Final-W2024

Monday, October 23, 2023 7:00 PM

Final

You are to write a Python program that simulates a very simple ATM machine. An account owner (User) should be able to enter a pin number and select from a menu of transactions: Deposit, Withdraw, Balance, Get Account Information and Quit. You are to assume that the user has only one account
on which these transactions can be performed. The ATM should properly and regularly communicate with the user. The ATM should get
information on current accounts from a file (accounts.txt). This file is posted with the assignment. Each line of the file contains a first name, last name, and the account balance.

1. Create a class called Account which implements the following functions
   a. The constructor should take two parameters: balance and account Number
   b. The default value for balance=0 and the default value for accountNumber = -1. When processing the data read from the file in the constructor, be sure to convert them to the appropriate values. Accounts will later be created by reading information in from a file so they will initially be read in as strings. You will also need to remove any hidden characters, like '\n'.
   c. In the constructor:
      i. Create a class variable accountNumber
         1. If the parameter accountNumber = -1 then set the class variable accountNumber to a randomly generated number between the values of 10000000 and 99999999
         2. If the parameter accountNumber is greater than or equal to 10000000 then the class variable accountNumber is set to the value of the passed parameter
      ii. Create a class variable transLog which will be an empty list
      iii. Create a class variable balance to keep the account balance
         1. Set the value of the class variable balance to the value passed as a parameter
   d. Create the following functions:
      i. getAccountNumber
         1. Returns the account number
      ii. getBalance
         1. Returns the account balance
      iii. getTransactions
         1. Returns the list of transactions
      iv. Deposit
         1. Takes one parameter, a number which will be added to the balance
         2. This action should also be added to the Transaction log as a string describing what the transaction was using addTransaction
      v. Withdraw
         1. Takes one parameter, a number which will be decremented from the balance.
         2. If the amount of the withdraw is greater than the balance then an error message should print out stating that there are insufficient funds and then return with the balance unchanged
         3. If the amount of the withdraw is below the amount of the balance, then deduct the amount from the balance
         4. This action should also be added to the Transaction log as a string describing what the transaction was using addTransaction
      vi. addTransaction
         1. Takes one parameter, a string representing the transaction that just occurred.

2. Create a class User. This class is a subclass of class Account and will implement the following functions: createPin(), getName() and getPin()
   a. The constructor will take three parameters.
      i. The parameter name will be set to a class variable called name. This parameter does not have a default value
      ii. The parameter balance will be passed to the constructor of the super class. This parameter does not have a default value
      iii. The parameter accountNumber will be passed to the constructor of the super class. The default value of this parameter is -1
   b. The constructor will call the class function createPin()
   c. Implement the function createPin()
      i. This function takes no parameters
      ii. It assigns a class variable called pin to a random integer between 1000-9999
   d. Implement the function getName()

i. This function returns the name of the user
e. Implement the function getPin()
i. This function returns the Pin to the caller

3. Create an ATM class with the following functionality
a. Constructor:
i. Create a class variable called accounts and set it to an empty list
ii. Read in the file containing all of the accounts.
iii. With each account in the file, create an instance of the User class and add it to the accounts list
1. Note: You will want to print out the name of the user as well as the PIN after creating it so you can login afterwards
iv. The constructor will then call the authorize function
v. Be sure to catch any exceptions so the application does not crash
b. Authorize():
i. Create a while loop that will break when the number of invalid logins reaches 3
ii. Prompt the user to enter a pin number.
iii. If the pin matches a pin of one of the users, i.e. it belongs to a User associated with an Account, call the displayMenu() function
iv. If the pin is invalid (meaning there is no match in the Account/User list), allow a total of 3 attempts to enter a pin. If the pin is invalid display (‘Invalid pin, try again: ‘) and accept another pin number.
v. After the third unsuccessful attempt, Display the message "You have performed too many invalid logins, Goodbye".
c. DisplayMenu():
i. Display the user’s name and menu options:
1. Deposit
2. Withdraw
3. Balance
4. AccountInfo
5. Quit
ii. Create a while loop that has no breaking condition
1) If the user enters a valid choice, execute the code for that choice
2) If the user enters an integer that is not 1,2,3,4 or 5, then display the message ‘Enter 1 or 2 or 3 or 4 or 5, try again ’
3) If the user enters a non-numeric value, Code an exception Value Error value that prompts the user with the message (‘ x invalid choice, non numeric characters not allowed, try again ‘) where x is the value entered
iii. Stay in the loop until a valid choice is entered
iv. When a valid choice is selected then call the class method that is associated with the choice. If the choice is to exit then set the customer value to None and return
1) Deposit calls the User.Deposit function
2) Withdraw calls the User.Withdraw function
3) Balance calls the User.Balance function
4) Prints out the account number of the User, the Balance and the Users name
5) Quit will exit the application

