

Ejercicios POO en PHP

Ejercicio 1

Confeccionar una clase Empleado con los atributos nombre y sueldo.

Definir un método "asigna" que reciba como dato el nombre y sueldo y actualice los atributos.

Plantear un segundo método que imprima el nombre y un mensaje si debe o no pagar impuestos (si el sueldo supera a 3000 paga impuestos).

Ejercicio 2

Confeccionar una clase Menu, con los atributos titulo y enlace (ambos son arrays). Crear los métodos necesarios que permitan añadir elementos al menú. Crear los métodos que permitan mostrar el menú en forma horizontal o vertical (según que método llamemos).

Ejercicio 3

Crear una clase cubo que contenga información sobre la capacidad y su contenido actual en litros. Se podrá consultar tanto la capacidad como el contenido en cualquier momento. Dotar a la clase de la capacidad de verter el contenido de un cubo en otro (hay que tener en cuenta si el contenido del cubo origen cabe en el cubo destino, si no cabe, se verterá solo el contenido que quepa). Hacer una página principal para probar el funcionamiento con un par de cubos.

Ejercicio 4

Creamos la clase factura con el atributo de clase IVA (21) y los atributos de instancia ImporteBase, fecha, estado (pagada o pendiente) y productos (array con todos los productos de la factura y que contienen nombre, precio y cantidad).

Define los métodos AñadeProducto, ImprimeFactura y los getters y setters de los atributos ImporteBase (solo getter, pues su contenido se actualiza automáticamente), fecha y estado.

Ejercicio 5

Queremos modelar una casa con muchas bombillas, de forma que cada bombilla se puede encender o apagar individualmente. Para ello haremos una clase Bombilla con un atributo privado que almacene si está encendida o apagada, otro para la potencia consumida y por ultimo otro atributo para la ubicación (salón, cocina, etc...); realizar un método que nos diga si una bombilla concreta está encendida, así como los getter y setters necesarios.

Además queremos poner un interruptor general de la luz, tal que si saltan los fusibles, todas las bombillas quedan apagadas. Cuando el fusible se repara, las bombillas vuelven a estar encendidas o apagadas, según estuvieran antes del percance.

Diseñar una página que genere las bombillas de una casa y las almacene en un array de sesión. Mostrar las bombillas de manera gráfica (desarrolla tu imaginación) dando la opción de encender y apagar cada una, así como de encender y apagar el interruptor general. Mostrar en todo momento la potencia consumida por las bombillas encendidas.

Ejercicio 6 (Realizarlo después de los ejercicios del libro de PHP)

Realizar la aplicación anterior del carrito de la compra en su última versión (productos almacenados en la base de datos), pero esta vez, los productos serán objetos con los

siguientes atributos: nombre, precio, stock y los siguientes métodos: constructor, reponer, vender y los getters de los atributos. Se añadirán las siguientes modificaciones: al añadir al carrito de la compra, se comprobará si hay existencias del producto (no solo hay que comprobar las existencias sino también la cantidad que hay en la cesta) y si no hay existencias no añadir el producto a la cesta. También hay que añadir un botón “Comprar” en la cesta, que la pulsar dejará la cesta vacía y restará las unidades vendidas al stock de cada producto.