OBJECT ORIENTED PROGRAMMING CA2

BANK SYSTEM PROGRAM

RAGHD ALJUMA
D19125768

Brief explanation of the program:

This program is a bank system that provides several services for the user. The program allows the user to login/ register as a new customer to system and provide the user with the following service:

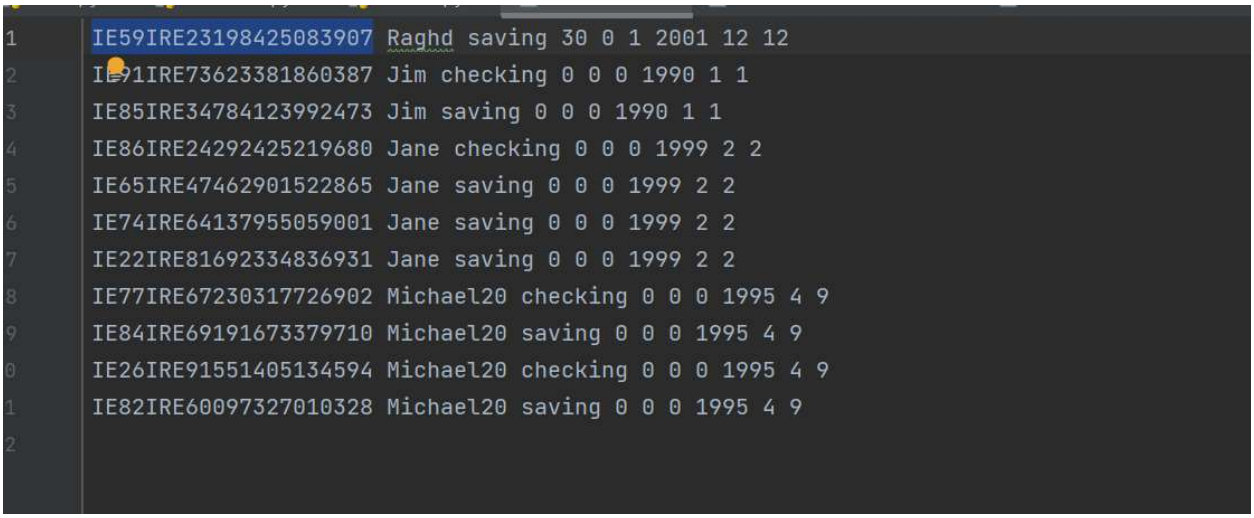
- Create a new account (Checking/Saving account)
- View balance in an account
- Deposit money
- Withdraw money
- Transfer money to a different account
- Print receipt for each transaction
- Print transaction history for an account
- Delete account

The data of the system are all stored in three different files in the same directory and are retrieved and stored in 3 different dictionaries during runtime.

Files:

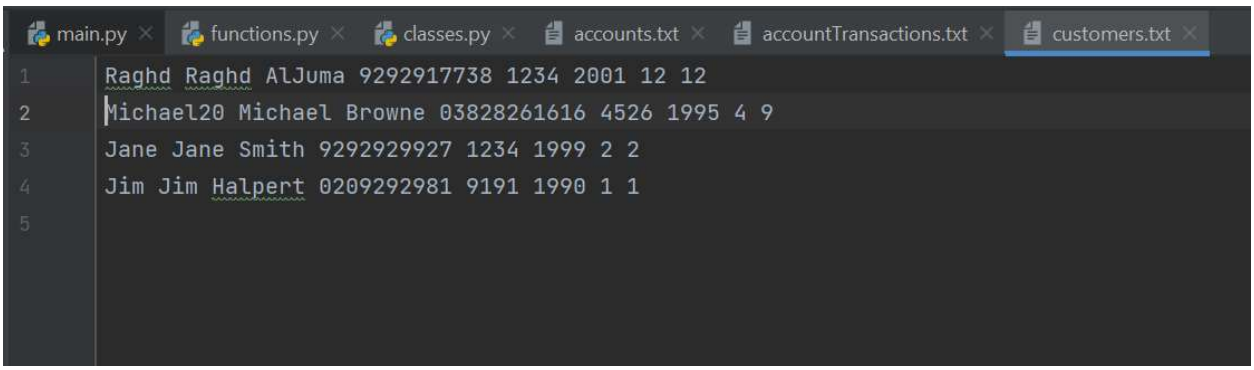
- main.py: Contains the main program
- functions.py: contains extra functions to be used
- classes.py: contains all the classes
- accounts.txt: Contains the data of the accounts. Each account is in a line and its data is separated by a space with the first element being the account's IBAN which is its unique ID

  This is the order: IBAN, customer id, account type, funds, number of withdrawals, number of transfers, year of birth, month of birth, day of birth



