

Desafíos de POO

Nota. Resuelve en orden dado que se enumeran de acuerdo a la complejidad.

1. **Persona.** crear una clase llamada Persona. Sus atributos son: nombre, edad y DNI. Construye los siguientes métodos para la clase:
 - a. Un constructor, donde los datos pueden estar vacíos.
 - b. Los setters y getters para cada uno de los atributos. Hay que validar las entradas de datos.
 - c. mostrar(): Muestra los datos de la persona.
 - d. esMayorDeEdad(): Devuelve un valor lógico indicando si es mayor de edad.

Además, crea en una aplicación de consola que permita al usuario crear un objeto Persona y evaluar si es mayor o menor de edad..

2. **Área y Perímetro.** Crea una clase Rectángulo que permita calcular su área y su perímetro. Además, crea en una aplicación de consola que permita al usuario crear un objeto Perímetro y realizar los cálculos. .
3. **Producto.** Crea una clase Producto con los siguientes atributos: Nombre, Precio y Stock siendo obligatorios sólo los atributos Nombre y Precio. El Stock debe comenzar con 0. Define métodos para actualizar el stock, para calcular el total del inventario y para ver los datos. Además, escribe una aplicación de consola que permita al usuario: 1- actualizar el stock y 2- calcular inventario hasta que decida detenerse. Al finalizar deberá mostrar los datos del producto, stock e inventario final.
4. **Cuenta Bancaria.** Crea una clase CuentaBancaria con los siguientes atributos obligatorios: número de cuenta, nombre y apellido. El saldo debe comenzar con 100.000. Agrega e implementa métodos para: 1- depositar (debe aceptar un valor entero y sumarlo al saldo), 2- retirar (debe aceptar un valor entero y restarlo al saldo sólo si hay dinero suficiente para retirar en la cuenta) y 3- ver saldo. Además, escribe una aplicación de consola que permita al usuario depositar, retirar o ver saldo hasta que decida detenerse. Al finalizar deberá mostrar los datos de la cuenta y el saldo.
5. **Kilometraje Recorrido.** Crea una clase vehículo que permita llevar el kilometraje recorrido. Agrega atributos para definir color, marca, modelo y patente del vehículo y métodos para: 1- conducir el auto (debe aceptar la cantidad de kilómetros y sumarlo al kilometraje recorrido) y 2- muestre en pantalla los datos del vehículo y el correspondiente kilometraje. Además, escribe una aplicación de consola que permita

al usuario conducir varios kilómetros y mostrar el kilometraje actual hasta que decida detenerse. Al finalizar deberá mostrar los datos del vehículo y el kilometraje recorrido.

6. **Control de Volúmen.** Crea una clase ControlVolumen que permita establecer el volumen de un parlante (1 mínimo volumen, 10 máximo volumen). El constructor establecerá el volumen en un nivel medio. Agrega métodos para: 1- ajustar el volumen permitiendo que el mismo sume o reste sin salirse de los límites y 2- mostrar el nivel de volúmen actual. Además, escribe una aplicación de consola que permita al usuario manipular y consultar el volumen hasta que decida salir. Al finalizar deberá mostrar el nivel actual de volumen.
7. **Expocoches.** Escribe un programa de consola para gestionar la venta de entradas (no numeradas) de Expocoches que tiene 3 zonas, la sala principal con 1000 entradas disponibles, la zona de compra-venta con 200 entradas disponibles y la zona vip con 25 entradas disponibles. Hay que controlar que existen entradas antes de venderlas. Define la clase Zona con sus atributos y métodos correspondientes y crea un programa de consola que permita vender las entradas.
El menú del programa debe ser el siguiente: 1- Mostrar número de entradas libres, 2- Vender entradas y 3- Salir

Cuando elegimos la opción 2, se nos debe preguntar para qué zona queremos las entradas y cuántas queremos. Lógicamente, el programa debe controlar que no se puedan vender más entradas de la cuenta.

8. **Compra de Productos.** Escribe un programa de consola para gestionar la compra de productos. Cada producto se define por: id (autoincremental), nombre, caducidad y precio que estará sujeto a la fecha de caducidad como sigue:
 - Si le quedan entre 3 y 5 días, se reduce un 40%
 - Si le quedan menos de 3 días, se reduce un 70%
 - No se podrán vender productos vencidos.
9. **Registro de Estudiantes.** Escribe un programa que permita registrar estudiantes en un arreglo. Cada estudiante debe tener un nombre, una edad y una lista de asignaturas en las que está inscrito. Implementa una clase Estudiante con los atributos mencionados y métodos para agregar asignaturas, mostrar la información del estudiante y calcular su promedio de calificaciones.
10. **Gestión de Biblioteca.** Escribe un programa de consola que simula la gestión de una biblioteca. Cada libro debe tener un título, un autor y un estado (prestado o disponible). Crea una clase Libro con los atributos mencionados y métodos para

prestar y devolver libros. Implementa una clase Biblioteca que almacena una lista de libros y permite buscar libros por título o autor, así como mostrar el estado de un libro específico.