CECS 277 HOMEWORK FILE I/O

OBJECTIVE: Give you some firsthand experience working with ASCII text files, handling I/O exceptions and using an ASCII text file as a means of communication between two applications.

INTRODUCTION: Please remember the coding standards here.

Part 1:

A hotel salesperson enters sales into a text file. Prompt the user for the location and name of the output text file. Each line contains the following, separated by semicolons:

- The name of the customer.
- The service sold. Support:
  - Breakfast
  - Lunch
  - Dinner
  - Conference
  - Tea
  - Massage
  - + any others that strike your fancy
- The amount
- The date that the service has been/will be rendered.

You will prompt the user for these inputs, and validate them. Be sure that the user only inputs valid services, that the price is a valid floating point number, and that the date is a valid date. For date validation, look up the class `DateTImeFormatter` and `LocalDate`, and use the format: "MM/dd/uuuu".

Create a class: `Service`, that has all of the information in it for a given row of the output file, and give it a `toString()` method so that you can easily output a given instance of Service to a file.

Part 2:

Prompt the user for the output file created in Part 1. You will use this as your input file. Read each record from the file and **validate the values**. That file could have come from anywhere, do not trust the records to have the proper format. Then, decide which file to output that record to. For instance, if the third record in the input file is: "Dave Brown;Lunch;12.95;09/03/2019", then you would output that entire record to a file named "Lunch.txt" in the same directory with your input file.

Both Parts:

- Be sure to catch and process all exceptions.

- Write your own exception class for any errors that might come up in the sales records, such as an invalid service name, invalid date or invalid price.

PROCEDURE:

1. Write EnterSales.java to prompt the user for their sales records.
2. In the same Java project, create DistributeSales.java that will read the records written out in EnterSales.java and output them to the appropriate output file.
3. Be sure to test your input validation, and demonstrate in your sample output file that you check the user's input.

WHAT TO TURN IN:

- EnterSales.java
- DistributeSales.java
- Your sample output file from Part 1.
- Your sample output file**s** from Part 2.

SAMPLE OUTPUT FILES:

- Sales.txt

```
Micheal Rutz;Massage;150.23;08/22/2019
Dave Brown;Massage;160.23;08/10/2020
Mary Brown;Tea;12.14;08/10/2019
Meisner;Conference;306.23;08/20/2019
Perry Benjamin;Breakfast;25.21;08/25/2019
```

- Breakfast.txt

```
Perry Benjamin;Breakfast;25.21;08/25/2019
```

- Conference.txt

```
Meisner;Conference;306.23;08/20/2019
```

- Massage.txt

```
Micheal Rutz;Massage;150.23;08/22/2019
Dave Brown;Massage;160.23;08/10/2020
```

- Tea.txt

```
Mary Brown;Tea;12.14;08/10/2019
```