

Taller de Programación Orientada a Objetos en JavaScript

Ejercicio 1: Definición de Clases y Objetos

Objetivo: Crear una clase básica y un objeto.

- Define una clase `Persona` con los atributos `nombre` y `edad`.
- Crea un objeto `personal` de la clase `Persona` e inicializa los atributos con valores.
- Imprime los valores de los atributos en la consola.

Ejercicio 2: Métodos en Clases

Objetivo: Agregar métodos a las clases.

- Extiende la clase `Persona` para que incluya un método `saludar()` que imprima un saludo personalizado en la consola.
- Invoca el método `saludar()` desde el objeto `personal`.

Ejercicio 3: Encapsulamiento

Objetivo: Aplicar encapsulamiento a los atributos.

- Modifica la clase `Persona` para que los atributos `nombre` y `edad` sean privados.
- Implementa métodos públicos `getNombre()`, `setNombre()`, `getEdad()`, y `setEdad()` para acceder y modificar estos atributos.
- Usa estos métodos para modificar e imprimir los atributos de `personal`.

Ejercicio 4: Herencia

Objetivo: Implementar herencia entre clases.

- Crea una clase `Estudiante` que herede de `Persona`.
- Agrega un atributo adicional `grado` y un método `estudiar()` que imprima el mensaje “Estoy estudiando”.
- Crea un objeto `estudiante1` y demuestra la funcionalidad de herencia y los métodos específicos de `Estudiante`.

Ejercicio 5: Polimorfismo

Objetivo: Aplicar polimorfismo.

- Añade un método `describir()` en `Persona` que imprima una descripción genérica.
- Sobrescribe el método `describir()` en `Estudiante` para que incluya la información de `grado`.
- Llama al método `describir()` en un objeto de tipo `Persona` y otro de tipo `Estudiante`.

Ejercicio 6: Constructores Múltiples (Simulación)

Objetivo: Simular constructores múltiples.

- Modifica la clase `Libro` (como el ejemplo anterior) para que acepte un título y autor opcionalmente, o use valores por defecto.
- Crea instancias de `Libro` utilizando diferentes combinaciones de parámetros y muestra los detalles.

Ejercicio 7: Relaciones de Composición

Objetivo: Implementar una relación de composición.

- Define una clase `Biblioteca` que contenga una lista de objetos `Libro`.
- Implementa métodos para añadir libros a la biblioteca y mostrar la lista de libros.
- Crea varios objetos `Libro`, añádelos a la biblioteca y muestra la lista en la consola.

Ejercicio 8: Abstracción

Objetivo: Implementar abstracción.

- Crea una clase abstracta `FiguraGeometrica` con un método abstracto `calcularArea()`.
- Implementa clases `Cuadrado` y `Circulo` que extiendan `FiguraGeometrica` e implementen `calcularArea()`.
- Crea objetos de estas clases y calcula sus áreas.