d. Deposit:
i. Takes one parameter, the authorized User.
ii. Create a while loop, prompt the user for an amount to be deposited. The amount must be converted to float. If the amount is negative, print ‘ Negative amount. Try again’.
iii. Code an exception (‘ Invalid amount. Use digits only.). An interruption would be caused if the user enters a string instead of a numeric value, or hits the enter key without entering anything.
iv. Calculate the new balance and update the balance in the Account. Also add the transaction to the customer Transaction Log
v. Return. There is no need to run the displayMenu function again, the menu should display again once you return from the function

e. Withdraw:
i. Takes one parameter, the authorized User.
ii. Create a while loop, prompt the user for an amount to be withdrawn. The amount must be converted to float. If the amount is negative, print ‘ Negative amount. Try again’.
iii. Code an exception (‘ Invalid amount. Use digits only.). An interruption would be caused if the user enters a string instead of a numeric value, or hits the enter key without entering anything.
iv. If the amount to be withdrawn is greater than the balance print (‘Insufficient funds to

complete the transaction’).

v. Calculate the new balance and update the balance in the customer. Also add the transaction to the customer Transaction Log

vi. Return. There is no need to run the displayMenu function again, the menu should display again once you return from the function

f. Balance:

i. Takes one parameter, the authorized User

ii. Add the transaction to the customer TransactionLog

iii. Prints the current balance

iv. Return. There is no need to run the displayMenu function again, the menu should display again once you return from the function

g. AccountInfo:

i. Takes one parameter, the authorized User

ii. Print out the Account Number

iii. Print out the Account Balance

iv. Print out the name of the User

v. Return. There is no need to run the displayMenu function again, the menu should display again once you return from the function

h. Exit:

