



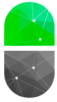
EJERCICIOS POO

Ejercicio 1:

Crea una clase capaz de crear cuentas corrientes bancarias con las siguientes características:

- Nombre de clase: **CuentaCorriente**.
- Propiedades:
 - saldo (double).
 - nombreTitular (String).
 - numeroCuenta (long).
- Constructor con dos parámetros:
 - nombreTitular (String)
 - saldo (double)
 - Este constructor se encargará de establecer un nombre de titular para la cuenta corriente, un saldo inicial y asignará un nº a la cuenta corriente de tipo long de forma aleatoria
- Métodos:
 - 2 setter: permitirán hacer ingresos y reintegros en un objeto de tipo CuentaCorriente.
 - 2 getter: Permitirán obtener saldo de la cuenta y datos generales de una CuentaCorriente.
 - 1 método adicional: permitirá realizar transferencias de dinero de una cuenta a otra.

Crea una clase principal con el nombre de **UsoCuenta**. Crea dos instancias de CuentaCorriente con el nombre de Cuenta1 y Cuenta2. Haz una transferencia de Cuenta1 a Cuenta2 por importe de 200 €. Imprime los datos de las dos cuentas en consola.



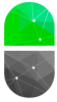
SOLUCIÓN

Una de las posibles soluciones:

- **Clase CuentaCorriente**

```
class CuentaCorriente{  
    public CuentaCorriente(String nombreTitular, double saldo){  
        this.saldo=saldo;  
        this.nombreTitular=nombreTitular;  
        Random rnd=new Random();  
        numeroCuenta=Math.abs(rnd.nextLong());  
    }  
    public void setIngreso(double ingreso) {  
        if (ingreso<0) System.out.println("No se permiten ingresos negativos");  
        else saldo+=ingreso;  
    }  
    public void setReintegro(double reintegro){  
        saldo-=reintegro;  
    }  
    public String getSaldo(){  
        return "El saldo de la cuenta es: " + saldo;  
    }  
    public static void Transferencia(CuentaCorriente titu1, CuentaCorriente titu2, double cantidad){  
        titu1.saldo-=cantidad;  
        titu2.saldo+=cantidad;  
    }  
    public String getDatosCuenta(){  
        return "Titular: " + nombreTitular + "\n" +  
            "Nº de Cuenta: " + numeroCuenta + "\n" +  
            "Saldo: " + saldo;  
    }  
    private double saldo;  
    private String nombreTitular;  
    private long numeroCuenta;  
}
```

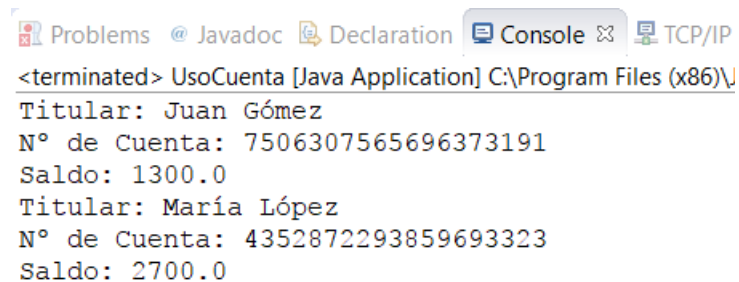
Clase principal y
Resultado en consola



- Clase principal. Creación de instancias. Transferencia e impresión de datos

```
public class UsoCuenta {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        CuentaCorriente Cuentas1=new CuentaCorriente("Juan Gómez", 1500);  
        CuentaCorriente Cuentas2=new CuentaCorriente("María López" , 2500);  
        CuentaCorriente.Transferencia(Cuentas1, Cuentas2, 200);  
        System.out.println(Cuentas1.getDatosCuenta());  
        System.out.println(Cuentas2.getDatosCuenta());  
    }  
}
```

- Resultado en consola



The screenshot shows the IDE's console window with the following output:

```
<terminated> UsoCuenta [Java Application] C:\Program Files (x86)\J  
Titular: Juan Gómez  
Nº de Cuenta: 7506307565696373191  
Saldo: 1300.0  
Titular: María López  
Nº de Cuenta: 4352872293859693323  
Saldo: 2700.0
```