SFWR ENG/COMP SCI 2S03 Principles of Programming Assignment 4

Due on Monday, December 03, 2012

This assignment paper includes 3 pages and 1 questions. You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of the instructor.

Special Instructions :

1. The burden of communication is upon you. Solutions not properly explained will not be considered correct. Part of proper communication is the appearance and layout. If we cannot "decode" what you wrote, we cannot grade it as a correct answer.
2. When writing Java programs, you should adhere to *The Elements of Java Style* by Vermeulen et al. **Your reasonable compliance with the** *Elements of Java Style* **will be taken into account in grading your programs.**
3. You are permitted to discuss *general aspects* of the problem sets with other students in the class, but each person must hand in her own copy of the solution. Any use of **any** source **must** be documented in the assignment log.
4. Your solutions to the problems should be submitted through the course website using your account and your password. More details on submitting assignment solutions can be found on the course website.
5. The assignments programs will be due **at the beginning of the lecture** on the due date. The system will automatically refuse accepting submissions after the beginning of the lecture on the due date (i.e., at 1:30 PM).
6. Any late assignment submission must be given to the professor on a **memory stick**.
7. **We do not accept submissions by attaching your assignment solutions to an email sent to the professor or the TAs.**
8. Your program files must be ASCII text files, otherwise they will not be marked.
9. The file for the main class must be named **RecordProcessor.java**, and the class that reads the file must be **RecordReader.java**.

continued on page 2

Question 1 [15 marks: 7.5 for each class] Write a program that prompts the user to enter the name of a text file. If the file could not be opened for *FileNotFoundException*, your program should catch it, notify the user, and ask for another input until the file can be opened successfully. It reads the content of the file, which is formed by no more than 50 lines. Each line is formed by a record where the fields are separated with commas. It counts the number of occurrences of each of the characters A, D, and W in the third field of the record. For each line in the text file, it produces a summary on the number of occurrences of each of A, D, and W. The summary is written in a text file named "output.txt". A line in the output text file has the following template:

$i, \sqcup n_{(i,A)}, \sqcup n_{(i,D)}, \sqcup n_{(i,W)}, \sqcup\sqcup\sqcup\sqcup\sqcup\sqcup i+6, \sqcup n_{(i+6,A)}, \sqcup n_{(i+6,D)}, \sqcup n_{(i+6,W)}$

where $\sqcup$ denotes a space character, and $n_{(i,\alpha)}$ is the number of occurrences of character α in line i. You should be counting the lines starting from 1. If the line indexed with $(i+6)$ does not exist in the file, the number of occurrences is output as follows:

$i, \sqcup n_{(i,A)}, \sqcup n_{(i,D)}, \sqcup n_{(i,W)}, \sqcup\sqcup\sqcup\sqcup\sqcup\sqcup i+6, \sqcup 0, \sqcup 0, \sqcup 0$

Each 6 lines are separated by a line of the following format:

For the above 6 lines, number of A: $\sqcup n_{A}\sqcup$ **** $\sqcup$number of D: $\sqcup n_{D}\sqcup$ **** $\sqcup$number of W: $\sqcup n_{W}$

The file for the main class must be named **RecordProcessor.java**, and the class that reads the file must be **RecordReader.java**. The class *RecordReader* should have a method with the following signature:

```
public static String[] readThirdFieldsFromFile(String inputFileName)
  throws FileNotFoundException, IOException
```

The class *RecordProcessor* should have two methods with the two following signatures:

```
public static void process(String[] thirdFields) throws IOException
```

and

```
public static void main(String args[]) throws IOException
```

Assumptions:

- The input file is not empty
- Each record in the file has at least 4 fields

For example, with user input being "input.txt", and the content of the file input.txt being

continued on page 3

```
t,e,ADW,s,t
t,e,AADDWW,s,t
t,e,A,s,t
t,e,D,s,t
t,e,W,s,t
t,e,W,s,t
t,e,D,s,t
t,e,A,s,t
t,e,WWW,s,t
t,e,TEST,s,t
t,e,WDA,s,t
t,e,WA,s,t
t,e,AD,s,t
t,e,DW,s,t
t,e,ADA,s,t
t,e,ADW,s,t
t,e,AADDWW,s,t
t,e,A,s,t
t,e,D,s,t
t,e,W,s,t
t,e,W,s,t
t,e,D,s,t
t,e,A,s,t
t,e,WWW,s,t
t,e,TEST,s,t
t,e,WDA,s,t
t,e,WA,s,t
t,e,AD,s,t
```

the result in output.txt should be as follows:

```
1,␣1,␣1,␣1,␣␣␣␣␣␣7,␣0,␣1,␣0
2,␣2,␣2,␣2,␣␣␣␣␣␣8,␣1,␣0,␣0
3,␣1,␣0,␣0,␣␣␣␣␣␣9,␣0,␣0,␣3
4,␣0,␣1,␣0,␣␣␣␣␣␣10,␣0,␣0,␣0
5,␣0,␣0,␣1,␣␣␣␣␣␣11,␣1,␣1,␣1
6,␣0,␣0,␣1,␣␣␣␣␣␣12,␣1,␣0,␣1
For␣the␣above␣6␣lines,␣number␣of␣A:␣7␣****␣number␣of␣D:␣6␣****␣number␣of␣W:␣10
13,␣1,␣1,␣0,␣␣␣␣␣␣19,␣0,␣1,␣0
14,␣0,␣1,␣1,␣␣␣␣␣␣20,␣0,␣0,␣1
15,␣2,␣1,␣0,␣␣␣␣␣␣21,␣0,␣0,␣1
16,␣1,␣1,␣1,␣␣␣␣␣␣22,␣0,␣1,␣0
17,␣2,␣2,␣2,␣␣␣␣␣␣23,␣1,␣0,␣0
18,␣1,␣0,␣0,␣␣␣␣␣␣24,␣0,␣0,␣3
For␣the␣above␣6␣lines,␣number␣of␣A:␣8␣****␣number␣of␣D:␣8␣****␣number␣of␣W:␣9
25,␣0,␣0,␣0,␣␣␣␣␣␣31,␣0,␣0,␣0
26,␣1,␣1,␣1,␣␣␣␣␣␣32,␣0,␣0,␣0
27,␣1,␣0,␣1,␣␣␣␣␣␣33,␣0,␣0,␣0
28,␣1,␣1,␣0,␣␣␣␣␣␣34,␣0,␣0,␣0
```

THE END.