Programação Orientada Objeto

https://dontpad.com/catolicaoop2025

Na aula de hoje dei continuidade no trabalho que foi passado anteriormente, sendo hoje já á segunda semana de prazo dado, como eu já tinha feito toda essa parte da lógica de como seria meu código foquei nessa parte das operações e funções, bom na verdade já tinha feito boa parte em outro dia hoje deixei mas mesmo para focar nas parte que não estava funcionando direito, como a operação 1 estava puxando toda as mudança da conta como alteração de chave pix ou senha da própria conta, sendo que era pra mostra apenas essa parte do balanço mesmo, passei boa tarde também focando nos menus estáticos do próprio shell.

Depois foquei no polimento e alinhamento dos menu, acabei criando 3 menus e um parâmetro para “pular” as linhas colocando uma linha simbólica para mostrar que ele pulou para a próxima tela, os menus que acabei criando foram os menu inicial quando se inicializa o programa esse podemos chamar de PYATM e apenas um logo com escrito bem vindo em inglês é o nome pyatm, o próximo menu foi das informações do usuário assim que ele faz o login no programa mostra o um Olá com seu nome conta é número, é também seu saldo atual, já o próximo e o menu das operações como ver o balançar da conta, fazer um depósito ou uma retirada manda um pix, ou até sair do programa.

O'Que foi feito até o momento segue o código:

```
import time
class Account:
    def __init__(self, password: int, agency: int, number: int,
balance: float = 0.0, keyPix: str = None, limit: float = 1000.0, name:
str = None):
        self.__password = password
        self.__agency = agency
        self.__number = number
        self.__balance = balance
        self.__keyPix = keyPix
        self.__limit = limit
        self.__transactions = []
        self.__name = name
    @property
    def balance(self):
        return self.__balance
    @property
    def agency(self):
        return self.__agency
```

```
    @property
    def number(self):
        return self.__number
    @property
    def keyPix(self):
        return self.__keyPix
    def setKeyPix(self, new_key: str):
        self.__keyPix = new_key
        print(f">> PIX key set to: {new_key}")
        jump
    def removeKeyPix(self):
        if self.__keyPix is None:
            print(">> No PIX key registered to remove.")
        else:
            print(f">> PIX key {self.__keyPix} removed.")
            self.__keyPix = None
        jump
    @property
    def limit(self):
        return self.__limit

    @limit.setter
    def limit(self, new_limit: float):
        self.__limit = new_limit

        self.__transactions.append(f"Limit updated: ${new_limit:.2f}")
        print(f">> Limit successfully updated to ${new_limit:.2f}")
        jump

    @property
    def name(self):
        return self.__name

    def login(self, agency: int, number: int, password: int):
        return self.__agency == agency and self.__number == number and
self.__password == password

    def changePassword(self, new_password: int):
        self.__password = new_password
        print(">> Password successfully updated!")
        jump

    def deposit(self, amount: float):
```

```
        self.__balance += amount
        self.__transactions.append(f"Deposit: +${amount:.2f}")
        print(f">> Deposit of ${amount:.2f} successful. Balance:
${self.__balance:.2f}")
        jump

    def withdraw(self, amount: float):
        if self.__balance - amount < -self.__limit:
            print(">> Withdrawal denied. Limit exceeded!")
            jump
        else:
            self.__balance -= amount
            self.__transactions.append(f"Withdraw: -${amount:.2f}")
            print(f">> Withdrawal of ${amount:.2f} successful. Balance:
${self.__balance:.2f}")
            jump

    def sendPix(self, amount: float, destination: str):
        if self.__balance - amount < -self.__limit:
            print(">> PIX transfer denied. Limit exceeded!")
            jump
        else:
            self.__balance -= amount
            self.__transactions.append(f"PIX sent: -${amount:.2f} to
{destination}")
            print(f">> PIX of ${amount:.2f} sent to {destination}.
Balance: ${self.__balance:.2f}")
            jump

    def statement(self):
        print("=== Account Statement ===")
        if not self.__transactions:
            print("No financial transactions yet.")
        else:
            for t in self.__transactions:
                print("-", t)
        print(f"Current balance: ${self.__balance:.2f}")
        jump
layout = """
========================================
|                                      |
|          WELCOME TO PYATM            |
|                                      |
```

```
========================================
|           [1] Check Balance           |
|           [2] Deposit                 |
|           [3] Withdraw                |
|           [4] Send PIX                |
|           [5] Change Password         |
|           [6] Manage PIX Key          |
|           [7] Adjust Limit            |
|           [8] Exit                    |
========================================
"""
layoutLG = """
========================================
|                                      |
|           WELCOME TO PYATM           |
|                                      |
========================================
"""
jump = "\n################################################\n"
def homeScreen(account: Account):
    print("========================================")
    print("|"f"Olá, {account.name}","                  |")
    print("|"f"Agência: {account.agency}","
|")
    print("|"f"Conta: {account.number}","                           |")
    print("|"f"Saldo atual: ${account.balance:.2f}","
|")
    print("========================================")
    time.sleep(2)
    jump
    atm_interface(account)
def atm_interface(account: Account):
    jump
    while True:
        print(layout)
        option = input("Select an option: ")
        jump
        if option == "1":
            account.statement()
        elif option == "2":
            value = float(input("Enter deposit amount: "))
            jump
            account.deposit(value)
```

```
        elif option == "3":
            value = float(input("Enter withdrawal amount: "))
            jump
            account.withdraw(value)
        elif option == "4":
            value = float(input("Enter PIX amount: "))
            dest = input("Enter destination PIX key: ")
            jump
            account.sendPix(value, dest)
        elif option == "5":
            new_pass = int(input("Enter new password: "))
            jump
            account.changePassword(new_pass)
        elif option == "6":
            print("=== PIX Key Management ===")
            if account.keyPix is None:
                print("No PIX key registered yet.")
            else:
                print(f"Current PIX key: {account.keyPix}")
            print("[1] Register/Change PIX key")
            print("[2] Remove PIX key")
            print("[3] Cancel")
            choice = input("Choose an option: ")
            jump
            if choice == "1":
                new_key = input("Enter new PIX key (email or phone): ")
                jump
                account.setKeyPix(new_key)
            elif choice == "2":
                account.removeKeyPix()
            elif choice == "3":
                print(">> PIX key operation canceled.")
                jump
            else:
                print(">> Invalid option.")
                jump
        elif option == "7":
            new_limit = float(input("Enter new limit: "))
            jump
            account.limit = new_limit
        elif option == "8":
            print(">> Thank you for using PYATM. Goodbye!")
            jump
```

```
            break
        else:
            print(">> Invalid option, try again.")
            jump
        time.sleep(1)
if __name__ == "__main__":
    acc = Account(password=1234, agency=1, number=101, balance=500.0,
name="Victor Emanuel")
    print(layoutLG)
    while True:
        agency = int(input("Agency: "))
        number = int(input("Account number: "))
        password = int(input("Password: "))
        if acc.login(agency, number, password):
            print(">> Login successful!\n")
            jump
            homeScreen(acc)
            break
        else:
            jump
            print(">> Invalid login. Try again.")
            print(layoutLG)
```