Universidad Autónoma del Estado de México
UAEM

A1 Ejercicio de encapsulamiento

Licenciatura en Ingeniería en Sistemas Computacionales

Periodo Educativo 2025-/

ICO 29

Paradigmas de

Programación

Docente: M. EN T. DE INF. JULIO ALBERTO DE LA TEJA LÓPEZ

Alumno: Ariel Morales Bernardino

Centro Universitario UAEMex Atlacomulco Agosto del 2025

Licenciatura de
Ingeniería en Computación

Resuelve el siguiente ejercicio.
Ejercicio:
Crea una clase llamada CuentaBancaria que represente una cuenta bancaria con los siguientes atributos privados:

- titular: el nombre del titular de la cuenta.
- saldo: el saldo de la cuenta.

La clase debe tener los siguientes métodos públicos:

1. **depositar(cantidad)**: Permite agregar una cantidad al saldo de la cuenta. No permite depósitos negativos.
2. **retirar(cantidad)**: Permite retirar una cantidad del saldo de la cuenta. No permite retiros mayores al saldo disponible ni retiros negativos.
3. **consultar_saldo()**: Devuelve el saldo actual de la cuenta.
4. **consultar_titular()**: Devuelve el nombre del titular de la cuenta.

Reglas adicionales:

- Si se intenta hacer un depósito o retiro con una cantidad negativa, se debe imprimir un mensaje de error y no se debe modificar el saldo.
- Si se intenta retirar más dinero del disponible, se debe imprimir un mensaje de error y no se debe modificar el saldo.

Desarrollo del código:

A1_encapsulamiento.py > CuentaBancaria

```python
class CuentaBancaria:
    def __init__(self, titular, saldo=0):
        self.__titular = titular
        self.__saldo = saldo

    def depositar(self, cantidad):
        if cantidad > 0:
            self.__saldo += cantidad
            print(f"Depósito exitoso. Nuevo saldo: {self.__saldo}")
        else:
            print("Error: No se pueden depositar cantidades negativas o nulas.")

    def retirar(self, cantidad):
        if cantidad <= 0:
            print("Error: No se pueden retirar cantidades negativas o nulas.")
        elif cantidad > self.__saldo:
            print("Error: Fondos insuficientes.")
        else:
            self.__saldo -= cantidad
            print(f"Retiro exitoso. Nuevo saldo: {self.__saldo}")

    def consultar_saldo(self):
        return self.__saldo

    def consultar_titular(self):
        return self.__titular
```

```python
if __name__ == "__main__":
    nombre = input("Ingrese el nombre del titular: ")
    saldo_inicial = float(input("Ingrese el saldo inicial: "))

    cuenta = CuentaBancaria(nombre, saldo_inicial)

    while True:
        print("\n--- Menú ---")
        print("1. Consultar saldo")
        print("2. Depositar")
        print("3. Retirar")
        print("4. Consultar titular")
        print("5. Salir")

        opcion = input("Seleccione una opción: ")

        if opcion == "1":
            print("Saldo actual:", cuenta.consultar_saldo())
        elif opcion == "2":
            cantidad = float(input("Ingrese cantidad a depositar: "))
            cuenta.depositar(cantidad)
        elif opcion == "3":
            cantidad = float(input("Ingrese cantidad a retirar: "))
            cuenta.retirar(cantidad)
        elif opcion == "4":
            print("Titular:", cuenta.consultar_titular())
        elif opcion == "5":
            print("Saliendo...")
            break
        else:
            print("Opción inválida.")
```

Pantalla de salida:

```
PS C:\Users\Ariel-Morales\Desktop\ejercicio2\PAF
Python/Python312/python.exe c:/Users/Ariel-Moral
Ingrese el nombre del titular: Ariel
Ingrese el saldo inicial: 4300

--- Menú ---
1. Consultar saldo
2. Depositar
3. Retirar
4. Consultar titular
5. Salir
Seleccione una opción: 2
Ingrese cantidad a depositar: 100
Depósito exitoso. Nuevo saldo: 4400.0
```

Rubrica Ejercicios de Programación

Nombre del docente: M. EN T. DE INF. JULIO ALBERTO DE LA TEJA LÓPEZ				
Nombre del estudiante: ARIEL MORALES BERNARDINO				
Indicador		**Nivel de desempeño**		
	Alto (.5)	**Medio (.3)**	**Bajo (.1)**	**Valor asignado**
Funcionalidad	La solución es completamente funcional y produce resultados precisos en todos los casos.	La solución produce resultados correctos en la mayoría de los casos, con algunos errores menores.	La solución tiene errores importantes y no produce los resultados esperados.	5
Claridad	El código es claro, con comentarios detallados que enriquecen la comprensión.	El código está comentado y utiliza nombres descriptivos, lo que facilita la comprensión.	El código es confuso y difícil de entender debido a la falta de comentarios y nombres descriptivos.	5
Creatividad	La solución es creativa, innovadora y muestra una perspectiva única.	La solución muestra un intento de creatividad, pero no es totalmente original.	La solución es una copia directa de ejemplos previos o carece de enfoque creativo.	5
Entrega de reporte	El reporte se entrega con estructura, Introducción, desarrollo (código), capturas de salida y ejemplos de funcionamiento, comentarios.	El reporte solo contiene tres de los cuatro apartados elementos.	El reporte no tiene la estructura especificada.	5