



47 lines (31 loc) · 2.42 KB

Preview

Code

Blame

Raw



Sistema Bancario en Python con POO

Información del Programa

Nombre: Teresa Rivas Gómez

Matrícula: 372565

Materia: Paradigmas de la Programación

Programa: Sistema Bancario

Fecha de Inicio: 19/05/2024

Fecha de Final: 23/05/2024

Descripción del Programa

El programa "Sistema Bancario" simula operaciones básicas de una cuenta bancaria. Este programa permite a los usuarios registrar cuentas, iniciar sesión, y realizar diversas transacciones como depósitos, retiros y transferencias. Además, proporciona funcionalidades para mostrar el saldo actual de las cuentas.

Funciones Implementadas

1. MENU INICIO (Registrar o Iniciar Sesión) Ingresar Datos (Nombre y número de cuenta):

- La función `menu_inicio()` presenta el menú inicial, permite registrar una nueva cuenta o iniciar sesión en una cuenta existente.

2. Transacciones (Transferir, Retirar, Depositar) con número de cuenta y cantidad:

- Las transacciones se gestionan mediante las funciones `depositar`, `retirar` y `transferir` dentro de la clase `Banco`.

3. Mostrar Saldo (Nombre, número de cuenta y saldo actual):

- La función `mostrar` en la clase `Banco` permite visualizar esta información.

4. Realizar otra operación antes de salir:

- El menú de transacciones dentro de la función `transaccion` permite a los usuarios realizar múltiples operaciones y salir del sistema cuando lo deseen.

Implementación de la Programación Orientada a Objetos (POO)

El programa cumple con los requisitos proporcionados de esta forma:

1. Clases y Objetos:

- Se utilizan las clases `Cuenta` y `Banco`. Los objetos de estas clases representan cuentas bancarias individuales.

2. Abstracción:

- La abstracción se logra encapsulando los detalles de la implementación, como el saldo de la cuenta, y exponiendo solo las funcionalidades necesarias mediante métodos públicos.

3. Encapsulamiento:

- Los atributos `_nombre`, `_numero` y `_saldo` están encapsulados dentro de las clases y se accede a ellos a través de propiedades y métodos públicos.

4. Herencia:

- La clase `Banco` hereda de la clase `Cuenta`, reutilizando así los atributos `nombre` y `numero`.

5. Polimorfismo:

- Se ejemplifica el polimorfismo mediante el uso de métodos como `transaccion`, que pueden manejar diferentes tipos de operaciones (mostrar, depositar, retirar, transferir) de manera flexible y dinámica.