OOP PYTHON APPLICATION - problem 2

Student: Danciulescu Theodora Ioana
CEN 2.1

Project structure

Module accounts.py : contains all the classes required for my application

Module menues.py: contains the functions for the user interactive menu

Module test.py: contains the test function that is testing all the possible situations in the described application

Module main.py: is the start-up menu that contains the call of the test function, but also can call the application with the interactive menu by simply uncommenting the specified line from this module

Module account.py

Class Bank - takes care of the operations of logging in and signing in

create_user(self, ID, password) : creates an User object with the ID and password given as parameters, and also checks the case when the ID is already used by another User;

login_user(self, ID, password) : checks if the ID and password matches with one of the user accounts from the specified bank;

OBS: Before the call of login_user and create_user the client will specify the bank where he/she wants to create an user account. Also if the user wants to log in to a nonexistent bank, the app will notify this. When signing in, the introduced bank will be created automatically.

Class Account: takes care of the responsibilities of a bank account

Initialisation of its members such as: currency, holder(User type), iban(generated by app), balance, start_datetime(creation time of the account), transactions, max_sum_allowed;

modifiy_balance(self, sum): takes care of inserting and extracting money from an account, it also checks corner and also notify the user in cases like: the bank account reaches the maximum sum allowed, check if there are sufficient funds in the bank account for extracting the sum inserted;

account_report(self): function that prints details about a given account and also all the transactions' history;

Class User: takes care of all the actions a logged in user can perform

open_account(self):creates an Account object and stores data about the new account, also displays the new account's IBAN generated by app;

close_account(self, iban):closes an existing account by eliminating it from the lists that keeps the Account's evidence, also will notify in the case when the user introduces a non existing IBAN code;

insert_sum(self,sum, dest_account_iban):calls function 'modify_balance' from the class Account in order to change the state of an existing account, also will notify in the case when the user introduces a non existing IBAN code;

extract_sum(self,sum, source_acount_iban): calls function 'modify_balance' from the class Account in order to change the state of an existing account, also will notify in the case when the user introduces a non existing IBAN code;

transfer_money(self,source_account_iban, dest_account_iban, sum): function that calls function 'modify_balance' form the class Account in order to change the states of 2 existing accounts, by extracting and inserting money from the 2 accounts;first, extract money from the source account, second, insert money in the source account and also app will notify in the case when the user introduces a non existing IBAN code;

account_report(self, iban_account_to_report):function that calls the 'account_report' function from Account class for an existing account, also the app will notify in the case when the user introduces a non existing IBAN code

Module menues.py

Fisrt_Menu(): function for displaying and interactive menu for SIGN IN and LOG IN operations, also will notify in the case when the ID is already used by another user, in the case when the user wants to LOG IN for a nonexisting bank, in the case when the ID or password doesn't match with any account created at the specified bank;

User_Menu(user_to_query):function for a logged in user, that displays an interactive menu for proceeding the desired transactions, this function is permitting the user to:

- choose to create a new bank account

-choose to close an existing bank account

-choose to insert money in an existing bank account

-choose to withdraw money from an existing bank account

-choose to transfer money between 2 existing bank accounts

-choose to display detailed rapport of an existing bank account

-the possibility to perform another transaction

-the possibility to return to the First Menu

-the possibility to end the session

Module test.py

list input_values: contains the input values for the test function

test_app():test function using the override of input and print function: input function will add in a list named output all the parameters of input in the app, and will return the first element from input_values list, while the print function will append to the same list named output everything that is printed in the app, and in the end will be made an assertion between the list named output and the output considered correct for the input_list. If the tests are passed, a message will be printed in the console

Module main.py

Is the start-up module of the project and calls the test_app function.

It also offers the possibility to experiment the described app experience personally by uncommenting the line specified in the module.