Aaron James George

ajgeorge@hotmail.co.uk

Smart Banking Feature

Assumptions, Thoughts and Installation Instructions

**Please see the README file at the GitHub URL provided below for instructions on running the application.**

**GitHub source code:** <https://github.com/ajames191/smart-banking-feature>

# Assumptions

* CSV file names will never begin with 0s
* All file names will follow the pattern “customer-1234567-ledgerEntry.csv”
* All accountType fields in the CSV files will be in uppercase
* As we are not simulating the end of each day, when the current account balances go below 0, that will represent the end of day and balances will be adjusted accordingly
* Ledger transactions are to be output to a separate CSV file and not appended to files already provided

# Thoughts

* Interfaces have been implemented to ensure class validity
* Initiator types could be enumerated to reduce the possibility of errors
  + I didn’t implement this because I wanted to stick to the design
* The BigDecimal class could be implemented for currencies to eliminate the inaccuracies of floats and doubles
  + I didn’t implement this because I wanted to stick to the design
* I could have used an ArrayList of ledger entries instead of an entirely new class called “Ledger” to store entries as this only really adds an additional, unnecessary layer to the application. However, I feel that Ledger should be an object and therefore I have implemented a class for it
* I haven’t included comments in my code because I have attempted to implement the Single Responsibility principle, where classes, methods and variable names are self-explanatory. I learned to do this from working on a hair salon application as a side project with a friend who works as a software engineer. He explained that developers often forget to update comments, leading to them being outdated and unusable. The approach I have taken eliminates maintainability issues caused by outdated comments
* After completing the project code and attempting to run it, I encountered the ConcurrentModificationException. As I had very little time to re-write the code, I had to add some temporary ArrayLists and Lists to get the solution working. In the future, I would attempt a different approach. After some research, I believe the Iterator class may provide a solution