

Ejercicios Unidad 3: Objetos.

EJERCICIO: u3e01_cuenta

Escribe un programa que cree un objeto "**cuenta**" a partir de un literal con las siguientes propiedades:

- Una propiedad **titular** con el valor "Alex".
- Una propiedad **saldo**, teniendo como valor inicial 0.

Crea funciones que nos permitan realizar las siguientes operaciones:

- **ingresar()** que permita añadir dinero a la cuenta, pasando la cantidad como parámetro.
- **extraer()** que permita retirar la cantidad pasada como parámetro.
- **informar()** que retorne la información del estado de la cuenta.

A continuación, añade al objeto las propiedades **saldoMáximo** con valor inicial 1000 y **fechaApertura** con valor inicial a la fecha actual. Por último, elimina la propiedad **saldoMáximo**.

Todas las operaciones que se realicen se irán mostrando en la página.

EJERCICIO: u3e02_discos

Crea una clase **Disco** que almacene la siguiente información:

- Nombre del disco.
- Grupo de música o cantante.
- Año de publicación.
- Tipo de música (podrá ser "rock", "pop", "punk" o "indie");
- Localización: almacenará un número de estantería.
- Prestado: almacenará un valor booleano. Por defecto será false.

Además, tendrá los siguientes métodos:

- Un constructor que se puede llamar con o sin parámetros:
 - Sin parámetros: las 4 primeras propiedades serán cadenas vacías, la localización será 0 por defecto y prestado estará a false.
 - Con parámetros: se pasarán solo las cinco primeras propiedades; la propiedad prestado será false.
- Un método que permitirá cambiar el número de estantería en la localización.
- Un método que permitirá cambiar la propiedad prestado.
- Un método que muestre toda la información de un disco.

El programa hará lo siguiente:

- Creará un objeto de tipo disco, para lo cual te pide los parámetros.
- Presente un menú que permita:
 - Mostrar la información del disco que se ha añadido
 - Cambiar el disco de estantería
 - Prestar un disco
 - Devolver un disco
 - Terminar

EJERCICIO: u3e03_vector

Crea una clase **Vector** que representa un vector. Al crearlo, se pasarán como parámetros dos valores numéricos que serán la **x** (distancia al punto 0,0 en el eje de las x), y la **y** (distancia al punto 0,0 en el eje de las y).

La clase tendrá definidos los métodos **get** y **set** correspondientes a las propiedades x e y.

Esta clase tendrá al menos dos métodos más: **sumar** y **restar**, que toman otro vector como parámetro, y devuelve un nuevo vector resultado de la suma o diferencia de los valores x e y de los dos vectores (el this y el parámetro).

Además, tendrá una propiedad **longitud**, que devuelve la longitud del vector, esto es, la distancia desde el origen (0,0) al punto (x, y). Para calcular esta distancia, puedes utilizar el teorema de Pitágoras: la longitud de la hipotenusa de un triángulo será rectángulo es la raíz cuadrada de la suma del cuadrado de los dos catetos del triángulo. Este cálculo, lo puedes hacer definiendo un método estático en la clase Vector.

```
console.log(new Vector(1,2).sumar(2, 3)); // => {x: 3, y: 5};  
console.log(new Vector(1,2).restar(2, 3)); // => {x: -1, y: -1};  
console.log(new Vector(3, 4).longitud); // => 5
```

EJERCICIO: u3e04_vehiculo

Crea una clase denominada **Vehículo** que tenga algunas características como el **color**, **marca**, **modelo** y la **velocidad máxima** que puede alcanzar.

También deberá tener al menos tres métodos además del constructor: **comenzar** que mostrará el mensaje "Encender motor", **parar** que mostrará "Apagar motor" y **distanciaMax** que recibirá un parámetro con indicando el tiempo de funcionamiento y calculará la distancia máxima que puede haber recorrido, que será el resultado de multiplicar la velocidad máxima por el tiempo.

Crea dos clases más, ambas deben heredar de Vehículo:

- **Coche**: cuya velocidad máxima será 120 y sobrescribirá el método parar para que muestre "Aparcar".
- **Avión**: cuya velocidad máxima será 1000 y sobrescribirá los métodos comenzar, para que muestre "Despegar", y parar, para que muestre "Aterrizar".

Crea las sentencias necesarias para crear objetos y utilizar todos los métodos definidos.