SISTEMA BANCARIO EN PYTHON

Alumno: Antonio Ramos Gonzalez

Matricula: 372576

Maestro: Carlos Gallegos

Materia: Paradigmas de la programacion

Fecha de entrega: 30 de mayo de 2024

Introduccion:

El paradigma de Programacion Orientada a Objetos nos permite codificar clases, las cuales estan basadas en objetos reales o en conceptos abstractos. Esta clases actuan como plantillas que definen las propiedades y el comportamiento de los objetos que se estan creando a partir de ellas.

Desarrollo:

Como actividad se nos pidio crear una aplicacion en Python usando POO, la cuan debe de cumplir con los siguientes requisitos.

- Hacer uso de clases
- Hacer uso de objetos
- Hacer uso de abstraccion de datos
- Hacer uso de encapsulamiento
- Hacer uso de herencia
- Hacer uso de polimorfismo

Para la creacion de la app el grupo quedo en hacer un **sistema bancario**, el cual permita realizar **depositos**, **retiros**, **checar estado de cuenta**, y **transferir** a otros usuario del banco.

Para la realizacion de la aplicacion se hizo uso de 3 clases. Clase Person, Clase User (Que hereda atributos de clase Person) y clase BankClass. Tambien se hace uso de harchivos de texto para almacenar las cuentas de los usuarios y poder acceder a ellas aunque el programa haya finalizado

Clase Person

```python
#Antonio Ramos Gonzalez
#372576
#Clase Person
#30/05/2024
class Person:
    def __init__(self,name,age,gender):
        self.name=name
        self.age=age
        self.gender=gender
    def greet(self):
        print("Hola, mi nombre es: ",self.name)
        print(f"Tengo {self.age} anios")
        print("Y mi genero es ", self.gender)

    def farewell(self):
        print("Adios tonoto")
```

Clase User: Hereda los atributos de la clase Person y se encarga del registro e inicio de sesion de los usuarios, leyendo y escribiendo las ceuntas en el archivo un archivo de texto. Para el registro se pide nombre, edad, contraseña, validacion de contraseña y genero. Una vez termine el registro se entrega un numero de cliente, el numero de tarjeta y clave interbancaria, todos los datos son agregados al archivo txt

```python
#Antonio Ramos Gonzalez
#372576
#Clase User
#30/05/2024
from  BankClass import BankClass
from Person import Person
from random import randint

class User(Person):
    def __init__(self):
        super().__init__('','','')
        self.__password=''
        self.__noClient=0
        self.account=BankClass()

    def register(self,name,age,password,valPass,gender):
        self.name=name
        self.age=age
        if int(self.age)<18:
            print("Lo sentimos, para crear una cuenta se debe de contar con una edad minima de 18 años:")
            return
        self.gender=gender
        self.__password=password
        while(True):
            if self.password!=valPass:
                print("Las contraseñas no coincide. Favor de intentarlo nuevamente")
            else:
                break
        print("La cuenta ha sido creada con exito")
        print("Bienvenido al la familia de BANCO ZAPOTECA.")
        self.account._noCard=randint(1000000000000000,9999999999999999)
        self.account._interBanKey='542965'+str(randint(10000000000,99999999999))+str(randint(0,5))
        self.__noClient=randint(10000,99999)
        print("Su numero de cliente es: ",self.__noClient)
        print("Su No. de tarjeta es: ",self.account._noCard)
        print("Su clave interbacaria es: ",self.account._interBanKey)
        print("Su saldo inicial sera de : ",self.account._balance," $")
        fa=open("Bank/Cuentas.txt","a")

        fa.write(str(self.__noClient) + ' ' + self.name + ' ' + self.__password + ' '
                 + str(self.account._noCard) + ' ' + str(self.account._interBanKey) + ' ' + str(self.account._balance) + '\n')
        fa.close()
```

Para el inicio de sesion se solicitan el numero de tarjeta y contraseña, si coninciden con los leidos del archivo de texto entos permite e inicio de sesion.

```python
def Login(self,noCard, password):
    fa=open("Bank/Cuentas.txt","r")
    for line in fa:
        part=line.split()
        if part[2] == password and part[3]==noCard:
            print("Login Realizado con exito")
            self.account._balance=part[5]
            self.account._interBanKey=part[4]
            self.account._noCard=part[3]
            fa.close()
            return True
    fa.close()
    return False
```