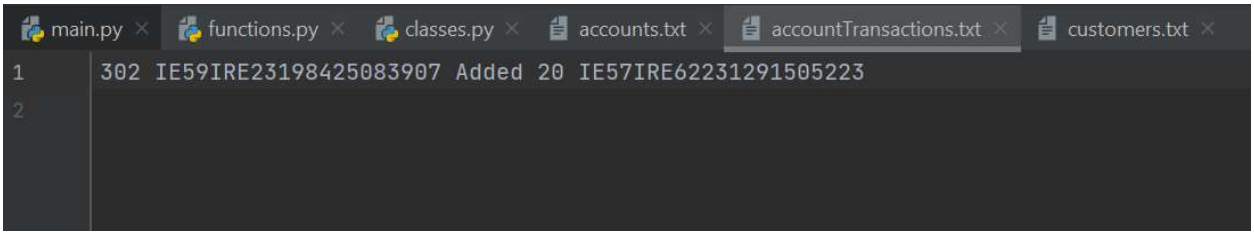
- customers.txt: Contains all the customers data. Each customer has a line with their data separated by a space, and the first element is the user ID.

This is the order: Customer ID, customer first name, customer last name, phone number, pin, year of birth, month of birth and day of birth.



- accountTransaction.txt: Contains all bank transactions. Each transaction has a line with the data separated by a space and the first element is the Transaction ID.

This is the order: transaction id, IBAN, transaction type, amount, Iban (sent from or sent to)



Classes:

1. Customer class:

class Customer(object):

The Customer class allowed creating a new customer. The class had the following attributes:

- Customer id (public)
- First name (public)
- Last name (public)
- Birth Day , month and year (public)
- Phone number (public)
- And pin (private)

The pin was created as private as it is considered sensitive data.

The class also had the following methods:

- __init__
- __str__
- get_pin : to get the private pin from class
- set_pin : to set private pin from class

```
class Customer(object):
    def __init__(self, identification: str, first_name: str, last_name: str, phone_number: str,
                 year: int, month: int, day: int):...

    def get_pin(self) -> str:...

    def set_pin(self, a: str):...

    def __str__(self) -> str:...
```

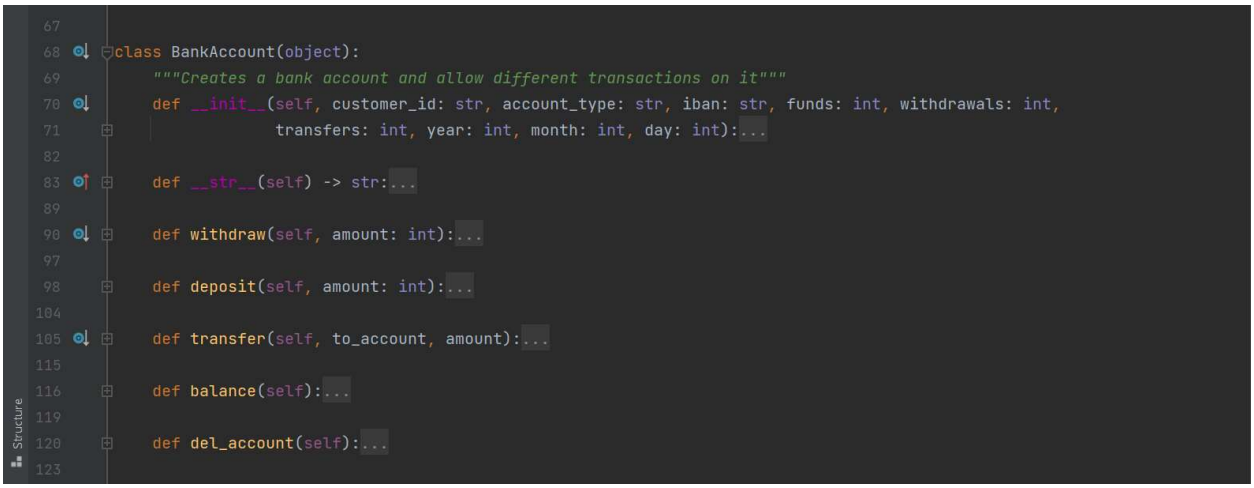
2. BankAccount class

This class allows the creation of Bank accounts. The class takes in the following attributes for account to be created:

- Customer id (public)
- IBAN (public)
- Account type (public)
- Funds (public)
- Birthday, month, and year (public)
- Number of withdrawals (public)
- And number of transfers (public)

Bank account class has the following methods:

- __init__: initialize and create account
- __str__ return string of specific bank account data
- Withdraw: takes in amount as a parameter and withdraw that amount from account funds, also it increments number of withdrawals.
- Deposit: takes in amount parameter and checks if the amount is not a negative number then deposit that amount in the bank account.
- Transfer: transfer will take in the amount parameter and another bank account as a parameter and transfer that money to the account.
- Balance: displays the current balance in the account
- Del_account : will delete the account