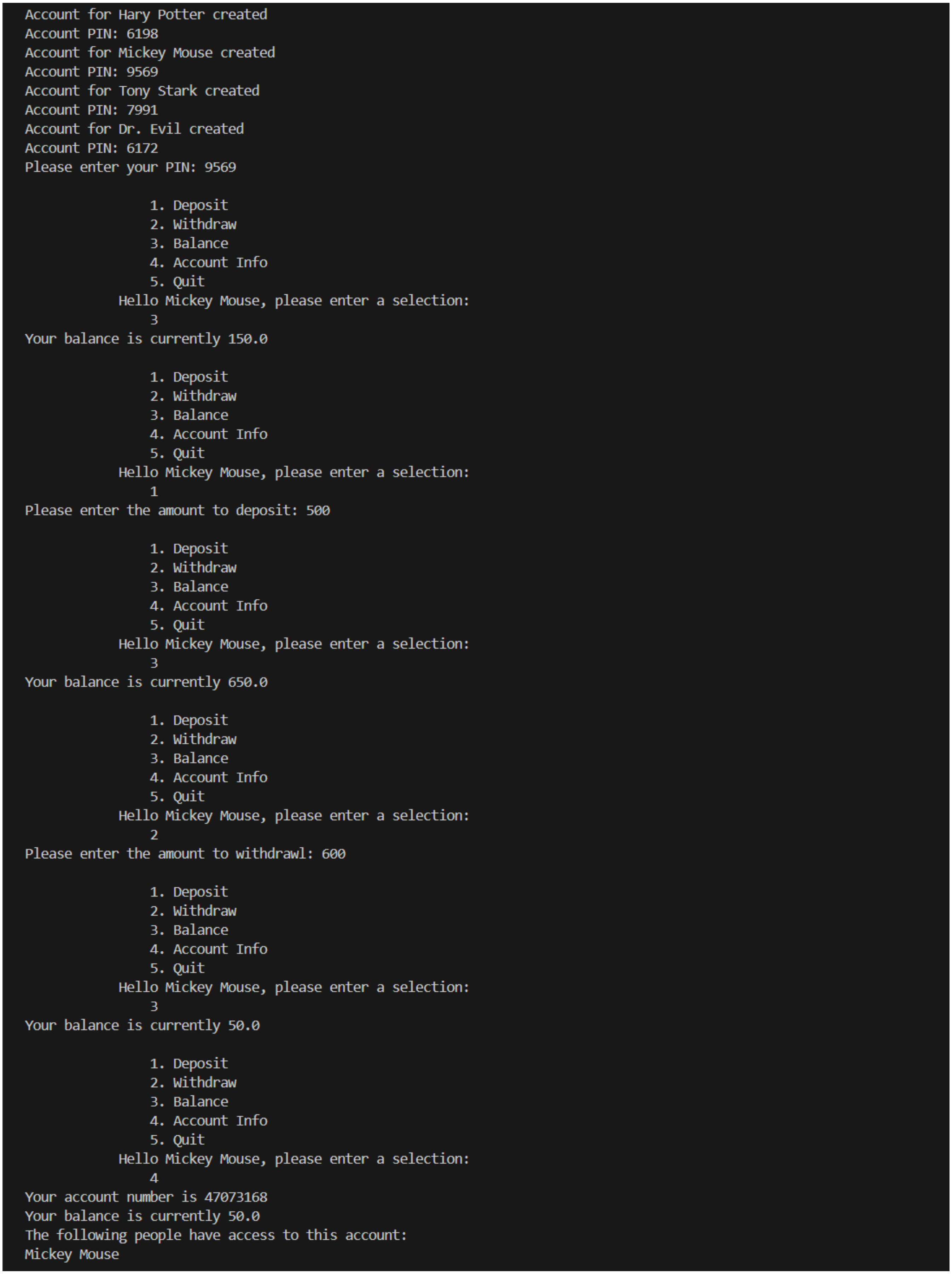
i. Takes one parameter, the authorized User

ii. Prompt the user to confirm they would like to exit with Y or N. This should NOT be case sensitive. If I enter y or n it should be able to match to Y or N

