Maggie Mulhall
Foundations of Programming
Final Bank Project

Algorithm

1. Import random and math libraries
2. Define Account class
    a. Initialize an account with the following attributes:
    b. first name, last name, social security number, randomly generated pin, balance, and a randomly generated account number.
    c. Define getters and setters for each attribute
        i. Define a getter for a protected social security number that only displays the last 4 digits
    d. Define a isValidPIN function
        i. Accept an entered pin number, get the pic number on the account and check if they match
    e. Define a deposit method
        i. Accept an entered deposit amount, get the account balance, add the deposit amount, use the setter to set the new account balance
    f. Define a withdraw method
        i. Accept an entered withdrawal amount, get the account balance,subtract the withdrawal amount, use the setter to set the new account balance
    g. Define a to String method
        i. Return a formatted display of all account information on an inputted object
        ii. Use the getter for protected social security number for display
3. Define BankManager class
    a. Initialize an instance of the bank to use as “bank manager”
    b. Define main function
        i. Continuously display options, and call evaluate choices method with users choice
        ii. Exit when the user choose to
    c. Define evaluate choice method with fed in chode from main method
        i. Choice 1: Open account with bank method
        ii. Choice 2: Display account information with:
            1. promptForAccountNumberAndPIN bank manager method
            2. to string account method
        iii. Choice 3: Change pin numberUse:
            1. promptForAccountNumberAndPIN bank manager method
            2. Prompt for/validate new pin
            3. Use account set pin method
        iv. Choice 4: Deposit money
            1. promptForAccountNumberAndPIN bank manager method

2. Prompt for/validate/convert deposit amount
3. Use account deposit method

v. Choice 5: Transfer money
    1. Both both accounts: promptForAccountNumberAndPIN bank manager method
    2. Prompt for/validate/convert transfer amount
    3. Use account withdraw and deposit methods to make transfer
vi. Choice 6: Withdraw money
    1. Use promptForAccountNumberAndPIN bank manager method
    2. Prompt for/validate/convert withdraw amount
    3. Use account withdraw method
vii. Choice 7: ATM withdrawal
    1. Use promptForAccountNumberAndPIN bank manager method
    2. Prompt for/validate withdraw amount
    3. Convert amount to cash
    4. Use account withdraw method
    5. Display results
viii. Choice 8: Deposit change
    1. Initialize instance of coin collector class
    2. Use promptForAccountNumberAndPIN bank manager method
    3. Display legend
    4. Prompt for/validate inputted change
    5. Use coin collector parse change method
    6. Convert amount
    7. Use account deposit method
    8. Display results
ix. Choice 9: Close an account
    1. Use promptForAccountNumberAndPIN bank manager method
    2. Use bank remove account method
    3. Display result
x. Choice 10: Apply interest
    1. Prompt for/validate.convert interest amount
    2. Use banks add monthly interest method
    3. Display balance/interest rates for all accounts
xi. Choice 11: exit
    1. Exit out of main loop to close program

d. Define prompt for account number and pin method
    i. Prompt for account number
    ii. Look for account number in bank accounts array
        1. If found, ask for pin
            a. Use accounts get pin method and check if inputted pin matches
            b. Return account if successful

4. Define the Bank class
    a. Initialize an empty array with 100 spaces
    b. Define a set up account method
        i. Prompt and verify user input for First name, last name, social security number and deposit amount
        ii. Throw an error if invalid input
        iii. Use random number generator for the pin and account number
        iv. Create an instance of the accounts object with new information
        v. Add new account to the account class
        vi. Display account information using to String method
    c. Define add account to bank method
        i. Accept new account
        ii. If there is space in the account array, add new bank account to the array
        iii. Return results
    d. Define remove account from bank
        i. Accept account
        ii. Look for account in accounts array with account number matching in putting account's number
        iii. If found, delete it
        iv. Return results
    e. Define find account method
        i. Accept account number
        ii. Look fro account in accounts array with matching account number
        iii. Return account if found
    f. Define get accounts
        i. Return all accounts in account array
    g. Define monthly interest
        i. Accept annual rate
        ii. Convert annual rate to monthly amount
        iii. Get all account balances, calculate and add interest
        iv. Return accounts array

5. Define Bank Utility class
    a. Define an is numeric method
        i. Accept a string
            1. If the string is a digit, return True,
            2. If the sting is not a digit, return False
    b. Define a convert dollars to cents method
        i. Multiply dollars by 100 to get/return cents
            1. If there's an error, return False
    c. Define a generate random integer method
        i. Use randint from random library (with a min and max for the range)
        ii. Throw an error if
        iii. Return the random number

6. Define Coin Collector class
    a. Define parse change method
        i. Accept a string of coins
        ii. Initialize amount in cents variable to 0
        iii. Continuously loop through items in string of coins variable
            1. Add respective amounts of money to the amount variable when valid letters are found
        iv. Return amount in cents variable
7. Call main bank manager method