Enunciado: clase CuentaBancaria

Se requiere un programa que modele una cuenta bancaria que posee los siguientes atributos:

▶ Nombres del titular.

▶ Apellidos del titular.

▶ Número de la cuenta bancaria.

▶ Tipo de cuenta: puede ser una cuenta de ahorros o una cuenta corriente.

▶ Saldo de la cuenta.

Se debe definir un constructor que inicialice los atributos de la clase. Cuando se crea una cuenta bancaria, su saldo inicial tiene un valor de cero.

En una determinada cuenta bancaria se puede:

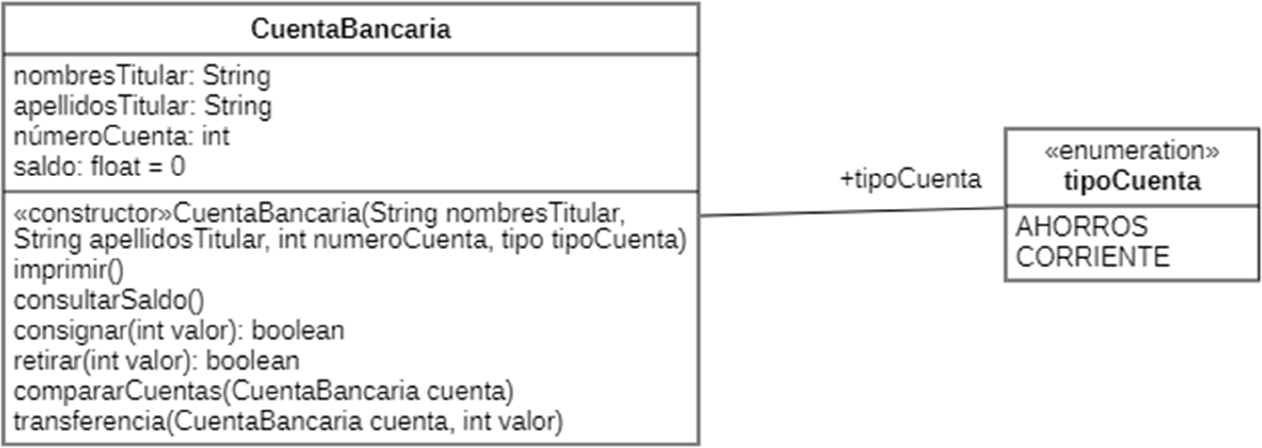
▶ Imprimir por pantalla los valores de los atributos de una cuenta bancaria.

▶ Consultar el saldo de una cuenta bancaria.

▶ Actualizar saldo con un determinado valor en la cuenta bancaria.

▶ Retirar un determinado valor de la cuenta bancaria, actualizando el saldo correspondiente. Es necesario tener en cuenta que no se puede realizar el retiro si el valor solicitado supera el saldo actual de la cuenta.

