



Ejercicio 1

Crear una clase PHP que se llame **rectángulo**. Esta clase deberá tener:

- Características:
 - Base en píxeles
 - Altura en píxeles
 - Color en hexadecimal
- Métodos:
 - Constructor: que reciba base, altura y color.
 - Get y Set
 - Calcula_area: que calcule el área del rectángulo
 - Calcula_perimetro: que calcule el perímetro del rectángulo
 - toString que muestre una cadena del tipo: “este rectángulo tiene una base de <base>px y una altura de <altura>px”
 - dibuja: que dibuje el rectángulo con las características indicadas

Pista: para dibujar el rectángulo basta con crear un *div* vacío, pero con los atributos *width*, *height* y *background* correspondientes a las características del rectángulo.

Crear un documento PHP que pida al usuario la base y la altura de 3 rectángulos diferentes y los muestre primero ordenados de menor a mayor perímetro y después de menor a mayor área.



Ejercicio 2

Crear una clase PHP que se llame **círculo**. Esta clase deberá tener:

- Características:
 - Radio en píxeles.
 - Color en hexadecimal.
- Métodos
 - Constructor: que reciba radio y color.
 - Get y Set
 - Calcula_area: que calcule el área del círculo. ($\text{Pi} * \text{radio}^2$)
 - Calcula_circunferencia: que calcule la circunferencia ($2 * \text{Pi} * \text{radio}$)
 - toString que muestre una cadena del tipo: “este círculo tiene un radio de <radio>px”
 - dibuja: que dibuje el círculo con las características indicadas

Pista: para dibujar un círculo basta con crear un div vacío, pero con los atributos *width*, *height* con el valor del radio indicado y el *background* con el color indicado. Después, le aplicaremos un *border-radius* del 50% a los cuatro bordes para redondear la forma.

Crear un documento PHP que pida al usuario el radio y el color de relleno de 2 círculos y los muestre por pantalla. Mostrará primero el que tenga un área más grande.

Además, deberá mostrar toda la información de los dos círculos recibidos.



Ejercicio 3

Crea una clase PHP llamada **cuenta** que permita representa la cuenta bancaria de una persona. Esta clase debe tener:

- Características
 - titular
 - número
 - cantidad
- Métodos
 - Constructor que reciba todos los datos: la cantidad podría ser null, pero el resto nunca
 - Get y set
 - toString: que muestre por pantalla un mensaje del tipo “La cuenta número <numero> es propiedad de <titular>. En estos momentos tiene un saldo de <cantidad>”
 - ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
 - retirar(cantidad): se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0.

Crear un documento PHP que permita al usuario crearse una cuenta con un saldo inicial y después le sume o reste saldo según quiera.



Ejercicio 4

Crear una clase PHP que se llame **día**. Esta clase va a permitir almacenar la información sobre el clima de un día.

Esta clase día deberá tener:

- Características:
 - Fecha: día, mes y año del día que se quiere guardar. Se guardará como timestamp.
 - Temperatura máxima
 - Temperatura mínima
 - Clima: se guardará S para soleado, N para nublado y L para lluvioso.
- Métodos:
 - Constructor: que reciba todos los valores.
 - Get y Set
 - toString que muestre una cadena del tipo: “el día <día> de <mes> de <año> fue un día <clima> con una temperatura mínima de <temperatura mínima> y una temperatura máxima de <temperatura máxima>”
 - día: devolverá el número de día
 - mes: devolverá el nombre del mes
 - año: devolverá el número de año
 - estado: devolverá una cadena de texto correspondiente al clima que hizo ese día, sin abreviar.
 - día_semana: que devuelva un número del 1 al 7 para saber si el día era lunes, martes...
 - día_semana_nombre: que devuelva el nombre del día de la semana en español
 - calcula_promedio: que devuelve la temperatura media del día.

Crear un documento PHP que pida al usuario los valores de 2 días diferentes y muestre por pantalla toda la información del día más caluroso teniendo en cuenta la temperatura media.



Ejercicio 5

Crear una clase PHP que se llame **semana**. Esta clase pretende representar el tiempo que ha hecho durante toda una semana.

Deberá tener:

- Características:
 - Días: array con 7 elementos de la clase anterior (Dia). Este array en principio estará vacío.
- Métodos
 - Esta clase NO tiene constructor
 - Esta clase NO tiene métodos get, set o toString
 - Insertar_día: todas las características necesarias para crear el objeto de tipo **día**. El método insertará el día en la posición que corresponda según la fecha indicada. Si se ha metido, devolverá un true y si ese día ya estaba relleno, devolverá un false y no lo meterá.
 - Obtener_día: recibirá el número de día que quiere obtener. Devolverá el objeto tipo DIA correspondiente.
 - Día_mas_caluroso: devolverá el objeto tipo DIA con la temperatura máxima más alta
 - Día_menos_caluroso: devolverá el objeto tipo DIA con la temperatura máxima más baja.
 - Mostrar_semana: dibujará una tabla en la que se muestren los 7 días de la semana con toda su información.

