Practica 2. Atributos públicos y privados

Castillo Dominguez Sonia Vanessa

19/09/25

Introducción

En esta práctica se trabajó con la programación orientada a objetos (POO) en Python. El objetivo principal fue crear clases que representen entidades de la vida real, en este caso una Persona y una CuentaBancaria, con atributos y métodos propios. A través de esta práctica se buscó comprender cómo funcionan los atributos privados, el concepto de encapsulamiento y la interacción entre objetos mediante instancias.

Desarrollo de la práctica

#Clase Persona

La clase Persona fue diseñada para almacenar información básica de un individuo como nombre, apellido y edad. Además, incluye un atributo privado ___cuenta que representa una cuenta bancaria asociada a la persona.

Clase CuentaBancaria

La clase CuentaBancaria representa una cuenta con un número de cuenta y un saldo inicial. El saldo se declaró como un atributo privado (___saldo) para proteger la información financiera.

Codigo python

class Persona:

```
def __init__(self, nombre, apellido, edad):
    # creación de atributos
    self.nombre = nombre
    self.apellido = apellido
    self.edad = edad
    self.__cuenta = None  # cuenta privada

def asignar_Cuenta(self, cuenta):
    self.__cuenta = cuenta
    print(f"{self.nombre} ahora tiene una cuenta bancaria")

def consultar_saldo(self):
    if self.__cuenta:
        print(f"El saldo de {self.nombre} es: $ {self.__cuenta.mostrar_saldo()}")
    else:
        print(f"{self.nombre} no tiene cuenta creada")

def presentarse(self):
    print(f"Hola, mi nombre es {self.nombre}, mi apellido es {self.apellido}, y tengo {self.e

def cumplir_anos(self):
    self.edad += 1
    print(f"{self.nombre} cumplió {self.edad} años")
```

class CuentaBancaria: def **init**(self, num_cuenta, saldo): self.num_cuenta = num_cuenta self.___saldo = saldo # atributo privado

```
def mostrar_saldo(self):
    return self.__saldo

def depositar(self, cantidad):
    if cantidad > 0:
        self.__saldo += cantidad
        print(f"Se depositó la cantidad de ${cantidad} a la cuenta")
    else:
        print("Ingresa una cantidad válida")

def retirar(self, cantidad):
    if 0 < cantidad <= self.__saldo:
        self.__saldo -= cantidad
        print(f"Se retiró la cantidad de ${cantidad}")
    else:
        print("Fondos insuficientes o cantidad inválida")
```

creación de objetos o instancias de la clase

estudiante1 = Persona(“Jose”, “Angel”, 19) estudiante2 = Persona(“Angel”, “Martin”, 18)

Presentaciones

estudiante1.presentarse() estudiante2.presentarse()

Cumpleaños

estudiante1.cumplir_anos()

Crear cuentas bancarias

cuenta1 = CuentaBancaria(12345, 500) cuenta2 = CuentaBancaria(67890, 1000)

Asignar cuentas

estudiante1.asignar_Cuenta(cuenta1) estudiante2.asignar_Cuenta(cuenta2)

Consultar saldo

estudiante1.consultar_saldo() estudiante2.consultar_saldo()

Operaciones con la cuenta de Miguel

cuenta1.depositar(200) cuenta1.retirar(100) estudiante1.consultar_saldo()