Guia de Ejercicios

Clase II

Multiple Choice

1. ¿Qué valor tiene la variable *pelota*, luego de iniciarla de la siguiente forma: "let pelota;" e imprimiendola en pantalla (console.log(pelota))?

    a. undefined

    b. null

    c. Ninguno. Tira error en compilacion

2. ¿Qué valor tiene la variable *pelota*, luego de iniciarla de la siguiente forma: "let pelota: number;" e imprimiendola en pantalla (console.log(pelota))?

    a. Ninguno. Tira error en compilacion

    b. undefined

    c. null

3. Si tengo el siguiente código:

```
const numero: unknown = 5;
const otroNumero: number = 3;
const suma = numero + otroNumero;
```

    ¿Cuál será el valor de suma?

    a. 8

    b. 3

    c. 5

    d. Ninguno. Tira error en ejecucion

    e. Ninguno. Tira error en compilacion.

¿Qué son las interfaces en TypeScript?

a) Un tipo de dato primitivo

b) Una plantilla para definir la forma de un objeto

c) Una clase sin implementación

Respuesta: b

¿Qué se utiliza para definir una propiedad opcional en una interfaz?

a) ?

b) !

c) &

Respuesta: a

¿Qué es una clase en TypeScript?

a) Un objeto

b) Una plantilla para crear objetos

c) Una variable

Respuesta: b

¿Qué se utiliza para definir una propiedad privada en una clase en TypeScript?

a) public

b) private

c) protected

Respuesta: b

¿Qué se utiliza para crear una instancia de una clase Clase en TypeScript?

a) new Clase ()

b) Clase()

c) class Clase{}

Respuesta: a

¿Qué son los enums en TypeScript?

a) Una clase sin implementación

b) Un tipo de dato primitivo

c) Una forma de definir un conjunto de valores constantes

Respuesta: c

¿Cómo se define un enum en TypeScript?

a) enum NombreEnum {}

b) class NombreEnum {}

c) interface NombreEnum {}

Respuesta: a

¿Qué es un constructor en TypeScript?

a) Un método especial que se ejecuta cuando se crea una instancia de una clase

b) Un método especial que se ejecuta cuando se destruye una instancia de una clase

c) Un método especial que se ejecuta cuando se llama a un método de una clase

Respuesta: a

Ejercicios de enums

1. Definir un enum que represente los días de la semana (Lunes, Martes, Miercoles, Jueves, Viernes, Sabado, Domingo). Los valores serán strings con el nombre de cada día.

2. Definir un enum que represente las estaciones del año (Otoño, Invierno, Primavera, Verano). Los valores serán enteros que representan las estaciones en orden: Otoño = 1, Invierno = 2, ...

Ejercicios de JSON/Interfaces

1. Definir una interfaz Libro, que modele las propiedades de un libro: titulo, autor, número de páginas e ISBN (ISBN es un string)
2. Crear una función que tome una lista de Libros y retorne el número total de páginas de todos los libros.
3. Definir una interfaz Articulo, que modele algunas propiedades: nombre, precio y descripción.
4. Crear una función que tome un Articulo e imprima su nombre, seguido de su precio.
5. Definir una interfaz Usuario, con las propiedades: nombre, email y edad.

6. Crear una función que tome un arreglo de Usuarios y retorne la edad promedio.

Ejercicios de Clases

1. Crea una interfaz llamada Rectangulo con las propiedades ancho y alto. Luego crea una función llamada getArea que acepte un objeto de tipo Rectangulo y devuelva su área.
2. Crea una interfaz llamada Empleado con las propiedades nombre, id y salario. Luego crea una clase llamada EmpleadoDatabase que tenga un arreglo de objetos de tipo Empleado. La clase debe tener un método llamado getEmpleadoPorId que acepte un parámetro id y devuelva el objeto Empleado correspondiente.
3. Queremos trabajar con un arreglo de Vehículos (Aviones, Autos, Submarinos), de los cuales nos interesan la cantidad de ruedas, el porcentaje de tanque lleno, representado con un número decimal entre 0 y 1 (por ejemplo, 0.1 significa 10%), la matrícula y el Medio de preferencia (TIERRA, AIRE o AGUA) representados en un enum. Todos los Vehículos se pueden mover en el espacio (tendrán un objeto Posición con variables *x, y* y *z* siendo números). Los Vehículos se diferencian en la forma de moverse, ya que dependen de cada Medio (Los aviones solo pueden moverse en valores de *z >= 0*, los Submarinos solo en valores de z <= 0, y los Autos solo en valores de z = 0). Para saber si se pueden mover hacia cierta posición, todos los Vehículos tienen un método sePuedeMoverA.
    a. Modelar un Enum de Medios donde TIERRA tenga el valor 0, AGUA el valor -1 y AIRE el valor 1.

    b. Modelar las interfaces Vehiculo y Posicion, con las especificaciones del párrafo anterior.

    c. Implementar las 3 clases de Vehiculo distintas (Auto, Avion, Submarino), con cantidad de ruedas variables para aviones y autos y 0 para submarinos. El porcentaje de tanque empieza en 1.0 (100%) para todos los vehículos. El medio de preferencia de los Aviones es AGUA, el de Autos es TIERRA, el de Submarinos es AGUA.

    d. Implementar el método sePuedeMoverA de cada tipo de Vehiculo, teniendo en cuenta las especificaciones para cada tipo de Vehiculo, que recibe como argumento una variable del tipo Posicion y devuelve un valor de verdad, dependiente de lo dicho en el enunciado.

    e. Crear un arreglo de Vehiculos, con al menos un Avión y un Auto.