

# Tofol Cano

## Índice:

1. Portada	pág 1
2. Índice	pág 1
3. Explicación detallada de las clases	pág 2
4. Explicación de los tests	pág 3
5. Comentarios	pág 3

Clase Banco.

Contiene un constructor para los objetos Banco y los getters y setters para sus parámetros, los cuales contienen datos como el nombre y el identificador del banco.

Clase Cliente.

Contiene un constructor para los objetos Cliente y los getters y setters para sus parámetros, los cuales contienen datos como el nombre, apellidos, dni, dirección y el teléfono del cliente

Clase Cuenta.

Contiene un constructor para los objetos Cuenta y los getters y setters para sus parámetros y un getter para el ID. Los parámetros contienen datos como el ID, saldo, el cliente, y el banco. En el constructor se añade el ID del banco delante del ID del cliente pasado por parámetro.

Clase Transferencia.

Contiene un constructor privado para los objetos Transferencia y los getters y setters para sus parámetros y el método transferir. El método transferir crea un objeto Transferencia y comprueba que la transferencia se pueda hacer, si se puede hacer modifica el saldo de las dos cuentas y guarda true en la variable resultado, si no se puede hacer modifica la cantidad a 0 y guarda false en la variable resultado. Los parámetros contienen datos como la cuenta de origen, la cuenta destino, la cantidad de dinero y si se ha realizado con éxito.

Clase Traspaso.

Contiene un constructor privado para los objetos Traspaso y los getters y setters para sus parámetros y el método traspasar. El método traspasar crea un objeto Traspaso y comprueba que el traspaso se pueda hacer, si se puede hacer modifica el saldo de las dos cuentas y guarda true en la variable resultado, si no se puede hacer modifica la cantidad a 0 y guarda false en la variable resultado. Los parámetros contienen datos como la cuenta de origen, la cuenta de destino, la cantidad de dinero y si se ha realizado con éxito.

Tests:

La clase Tests contiene varios clientes, bancos y cuentas con datos para hacer pruebas.

El test `comparaClientesIgualesDeDosCuentas()` comprueba si dos clientes son iguales.

El test `compruebaSaldoOrigenDespuesDeTransferenciaCorrecta()` comprueba si en una transferencia que cumple las condiciones establecidas se le resta al saldo de la cuenta origen la cantidad transferida.

El test `compruebaSaldoDestinoDespuesDeTransferenciaCorrecta()` comprueba si en una transferencia que cumple las condiciones establecidas se le suma al saldo de la cuenta destino la cantidad transferida.

El test `conSaldoSuficiente()` comprueba si en la variable resultado se guarda true en una transferencia que se ha realizado con éxito.

El test `conSaldoInsuficiente()` comprueba si en la variable resultado se guarda false en una transferencia que no se ha realizado con éxito.

El test `traspasoAOtraCuentaDelMismoClienteYBanco()` comprueba si en la variable resultado se guarda true en un traspaso que se ha realizado con éxito.

Comentarios.

No he visto la forma de guardar IDs de bancos como int si podían empezar con dos ceros como en el ejemplo. En vez de eso los he guardado como String y de paso he dejado espacios cada 4 números.