



Cuentas Bancarias (Simulacro de primer parcial)

Un banco contiene las Cuentas de sus clientes.

Las **cuentas de ahorro** no pueden tener números rojos. Además del saldo, tienen una reserva donde el titular reserva dinero. Ese pozo de reserva no se informa como parte del saldo disponible, pero de allí se puede obtener dinero y pasarlo al saldo.

Las **cuentas corrientes** pueden girar en descubierto hasta un monto que se define al momento de su creación.

1. Define **Cuenta** de forma que no pueda instanciarse.
2. 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 devuelva al menos el saldo y el DNI del titular.
3. Implementar las clases `CuentaCorriente` y `CuentaDeAhorro` como una especialización de `Cuenta`.
4. De una cuenta se debe poder transferir dinero a otra cuenta (si tiene dinero disponible).
5. Implementa la clase `Banco` que contiene una colección de Cuentas, incluye los métodos:

```
public Cuenta abrirCuentaCorriente(int dni, double descubierto){}
public Cuenta abrirCajaDeAhorro(int dni){}
public double totalSaldoEnDescubierto(){ } // devuelve la suma de los saldos
de todas las cuentas corrientes que están en descubierto,
public void listarCuentas(){ } // que muestra los datos de todas las cuentas
del banco, ordenado por saldo de manera ascendente.
```
6. Implementar casos de prueba JUnit para cada funcionalidad implementada entre cuentas.

Ayudin:

JUnit 5

```
Error thrown = assertThrows(Error.class,() -> [codigo que genera el
Error]);
```

JUnit4

```
@Test(expected=Error.class)
```