

Ejercicio 1: Clase Coche

Enunciado:

Crea una clase Coche que tenga los siguientes atributos:

- **Atributo público:** marca (String)
- **Atributos privados:** modelo (String), año (int), precio (double)

Requisitos:

1. Implementa los **getters** y **setters** para los atributos privados con las siguientes validaciones:
 - El año debe ser mayor que 1886 (el año en que se inventó el automóvil).
 - El precio debe ser positivo.
2. Sobrecarga el método **toString()** para mostrar la información completa del coche.
3. Crea un constructor que inicialice todos los atributos.

Ejercicio 2: Clase Empleado

Enunciado:

Crea una clase Empleado con los siguientes atributos:

- **Atributo público:** nombre (String)
- **Atributos privados:** salario (double), edad (int)

Requisitos:

1. Implementa los **getters** y **setters** con las siguientes validaciones:
 - El salario debe ser mayor que 0.
 - La edad debe estar entre 18 y 65.
2. Sobrecarga el método **toString()** para mostrar la información del empleado.
3. Crea un constructor que inicialice todos los atributos.

Ejercicio 3: Clase Producto

Enunciado:

Crea una clase Producto con los siguientes atributos:

- **Atributo público:** nombre (String)
- **Atributos privados:** precio (double), stock (int)

Requisitos:

1. Implementa los **getters** y **setters** con las siguientes validaciones:
 - El precio debe ser mayor que 0.
 - El stock debe ser igual o mayor que 0.
2. Sobrecarga el método **toString()** para mostrar la información del producto.
3. Crea un constructor que inicialice todos los atributos.

Ejercicio 4: Clase CuentaBancaria

Enunciado:

Crea una clase CuentaBancaria con los siguientes atributos:

- **Atributo público:** titular (String)
- **Atributos privados:** saldo (double), numeroCuenta (String)

Requisitos:

1. Implementa los **getters** y **setters** con las siguientes validaciones:
 - El saldo debe ser mayor o igual a 0.
2. Implementa los métodos **depositar** y **retirar**, que modifican el saldo según validaciones (no se puede retirar más del saldo disponible).
3. Sobrecarga el método **toString()** para mostrar la información de la cuenta.
4. Crea un constructor que inicialice todos los atributos.