

Un banco contiene las Cuentas de sus clientes. Las cuentas de ahorro no pueden tener números rojos. Las cuentas corrientes pueden tener una cuenta de ahorro asociada, de forma que si se intenta retirar más dinero del saldo actual, se debe retirar el dinero que falte de la cuenta de ahorro asociada.

1. Define Cuenta de forma que no pueda instanciarse. De toda Cuenta se debe poder ingresar y retirar dinero, preguntar por el saldo, por el DNI del titular y debe tener un método toString que devuelve al menos el saldo y el DNI del titular.
2. Implementa las clases CuentaCorriente y CuentaDeAhorro.
3. Crea una especialización CuentaDeAhorroEsp de CuentaDeAhorro en la que se añade un entero penalización, de forma que se penaliza la retirada con una penalización % del dinero retirado. Sobrescribe sólo los métodos necesarios (incluyendo constructor y toString).
4. Si el saldo de CuentaDeAhorro fuese a quedar negativo, antes se debe lanzar una excepción SaldoNegativo (que hereda de Exception).

5. Implementa la clase Banco que contiene una colección de Cuentas, incluyendo el constructor que consideres más apropiado.
6. Incluye el método totalSaldoMaxPenalización en la clase Banco que devuelva la suma de los saldos de todas las cuentas corrientes y la máxima penalización entre las CuentaDeAhorroEsp .
7. Crea una clase con método main en la que instanciamos un Banco de 100 Cuentas, nos creamos una CuentaCorriente con 5000 pesos y una cuenta de ahorro especial con una penalización del 5%. Finalmente debe mostrar información de todas las Cuentas del Banco. En un bloque try & catch retirar dinero de una cuenta de Ahorro.