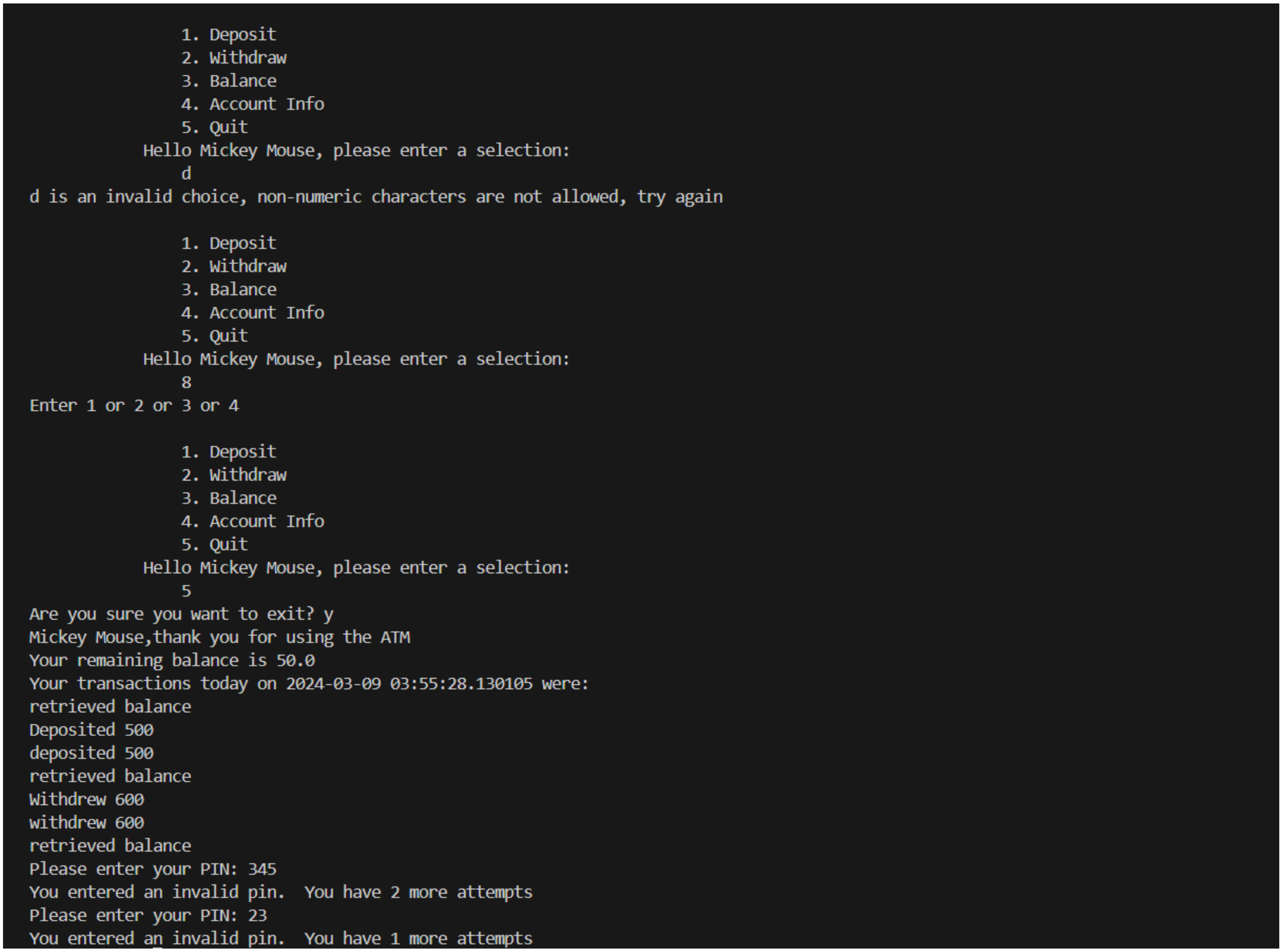
iii. If the user selects N then return to the main menu

iv. If the user selects Y print a receipt to the screen. The receipt should display all of the transactions the User has performed during this session only as well as the date, customer name and remaining balance.

4. Create a function called main() which takes no parameters. Instantiate your ATM class



```
Account for Hary Potter created
Account PIN: 6198
Account for Mickey Mouse created
Account PIN: 9569
Account for Tony Stark created
Account PIN: 7991
Account for Dr. Evil created
Account PIN: 6172
Please enter your PIN: 9569

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                3
Your balance is currently 150.0

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                1
Please enter the amount to deposit: 500

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                3
Your balance is currently 650.0

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                2
Please enter the amount to withdrawl: 600

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                3
Your balance is currently 50.0

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                4
Your account number is 47073168
Your balance is currently 50.0
The following people have access to this account:
Mickey Mouse
```



```
                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                d
d is an invalid choice, non-numeric characters are not allowed, try again

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                8
Enter 1 or 2 or 3 or 4

                1. Deposit
                2. Withdraw
                3. Balance
                4. Account Info
                5. Quit
            Hello Mickey Mouse, please enter a selection:
                5
Are you sure you want to exit? y
Mickey Mouse,thank you for using the ATM
Your remaining balance is 50.0
Your transactions today on 2024-03-09 03:55:28.130105 were:
retrieved balance
Deposited 500
deposited 500
retrieved balance
Withdrew 600
withdrew 600
retrieved balance
Please enter your PIN: 345
You entered an invalid pin.  You have 2 more attempts
Please enter your PIN: 23
You entered an invalid pin.  You have 1 more attempts
```