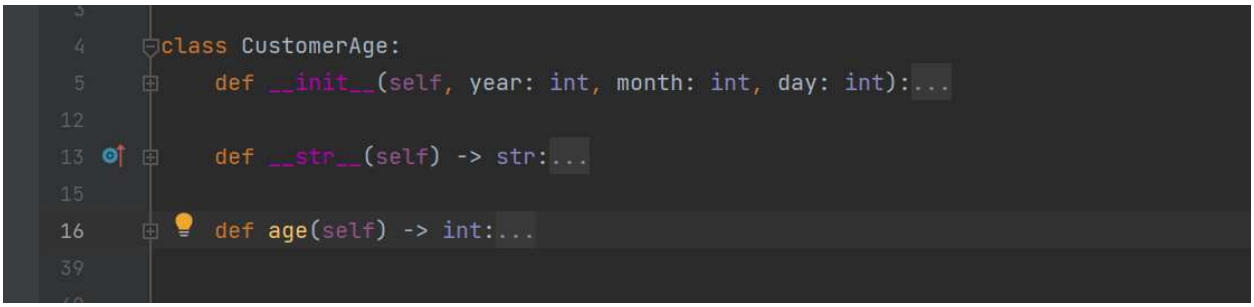


3. **CustomerAge class**

   This class takes in the birth day , month and year and with its age() method calculates the age of the customer. The age method takes the birthday and calculates it from the current day date to find the age and returns it.

   .

   The class contain an __init__, __str__ (returns a string of the birthday) and an age method (returns customer age).



4. **CheckingAccount**

Checking account class is a **child class inheriting** its methods from the BankAccount class.

However, the checking class checks for the user age using the CustomerAge age method using **Composition** to allow only users who are 18 years old and higher.

It also checks before transferring or withdrawing money that the user is not exceeding maximum negative credit with is -1000 in this case.

5. SavingAccount

The SavingAccount is also a child class inheriting all the methods of the BankAccount class. Like the CheckingAccount, the SavingAccount checks the user age and makes sure that the user is 14 years old or higher to create the account. Also, when withdrawing or transferring, the account checks that the user haven't withdrawed or transferred money already, otherwise it will not allow it.

6. AccountTransactions

This class acts like a transaction receipt, and it takes the transaction id, IBAN, transaction type, amount, and the account money was sent to if applicable.

The class has 3 methods:

- __init__
- __str__
- __display__: prints the transaction

Functions:

The program has a file for functions. The file has four functions:

- **Retrieve data**: This function Read the data from the accounts, customers and transactions file and store them in 3 different dictionaries.
- **Login**: Function to allow user to Login to the bank system User is asked to enter their credentials (customer ID and pin number),
  The function checks if the credentials are correct and allow login, otherwise customer needs to re-enter credentials.
- **Update files**: Function to update the 3 files in current directory with the updated dictionaries. The dictionaries are over-written in the files without the brackets and commas.
- **Service Menu**: Function to allow user to choose from services menu and return user's choice.

User Manual:

To use this program, start with logging in with the customer ID and pin. If you are a new customer, you can register as a new customer.

Once you are allowed in, a menu will display containing 3 choices i.e. create an account, check account services, exit program. To choose enter the number associated with the option (1, 2 or 3)

Create New Account:

If you entered 1, the program would ask you for the type of account you would like to create (Saving or checking). Choose the type you would like, and the account will be created for you and you will be informed. You can create as many accounts as you wish. Once you are done creating an account you can user the account services.

Account services:

If you choose account services, all your accounts will be presented to you with your IBAN, if you have more than one account, the program will ask you to enter the IBAN of the account you wish to use.

A menu will be presented with all the services, you will have to enter the number of the service you want to do.

- View balance
- Deposit money to you account: you will be asked to enter the amount. The program will ask you if you wish to get a receipt.
- Transfer money: the program will ask you for the IBAN of the account you wish to send to and the amount. Once transaction is completed the program will ask if you would like a receipt.
- Withdraw: program will ask for amount of money to withdraw. And then receipt.
- View transactions of the account: will print a list of all the transactions.
- Delete account: will delete the current account from the system and will delete the transactions as well.

Once you are done you can exit the program by choosing to exit from the main menu.

Difficulties and challenges:

The most challenging part of this program was figuring out how to story the data in the files in a way that will make retrieving them later with no issues. And when retrieving them, how to store them during runtime. However, I overcame this by storing the data one line each, with each element spaced with a space. So, when I retrieve them, I can just split them. Also, I retrieved them and stored them in a dictionary so that I could iterate through them and find each key. So, in the dictionary the key would be the ID of the customer, account or transaction id and the value associate to that was a list of the rest of the data.