UD4

Ejercicios - Objetos definidos por el usuario

Ejercicio 1

El objetivo de esta actividad es practicar la creación de objetos de forma literal (haciendo uso de las **{ }** y la lista de **propiedad : valor**), actualizarlos, aumentarlos y eliminar propiedades y métodos.

<< La empresa Entretenimientos IOC, SL está trabajando en un juego de zombies y os ha pedido que genera una estructura de datos para representar los jugadores. Como se trata de una versión alfa no tiene que preocuparse por el encapsulamiento de la información, les interesa que sea lo más simple posible >>

1.1. La estructura de datos para cada “jugador” debe tener las siguientes propiedades:

nombre
apellido
empleo
nivel
puntuación

También os piden que crear una lista (array) de jugadores con 2 jugadores de ejemplo, utilizando esta estructura.

1.2 Para facilitar la comprobación de los datos de los jugadores también se pide que se añada un **método toString** a la estructura de datos para poder ver la información en una sola línea y demostrar que funciona correctamente.

1.3. Finalmente, se considera que un jugador podrá eliminar/resetear su puntuación (debe de crear un método para tal fin). Para comprobarlo se pide que se elimine la puntuación del segundo jugador para demostrar el funcionamiento correcto de este nuevo método.

Ejercicio 2

El objetivo de esta actividad es reforzar los conocimientos sobre los patrones de construcción de objetos (nosotros hemos estudiado solo uno).

El juego de zombis en el que está trabajando Entretenimientos IOC, SL está a punto de entrar en fase beta, es decir, comenzará a ser probado por personas ajenas a la empresa. Por esta razón hay que encapsular la información para evitar que usuarios malintencionados puedan modificar los datos.

Os han encargado que utilice alguna técnica (Clases de ES6) para adaptar la estructura de datos del jugador para evitar que éstas puedan ser manipuladas externamente, pero debe ser posible acceder al nombre y al método toString y debe demostrar su funcionamiento. De la misma manera, se podrá crear diferentes objetos a partir de dicha estructura.

2.1. Realice haciendo uso de clases el diseño de un objeto similar al del ejercicio 1 y pruebe su funcionalidad.

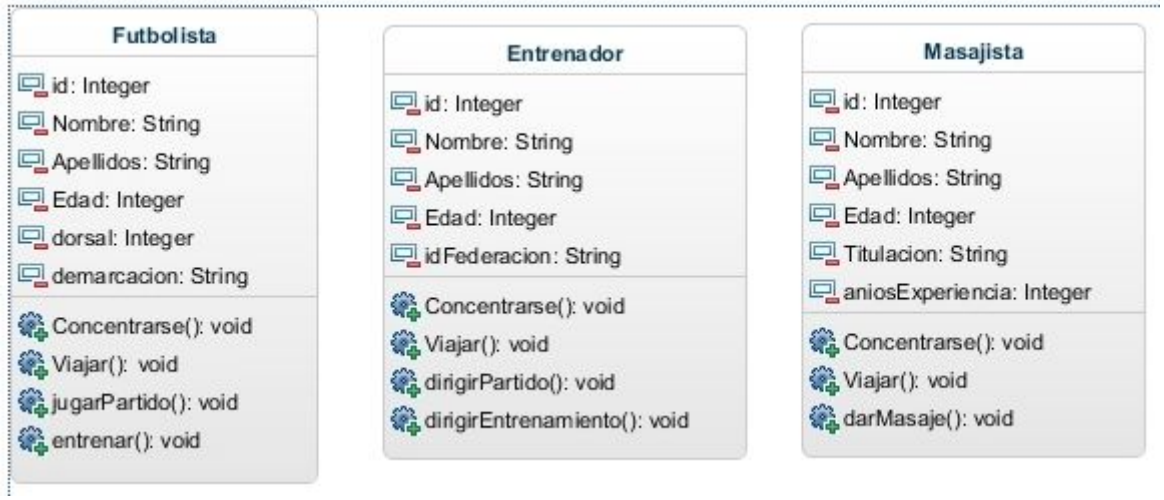
Ejercicio 3

Realice un ejercicio en que modele el siguiente caso de herencia usando Clases (ES6) que hemos estudiado en nuestros apuntes:

