

## Ejercicio 1 - Clase Película

Crear una clase Pelicula con:  
Atributos privados:

titulo (String)  
director (String)  
duracion (int) - en minutos  
vista (boolean)

Constructores:

Constructor completo: titulo, director, duracion, vista  
Constructor sin estado: titulo, director, duracion (vista = false por defecto)  
Constructor básico: solo titulo y director (duracion = 0, vista = false)

Métodos:

Todos los getters y setters  
marcarComoVista() - cambia vista a true  
toString() - muestra toda la información

Validaciones:

Duración no puede ser negativa  
Título no puede estar vacío

En el main crear:

Una película con el constructor completo  
Una película con el constructor sin estado (me refiero a si fue vista o no)  
Una película con el constructor básico  
Mostrar todas las películas  
Marcar una como vista  
Mostrar el resultado final

## Ejercicio 2 - Clase CuentaBancaria

Crear una clase CuentaBancaria con:  
Atributos privados:

titular (String)  
numeroCuenta (String)  
saldo (double)  
activa (boolean)

Constructores (mínimo 3):

Constructor completo con todos los datos  
Constructor cuenta nueva: titular, numeroCuenta (saldo = 0, activa = true)  
Constructor básico: solo titular (numeroCuenta = "SIN-ASIGNAR", saldo = 0, activa = false)

Métodos requeridos:

Todos los getters y setters apropiados  
depositar(double monto) - suma al saldo, SOLO si cuenta activa  
extraer(double monto) - resta del saldo, VALIDAR saldo suficiente Y CUENTA ACTIVA  
activarCuenta() - cambia activa a true

desactivarCuenta() - cambia activa a false, SOLO si saldo = 0  
esCuentaVip() - retorna true si saldo > 10000 (imaginemos que son USD)  
getEstadoCuenta() - retorna String con resumen completo (o sea si esta activa o no  
y si es VIP o no)  
toString() - información clara y completa

Validaciones importantes:

Monto de depósito debe ser positivo  
No se puede extraer más del saldo disponible  
Solo se puede operar con cuentas activas  
Titular no puede estar vacío  
Solo se puede desactivar cuenta si saldo = 0

Crear en el main:

Al menos 3 cuentas usando diferentes constructores  
Realizar varias operaciones (depósitos, extracciones)  
Probar activar/desactivar cuentas  
Mostrar estado final de todas las cuentas  
Probar el método esCuentaVip()