JAV745 Fall 2019: Lab 4 - (4%)

Dr. Eden Burton
School of ICT, Seneca College of Applied Arts and Technology
Fall 2019

Due Tuesday October 29, 2019 - 11:30 pm

Instructions

Please read the instructions carefully and follow the naming conventions specified for each question. Solutions must be submitted in the Blackboard Dropbox created for Lab 4.

Note that the deadline is strictly enforced. The system tracks the exact time that submissions are uploaded. **There is a 10% per day penalty for late submissions.**

Additional Notes

- You may use any IDE for development but note that demonstrations and professor testing will be done exclusively on the command line. Ensure that you test your programs on the command line before submission.
- ensure that your programs are documented using *JavaDoc* standards

Question Descriptions

Question 1) In this lab, you will create accounts based on the information obtained from a text file (as opposed to user input from a keyboard). Add a **static** method with the following signature (you may need to slightly modify it).

```
Account[] createAccounts(String inputFile)
```

to the Bank class.

The execution of this method should result an array of account object references, where each row in the file shall create either one or zero objects. (You can assume a maximum of 2000 accounts can be stored). *You must use Java's new NIO package to do the file processing.*

A row should be structured as follows

- account type (exactly 1 character, S - Saving Account, C - Chequing Account)
- customer name (string of arbitrary length)
- account number (an integer value ranging between 7000 and 8999, savings account have a first digit '8', chequing account have a first digit '7')
- balance (optional - a fractional value, rounded to 2 decimal places, if not provided, set to 0)

with the values comma-delimited.

A properly structured row triggers the creation of one account object, initialized with respective values from the row data. It is your job to create an Account class, following design guidelines discussed in the course.

The solution must handle exception events which may be occur while processing a file. Errors such as the following should terminate the program as the system can no longer respond to the request.

- an empty file
- file cannot be opened for some reason (permissions, file not found etc)

Certain errors can be recovered from such as these found below:

- a row with an incorrect number of fields
- a field with incorrect format....(ie a balance with a string value)
- a field with invalid data according to system rules..(no duplicate account numbers are allowed, savings accounts cannot have negative balances.....saving accounts start with 8, chequing accounts 7)

The program must output the following to the System.err stream.

(a) the number of successfully create accounts
(b) for each failed creation, the type of error should be reported

Deliverables

(a) `Bank.java,Account.java` and any other classes implemented
(b) input files used to test your application