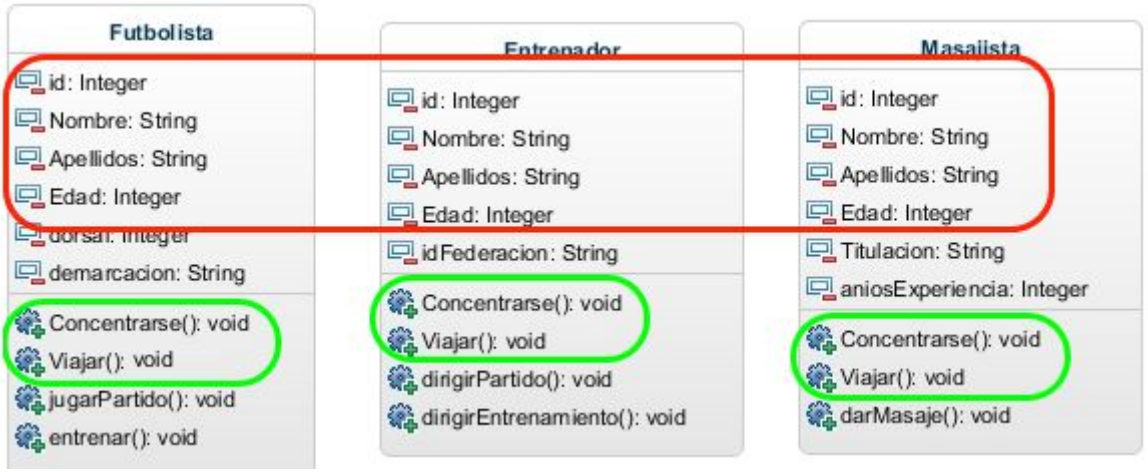
- **Utilizando la nueva especificación EcmaScript 6 que hace uso de clases.**

Importante: Los métodos constructores deben de incluir todos los parámetros necesarios para configurar todas las propiedades de los objetos/clases.

Debes de probar su funcionalidad. Debes de ser ser un poco creativo si alguno de los métodos no está claro su funcionamiento. El ejercicio es el siguiente:



Fíjese en aquellas propiedades y funciones que están en común para realizar o crear un objeto padre del que hereden los demás. Deberá usted mismo de crear el objeto padre. Una pista:



Haga una prueba de que funciona en los 3 tipos de implementación distinta.

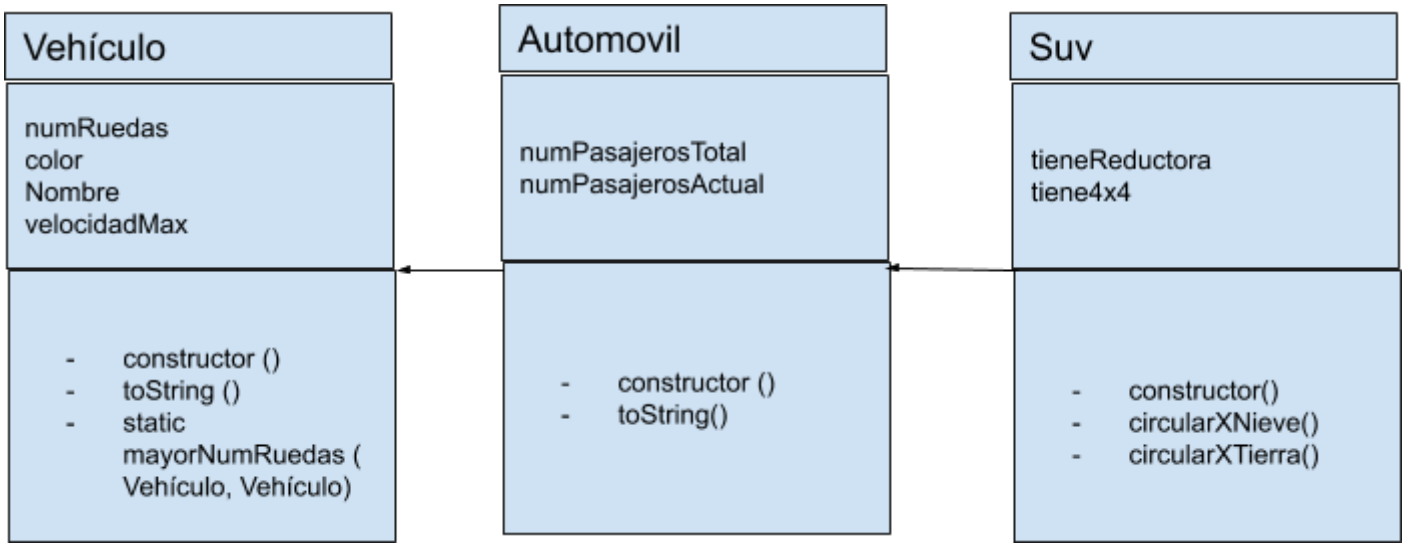
Ejercicio 4

Realice un ejercicio en que modele el siguiente caso de herencia usando “Clases” (ES6) que hemos estudiado en nuestros apuntes:

- **Utilizando la nueva especificación EcmaScript 6 que hace uso de clases.**

Importante: Los métodos constructores deben de incluir todos los parámetros necesarios para configurar todas las propiedades de los objetos/clases.

Debes de probar su funcionalidad. Debes de ser ser un poco creativo si alguno de los métodos no está claro su funcionamiento. El ejercicio es el siguiente:



Ejercicio 5

¿ Es posible cambiar un objeto declarado como “**const**” ? Razona tu respuesta

```
const usuario = {
  nombre: "Juan"
};

// Esto funciona?
usuario.nombre = "Diego";
```