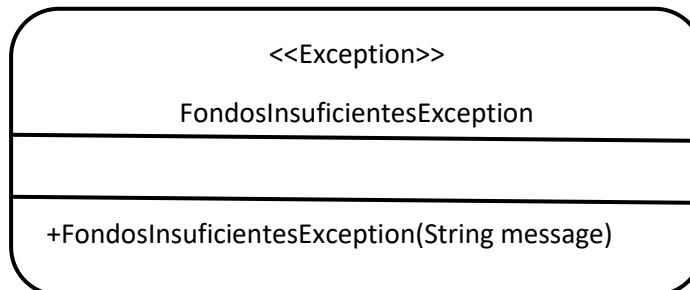
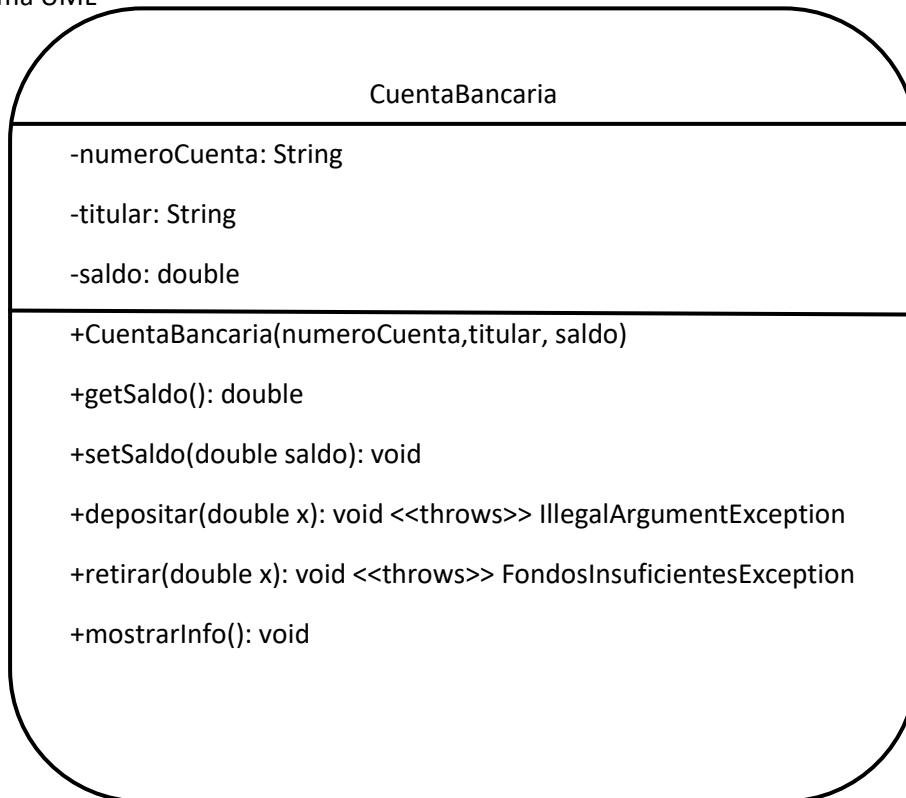


**Ejercicio 6.** Sea la clase CuentaBancaria con los siguientes atributos: numeroCuenta (String), titular (String), saldo (double)

- a) Cree la excepción personalizada **FondosInsuficientesException**, que se lance cuando se intente retirar un monto mayor al saldo disponible.
- b) Implemente los métodos:
  - depositar(double monto) → aumenta el saldo si el monto es positivo, si no, lanza una excepción `IllegalArgumentException`.
  - retirar(double monto) → descuenta el saldo si hay fondos suficientes, de lo contrario lanza `FondosInsuficientesException`.
  - mostrarInfo() → imprime en pantalla número de cuenta, titular y saldo.
- c) Cree un programa principal que:
  - Genere una cuenta con número "12345", titular "Juan Pérez", y saldo inicial de 1000.
  - Realice un depósito válido y otro con monto negativo para disparar la excepción.
  - Realice un retiro válido y otro que supere el saldo para disparar la excepción personalizada.
  - Use bloques try-catch para capturar y mostrar los errores.

Diagrama UML



## Prueba de corrida

```
C:\Users\LENOVO\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrain
-----
Informacion de la cuenta bancaria
-----
Numero de cuenta: 12345
Titular: Juan Perez
Saldo: 1000.0 bs
-----
Se deposito exitosamente la cantidad de: 427.78 bs, el saldo actual es: 1427.78
Error en el deposito... el monto a depositar de -302.0 bs, es negativo
-----
Se retiro exitosamente la cantidad de 671.36 bs, el saldo actual es: 756.42
Error en el retiro... fondos insuficientes para retirar la cantidad de 1364.9 bs
-----
Informacion de la cuenta bancaria
-----
Numero de cuenta: 12345
Titular: Juan Perez
Saldo: 756.42 bs

Process finished with exit code 0
```