Assignment using the Concept of Inheritance - Design

MariconBank Program

The Maricon Bank needs to maintain information on the bank accounts. The bank allows two types of accounts –checking accounts and savings accounts for its customers. Create an application that will allow you to create a new savings or checking account for your customers. It must allow the customer to deposit as well as withdraw from his account. Assume that the user deposits $100 minimum to create a new checking account and $500 minimum for the savings account when a new account is created. You must create an array of at least 10 accounts, each element of which could be either a savings or checking account. Savings account and checking account must be subclasses of the Account class. You need to determine if you need to make the account class an abstract class or not.

In withdrawals you must charge $0.75 for any withdrawal over $2000 for savings accounts and $0.50 for any withdrawal over $750 for checking accounts. You must not let the customer withdraw, if his withdrawal plus any charges incurred would exceed the account balance

**Presentation class:**

* Create an array of objects of the Bank Account class
* Create the components for creating a new account, deposit to an existing account, and withdrawal from an existing account.
* If creating a new account, ensure that the PIN number is unique and not already being used.
* If depositing or withdrawing from an existing account, ensure that the account exists by checking for the PIN number.
* Validate the inputs for creating a new account, depositing to an existing account, and withdrawing from an existing account.
* Get the input from the user for one of the three functions
* Instantiate the Bank Account array with checking or savings for new accounts only and populate the next element of the array when creating a new account.
* Ensure that the user cannot overrun the array by creating too many account elements.
* For deposits or withdrawals, make sure that you are accessing the correct account in your array.
* Update the balance and charges if incurred
* Display the results

**Bank Account class:**

* This is the framework – because all are bank accounts
  + Abstract class
* What information makes them all bank accounts?
  + First Name
  + Last Name
  + Pin Number
  + Balance in the account
* What other information do you need for the bank account?
  + What type of account is this?
  + The current balance
  + Deposits to the account
  + One boolean flag if the withdrawal amount exceeds the balance
  + Remember that all methods used in either the checking or savings account must be included in the account class, although the methods may be abstract
* What methods/tasks needed for all accounts but differs depending on whether checking or savings?
  + Minimum deposit
  + Withdrawal charge
  + Threshold value (when the additional charge is incurred with a withdrawal)
* What kind of modifiers do I need? (public, protected or private)
  + All variables are protected
  + Make the constants protected
* Do you I need mutator and accessor methods?
  + Yes, I will need to input the entries of the bank account through mutator methods
  + The set methods will enter the account data assign to protected variables
  + The get methods will return the needed data back to the driving class.
  + You may use an is method if you need to return a boolean value to the presentation class

**Checking and Savings Account classes:**

* Inherit the Bank Account class
* Access to all the protected variables from the Bank Account class
* Charges that may be incurred if the withdrawal amount is higher than the threshold withdrawal amount
* Create an empty constructor and a constructor containing input from the presentation class
* Override the abstract methods from the account class.
* In the method of withdraw
  + If the amount of the withdrawal is higher than the threshold (withdrawal limit), send a confirmation message (or a flag) back to the presentation class to inform the user that additional charges will be incurred and give them an opportunity to back out of the withdrawal.
  + Check if the withdrawal amount is greater than or equal to the balance.
  + If the withdrawal amount plus any additional charges are greater than the balance – send a message or a flag
    - If the withdrawal amount plus any additional charges are not greater than the balance, update balance