Diagrama de clases

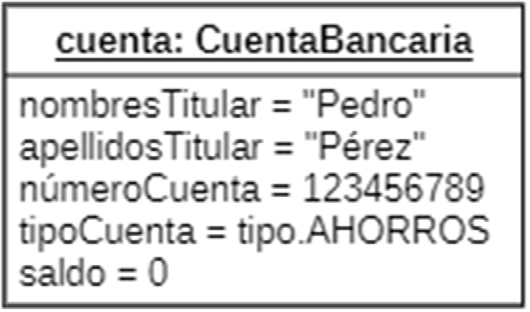


Explicación del diagrama de clases

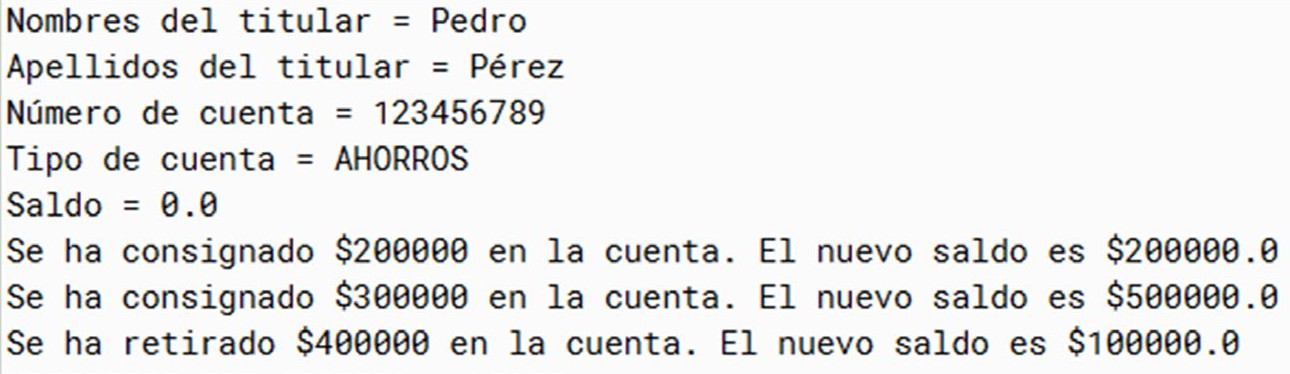
Se ha definido una clase para modelar una cuenta bancaria. La clase cuenta con los atributos que representan los nombres del titular (de tipo *String*), los apellidos del titular (de tipo *String*), su número de cuenta (de tipo *int*) y el saldo de la cuenta con un valor inicial de cero (de tipo *float*). Se han definido los métodos de la cuenta bancaria: su constructor, un método para imprimir los datos en pantalla y consultar saldo (ambos no retornan ningún valor) y consignar y retirar que tienen un parámetro (el valor a con- signar y retirar respectivamente). Los métodos consignar y retirar devuelven un valor booleano que determina si se pudo realizar la operación o no.

La clase CuentaBancaria tiene un atributo denominado tipoCuenta, que es un valor enumerado. En UML, el valor enumerado se puede representar como una clase con el estereotipo <<enumeration>> con sus valores constantes localizados como atributos en la clase.

Diagrama de objetos



Ejecución del programa



Ejercicios propuestos

▶ Agregar a la clase CuentaBancaria, un atributo que represente el porcentaje de interés mensual aplicado a la cuenta.

▶ Agregar un método que calcule un nuevo saldo aplicando la tasa de interés correspondiente.

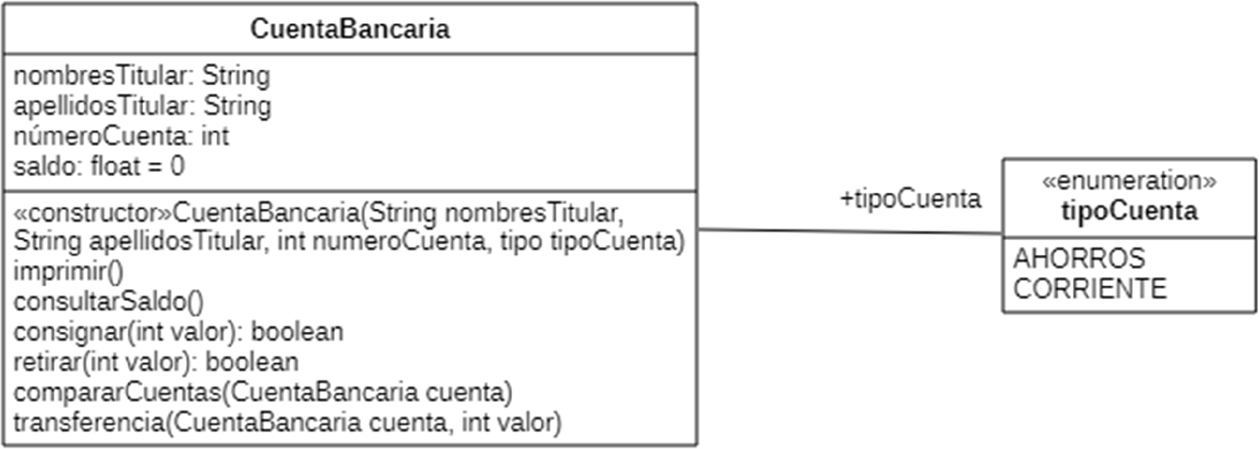
Modificaciones sobre clase CuentaBancaria

Se requiere modificar el programa de la cuenta bancaria para que realice las siguientes actividades:

▶ Comparar saldos entre cuentas bancarias. La cuenta para comparar es un objeto que se envía como parámetro del método. El método devuelve un valor booleano de verdadero si la cuenta actual es mayor o igual a la cuenta que se pasó como parámetro.

