. The automatic grader will verify it.

Ex.

Program 4a, Patty Kraft

(1) Your task is to fix all the "TODO"s in the following code.

// TODO import statements

public class WriteCSV {

public static void main(String[] args) {

// Grading program needs hardcoded filename. Oh, well. "

String inputFilename = "coords.txt";

String outputFilename = changeFileExtToCsv(inputFilename);

// Open files

Scanner input = //TODO: call method to open input file

PrintWriter output = //TODO: call method to open output file

// TODO: Read input line, replace all spaces with commas,

// and write output line

while (input.hasNextLine()) {

}

// TODO: close streams

}

/\*\*

\* Changes file extension to ".csv"

\* @param filename

\* @return

\*/

public static String changeFileExtToCsv(String filename) {

return filename.substring(0,filename.lastIndexOf('.')) + ".csv";

}

/\*\*

\* Open input for reading

\* @param filename

\* @return

\*/

public static Scanner openInput(String filename) {

Scanner in = null;

try {

File infile = new File(filename);

in = new Scanner(infile);

} catch (FileNotFoundException e) {

//e.printStackTrace();

System.out.println(filename + " could not be found");

System.exit(0);

}

return in;

}

/\*\*

\* Open output for writing

\* @param filename

\* @return

\*/

public static PrintWriter openOutput(String filename) {

//TODO: Write method to open a PrintWriter ; use openInput() as a guide

}

}