Clase BankClass: Esta clase guarda la informacion bancaria del usuario leyendola directamente desde el archivo de texto, permitiendo hacer depositos, retiros, consulta de saldo y transacciones.

```
#Antonio Ramos Gonzalez
#372576
#Clase BankClass
#30/05/2024
class BankClass:
    def __init__(self):
        self._interBankKey=''
        self._noCard=''
        self._balance=0

    def WithdrawCash(self,amount):
            if float(self._balance)>0 and float(self._balance)>=float(amount):
                lines=[]
                with open("Bank/Cuentas.txt", "r") as fa:
                        for line in fa:
                            part=line.split()
                            if part[3] == self._noCard:
                                self._balance=float(part[5])-amount
                                part[5] = str(self._balance)
                            newLine=' '. join(part) + '\n'
                            lines.append(newLine)
                with open("Bank/Cuentas.txt","w") as fa:
                    fa.writelines(lines)
                print("El retiro se ha realizado con exito")
            else:
                print("Saldo insuficiente. Favor de ingresar otra cantidad")


    def DepositCash(self,amount):
        lines=[]
        with open("Bank/Cuentas.txt", "r") as fa:
                for line in fa:
                    part=line.split()
                    if part[3] == self._noCard:
                        self._balance=float(part[5])+amount
                        part[5] = str(self._balance)
                    newLine=' '. join(part) + '\n'
                    lines.append(newLine)
        with open("Bank/Cuentas.txt","w") as fa:
            fa.writelines(lines)

    def Transaction(self,amount,InterBankKey):
        lines=[]
        with open("Bank/Cuentas.txt", "r") as fa:
                for line in fa:
                    part=line.split()
                    if part[3] == self._noCard:
                        self._balance=float(part[5])-amount
                        part[5] = str(self._balance)

                    if part[4] == InterBankKey:
                        tempB = float(part[5])+amount
                        part[5] = str(tempB)

                    newLine=' '. join(part) + '\n'
                    lines.append(newLine)
        with open("Bank/Cuentas.txt","w") as fa:
            fa.writelines(lines)

    def CheckBalance(self):
        print("No.Cuenta: ",self._interBankKey)
        print("No.Tarjeta: ",self._noCard)
        print("Saldo en cuenta: ",self._balance)
```

Archivo main: El archivo main es donde se controla la aplicacion, desde ahi podremos registrarnos e iniciar sesion. Una vez iniciada la sesion podremos realizar las operaciones de depositar, trasnferir, retirar y consulta de saldo, asi como cerrar sesion

```python
#Antonio Ramos Gonzalez
#372576
#Archivo principal
#30/05/2024
from User import User
import os
import platform
system=platform.system()
def clear():
    if system == "Linux":
        os.system("clear")
    if system == "Windows":
        os.system("cls")
def pause():
    input("Oprima cualquier tecla para continuar")



def menuGeneral(opc):
    clear()
    if opc==1:
        print("registro banco ZAPOTECA")
        name=input("Nombre: ")
        age=input("Edad: ")
        password=input("Contrasenia: ")
        valPass=input("Validar contrasenia")
        gender=input("Genero: ")
        client=User()
        client.register(name,age,password,valPass,gender)
    elif opc==2:
        while(True):
            print("Login")
            noCard=input("Ingrese numero de tarjeta: ")
            password=input("Ingrese contrasenia: ")
            client=User()
            val=client.Login(noCard,password)

            if(val):
                menuLogin(client)
                return
            else:
                print("Los datos no coinciden. Favor de intentarlo nuevamente")
```

```python
def menuLogin(client):
    while(True):
        clear()
        print("1.-Depositar")
        print("2.-Retirar")
        print("3.-Transferir")
        print("4.-Mostrar informacion cuenta")
        print("5.-Cerrar Sesion")
        opc2=int(input("Elija una opcion: "))
        clear()
        if opc2 == 1:
            amount=float(input("Ingrese cantidad a depositar: "))
            client.account.DepositCash(amount)
```

```
        elif opc2 ==2:
            amount=float(input("Ingrese cantidad a retirar: "))
            client.account.WithdrawCash(amount)
        elif opc2 ==3:
            amount=float(input("Ingrese cantidad a transferir: "))
            InterBankey=input("Ingrese clave interbancaria: ")
            client.account.Transaction(amount,InterBankey)
        elif opc2 ==4:
            client.account.CheckBalance()
            pause()
        elif opc2 ==5:
            print("Hasta luego ",client.name)
            return

clear()
print("1.-Registrarse")
print("2.-login")
opc=int(input("Elija una opcion: "))
menuGeneral(opc)
```

Ejecutable Dentro de la carpeta *dist* existe un main.exe para ejecutar la aplicacion sin nececidad de compilar

Conclusion:

Esta actividad me ayudo a comprender y fortalecer mis conocimientos sobre el paradigma de programacion orientado a objetos, aplicando los distintos principios del mismo como lo son el encapsulamiento, herencia, creacion de clases y objetos, entre otros.