

UD2E5. Básicos POO

Ejercicio 1.

A partir de la clase Persona, con la siguiente estructura:

- Campos: “nombre”, “nacimiento” (fecha) y “hobbies” (lista de cadenas).
- Propiedades: “edad” (get), computada a partir de nacimiento.
- Métodos: “Saludar”, devuelve la cadena “Hola, me llamo NOMBRE y me gusta LISTA_HOBBIES”.

Se pide:

Define y prueba la clase “Persona” según los 3 formatos de definición de clases soportados por JS: objeto literal, función constructora y ES6.

Ejercicio 2.

Primera parte

Crea una **función constructora** “Animal” que reciba como parámetros el tipo y el nombre.

Contamos con 3 tipos de animal: “animal” (genérico), “perro” y “gato”.

La clase contará con 3 métodos:

- “comer”, devuelve la cadena “nombre está comiendo”.
- “dormir”, devuelve la cadena “nombre está durmiendo”.
- “hacerRuido”, devuelve una cadena distinta según el tipo, por ejemplo, para el tipo perro “nombre hace guau”.

Codifícalo de la manera más correcta posible.

Segunda parte

Vuelve a codificar la clase Animal empleando herencia con el **formato ES6**.

Opcional (aunque recomendado)

Vuelve a codificar la primera parte empleando herencia en funciones constructoras, te recuerdo que hay una propiedad especial llamada “__proto__” (no confundir con “prototype”).

Añade el código necesario para probar el código desarrollado, y no seas cutre borrando una parte para desarrollar la siguiente (crea varios ficheros .js).

Ejercicio 3.

La estructura básica de una “Factura” se compone de:

Campos:

- “clienteNIF”.
- “fecha”, fecha sin la hora.
- “hora”, fecha sólo con la hora y minutos.
- “pagada”, booleano.
- “líneas” (lista), colección de detalles de la factura.

Propiedades:

- “importeTotal” (get), devuelve el importe agregado de las líneas de factura.
- “numeroArticulos” (get), devuelve el número de líneas de factura.

Métodos:

- “imprimirFactura()”, devuelve una cadena formateada con la información de la factura, cabecera con los datos básicos y la lista con las líneas de la factura.
- “agregarLinea(concepto, cantidad, precio)”, añade una nueva línea al final de “líneas”.
- “eliminarLinea()”, elimina la última línea de “líneas”.

La estructura “Línea” de factura se compone de:

Campos:

- “concepto”
- “cantidad”, entero.
- “precioUnitario”, real.

Clase “Utilidades”, se compone de:

Métodos:

- “serializarFactura(facturaOBJ)”, estático, recibe un objeto de tipo Factura y devuelve la cadena JSON serializada.
- “deserializarFactura(facturaJSON)”, estático, recibe una cadena JSON y devuelve el objeto Factura equivalente.

Se pide:

Crea un formulario en el que poder probar lo anterior, su estructura será la siguiente.

Un formulario para introducir “clienteNIF”, “fecha”, “hora” y “pagada”, con un botón “Actualizar” que guarde el valor de los campos en la factura.

Un formulario para gestionar las líneas de factura, contará con los campos “concepto”, “cantidad” y “precio”, y los botones correspondientes para agregar línea y para eliminar línea.

Un formulario para gestionar la serialización y deserialización. Incluirá campo dos campos de texto, “entrada” para indicar una cadena JSON de entrada. Y “salida” para devolver el objeto factura serializado en formato JSON. Y los botones correspondientes para lanzar estas funcionalidades.

Un bloque “impresión” en el que se imprimirá la factura. El contenido se actualizará al modificarse algún elemento de factura (es decir, al pulsar alguno de los botones definidos previamente).

No seas cutre, da un maquetado y formato mínimos al formulario,

NOTA: te recuerdo que `JSON.parse()` tiene un tercer parámetro muy útil.