Java Challenge 2025

Overview

Your task is to develop a digital banking system. This banking system should allow users to create a new account, login to an account, deposit/withdraw funds from an account, logout of the account, and display general information regarding the account. It is **not** a requirement that application data persist when the application is closed.

1. Application Interface/Landing Screen

The application should run entirely from the console and be text based. Upon launching the application, the user should be prompted to select an option from a list of available actions. These actions should include signing into an account, or creating a new account. If a user selects to sign into an account, they should then be prompted to enter an account number, and then be shown a list of available actions for their account. If they wish to create a new account, they should be prompted to enter in the required fields to create an account (initial balance, account type, and account holder name), and then be redirected to the same set of options for when a user signs into an account.

2. Account Options/Main Menu

Once a user has successfully accessed/created an account, they should be presented with the following options in the main menu of the application. Once an action has been completed, the user should be redirected back to the main menu, except when logging out. When logging out, the user should be redirected back to the landing screen.

I. Depositing Funds

A user should be able to enter in a valid amount of money (non-negative/non-zero), and then be shown their resulting account balance.

II. Withdrawing Funds

The user should also be able to withdraw a valid amount of funds from their account (non-negative/non-zero/not greater than funds), and then be shown their resulting account balance.

III. Display General Information

Lastly, the user should have the ability to display general information about their account. Displayed information should include the account balance, the name of the account holder, the account type, and the account number.

3. Types of Accounts

There should be two types of accounts that a user can create, those being a checking account or a savings account.

I. Checking

A checking account should have all of the options listed under section 2 with no extra restrictions.

II. Savings

A savings account should also have all of the options listed under section 2, but with a couple of notable restrictions.

a. Minimum Account Balance

A savings account should never be allowed to have an account balance of under $1,000.

b. Withdraw Limit

The user should be limited to withdrawing in amounts of $500.

4. Bonus Features

If time permits it, your team can explore implementing the following additional features:

I. Transfer Funds

By implementing this, a user would have the option to enter in an account number and valid amount of funds to transfer to the target account. Keep in mind what constitutes a valid amount, as well as the restrictions that may exist for the target account.

II. Transaction List

A list of transactions can be maintained during the lifecycle of the application, and the user should be given the option in the main menu to display all of the transactions that have occurred in their account.

III. Data Persistence

The application can use some form of storage and retrieval to allow data to persist between usages of the application.

IV. Pin

In order to make the application more secure, the usage of a pin can be added when logging into an account.

Scoring

The most important consideration will be whether the application works as expected. It is advised that participants focus on the core functionality listed in sections 1-3 before implementing advanced features found in 1-4.

Entries will be further judged based on the following criteria:

- User friendliness and formatting of text displayed via the console
- Coding best practices
    - Readable, easy to understand
    - Proper naming conventions
    - Effective design patterns
    - Helpful documentation
- Error handling. Your application should not crash unexpectedly. It should gracefully handle any errors that may occur during runtime.