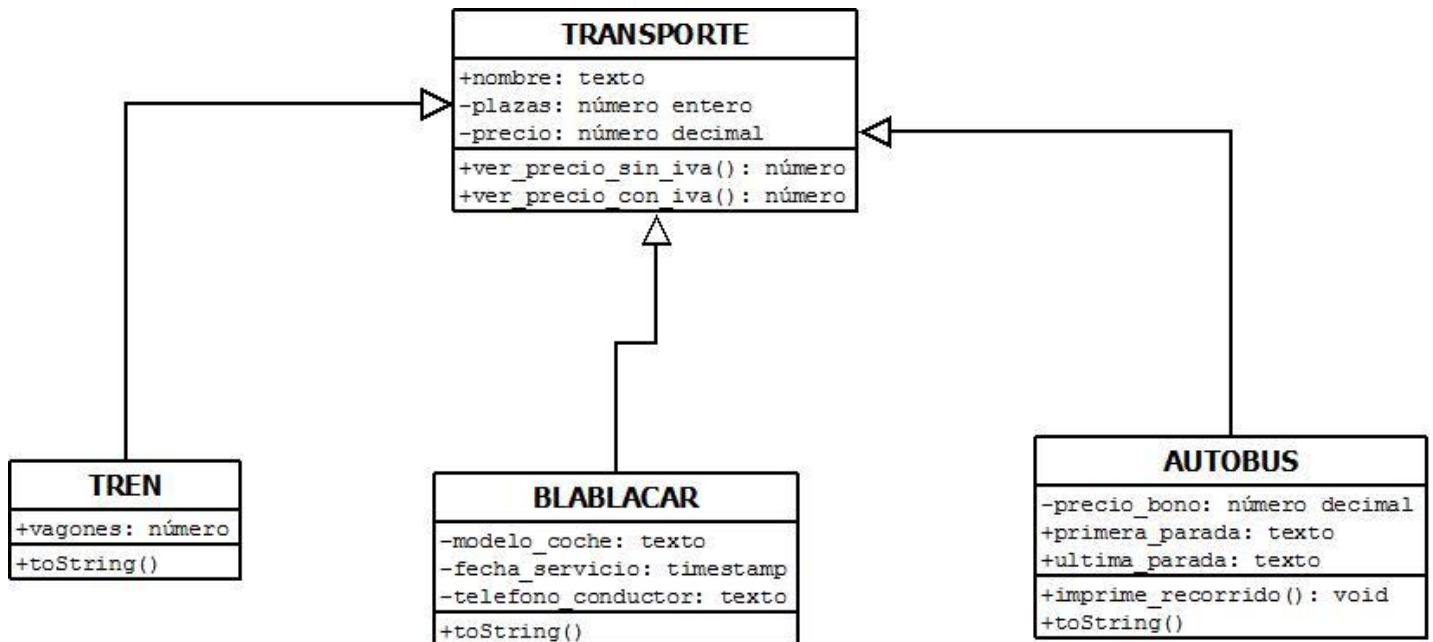
Crear un documento en PHP que por medio de formularios pida al usuario los valores de 7 días y muestre por pantalla la tabla con toda la semana.

Además, mostrará el día más caluroso y el menos caluroso.



Ejercicio 6

Crear en PHP la siguiente estructura de clases y subclases:



Se deberá tener en cuenta que:

- Para cada clase se deben crear los métodos: constructor, get y set.
- La clase **TRANSPORTE**:
 - No tendrá método `toString`
 - `Ver_precio_sin_iva` devuelve el precio, pero con el símbolo €
 - `Ver_precio_con_iva` devuelve el precio con el símbolo € y el IVA aplicado (21%)
- La clase **TREN**:
 - El método `toString` debe mostrar toda la información del tren, incluyendo los datos de nombre, plazas y precio. Además, deberá indicar el número de plazas por vagón.
- La clase **BLABLACAR**:
 - La fecha de servicio se guardará como `timestamp`
 - `Ver_fecha`: devolverá la fecha en formato `dd/mm/yyyy`
 - `Ver_dia`: devolverá el día de la fecha
 - `Ver_mes_nombre`: devolverá el nombre del mes de la fecha
 - `Ver_mes`: devolverá el número del mes de la fecha
 - `Ver_año`: devolverá el número de año con 4 dígitos.



- El método toString debe mostrar toda la información del viaje, además deberá mostrar lo que vale el viaje completo teniendo en cuenta lo que valdría una plaza.
- La clase AUTOBUS:
 - Primera_parada: almacenará el nombre de la ciudad de partida.
 - Ultima_parada: almacenará el nombre de la ciudad de destino.
 - Imprime_recorrido: mostrará por pantalla una cadena de texto con el inicio y el fin del recorrido.
 - toString: deberá mostrar toda la información del viaje. Además, deberá indicar lo que sale el viaje al comprar un bono.

Crear varios formularios y páginas PHP que permitan al usuario introducir los datos de varios viajes diferentes e interactuar con las clases.