▶ Transferir dinero de una cuenta bancaria a otra. El método debe reescribir como parámetro la cuenta de destino y el valor a transferir. El saldo de la cuenta actual debe disminuir el valor a transferir y el saldo de la cuenta destino debe aumentar. El método debe reutilizar el método retirar para evaluar si la cantidad a transferir se encuentra en la cuenta de origen.

Diagrama de clases



Explicación del diagrama de clases

Comparando este diagrama con el del ejercicio anterior, se han agregado dos nuevos métodos a la clase CuentaBancaria:

▶ Comparar cuentas, que tiene como parámetro un objeto de tipo CuentaBancaria.

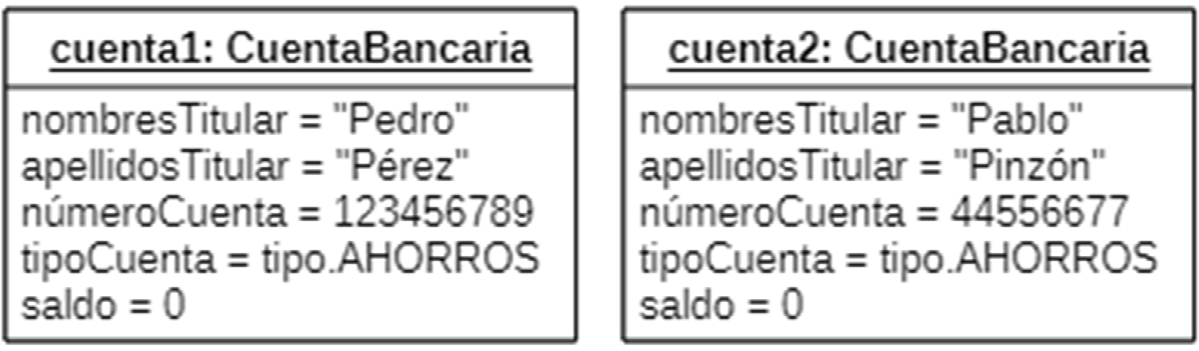
▶ Transferencia, cuyos parámetros son un objeto de tipo CuentaBan- caria y el valor a transferir a dicha cuenta.

Ambos métodos no tienen valor de retorno. La clase CuentaBancaria tiene un atributo denominado tipoCuenta, un valor enumerado. En UML, el

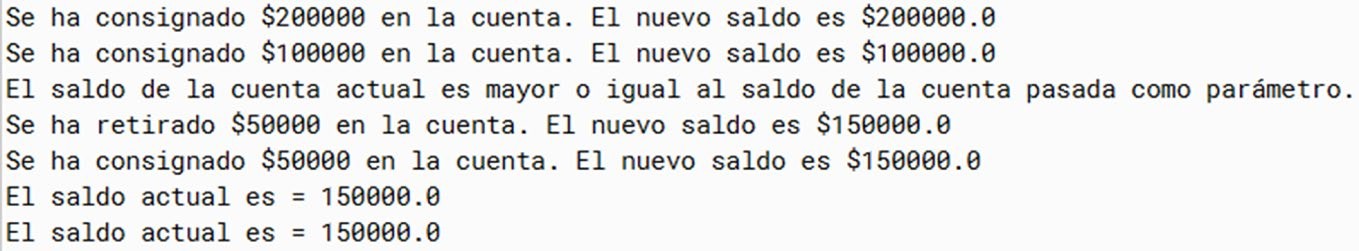
valor enumerado se puede representar como una clase con el estereotipo

<<enumeration>> con sus valores constantes localizados como atributos en la clase.

Diagrama de objetos



Ejecución del programa



Ejercicios propuestos

▶ Agregar un atributo a la clase CuentaBancaria, que determine si la cuenta está activa (de tipo *boolean*). Una cuenta está activa si tiene un saldo positivo. No se pueden realizar consignaciones a la cuenta si está inactiva. Si al retirar dinero, el saldo de la cuenta es cero, la cuenta pasa a considerarse inactiva.