

1. Crea un proyecto PHP que nos permita utilizar la clase Cafetera con atributos `capacidadMaxima` (la cantidad máxima de café que puede contener la cafetera) y `cantidadActual` (la cantidad actual de café que hay en la cafetera).

Implementar los siguientes métodos:

- Constructor `__construct()` establece la capacidad máxima en 1000 (c.c.) y la actual en cero (cafetera vacía).
- `llenarCafetera()`: hace que la cantidad actual sea igual a la capacidad máxima.
- `servirTaza(int)`: simula la acción de servir una taza con la capacidad indicada. Si la cantidad actual de café "no alcanza" para llenar la taza, se sirve lo que quede.
- `vaciarCafetera()`: pone la cantidad de café actual en cero.
- `agregarCafe(int)`: añade a la cafetera la cantidad de café indicada comprobando que no supere la capacidad máxima.
- `__toString()`: para visualizar los datos de la cafetera.

Crear un objeto `cafeteraNormal`, otro `cafeteraDescafeinado`, y comprobar su funcionamiento a través de una página web que nos permita realizar dichas funcionalidades.

2. Crea una clase `Persona` para almacenar personas. Los atributos serán el nombre, el número de DNI (entero) y la letra que le corresponde al DNI.

Implementar los siguientes métodos:

- Constructor `__construct(nombre, DNI)`: establece el nombre y el DNI (la letra se establecerá por el método privado `calculaLetra(DNI)`).
- `calculaLetra(DNI)`: calcula automáticamente la letra:
- tabla para obtener la letra que corresponde (resto del DNI/23): 0-T 1-R 2-W 3-A 4-G 5-M 6-Y 7-F 8-P 9-D 10-X 11-B 12-N 13-J 14-Z 15-S 16-Q 17-V 18-H 19-L 20-C 21-K 22-E
- `__toString()`: muestra la persona (ejemplo: Pepe__00395469-F)

Comprueba su funcionamiento a través de una página web que nos permita realizar dichas funcionalidades.

3. Crea la clase SalaCine que estará formada por una cola de objetos Persona, número de butacas total, número de butacas ocupadas y título de la película.

Implementar los siguientes métodos:

- Constructor `__construct()` que se inicializará con un título de película, número de butacas que deseemos, 0 ocupadas, 0 Personas.
- `comprarEntrada()`: pedirá el número de entradas y los datos de la persona que compra la entrada, actualizando el resto de los datos. En caso de intentar sacar una entrada y no queden butacas libres se visualizará un mensaje indicando que se ha sobrepasado el número de butacas total de la sala.
- `vaciarSalaCine()`: dejará la sala vacía.
- `__toString()`: visualizará los datos de todas las personas que se encuentran en la sala de cine.
- Añadir la clase `ExcepcionSalaLlena`, que contemple el hecho de no poder comprar entradas si la sala está llena, o no poder comprar todas las que se quieren por no quedar suficientes, mostrando el mensaje "Lo sentimos, solo le podemos ofrecer entradas".

Comprueba su funcionamiento a través de una página web que nos permita realizar dichas funcionalidades.

4. Crear una clase Empleado que modele la información que una empresa mantiene sobre cada empleado: NIF, sueldo base, pago por hora extra, horas extra realizadas en el mes, tipo (porcentaje) de IRPF, casado o no y número de hijos.

Implementar los siguientes métodos:

- Constructor `__construct` (nif, sueldobase, pagohoraextra, horasextra, tipoIRPF, estado, nHijos)

- `devuelveComplemento()`: Cálculo y retorno del complemento correspondiente a las horas extra realizadas.
- `devuelveSueldoBruto()`: Cálculo y retorno del sueldo bruto.
- `devuelveIRPF()`: Cálculo y retorno de las retenciones (IRPF) a partir del tipo, teniendo en cuenta que el porcentaje de retención que hay que aplicar es el tipo menos 2 puntos si el empleado está casado y menos 1 punto por cada hijo que tenga; el porcentaje se aplica sobre todo el sueldo bruto.
- `__toString()`: visualización de la información básica del empleado.
- `printAll()`: visualización de toda la información del empleado. La básica más el sueldo base, el complemento por horas extra, el sueldo bruto, la retención de IRPF y el sueldo neto.

Comprueba su funcionamiento a través de una página web que nos permita realizar dichas funcionalidades.

5. Crea una clase `Libro` que modele la información que se mantiene en una biblioteca sobre cada libro: título, autor (usa la clase `Persona`), ISBN, páginas, edición, editorial, lugar (ciudad y país) y fecha de edición.

La clase debe proporcionar los siguientes métodos:

- Constructor `__construct`
- `__toString()`: mostrará la información del libro con este formato:

Título: Introduction to PHP Programming

3a. edición

Autor: Liang, Y. Daniel

ISBN: 0-13-031997-X

Prentice-Hall, New Jersey (USA), 16 de noviembre de 2023

784 Páginas

Comprueba su funcionamiento a través de una página web que nos permita realizar dichas funcionalidades.

6. Crea la clase ListaLibros que incluya un array de Libros. La clase debe contener métodos para:

- Insertar un libro en la última posición.
- Eliminar un libro buscado por título.
- Modificar los datos de un libro.
- Mostrar los datos completos de un libro.
- Mostrar un listado de los títulos de todos los libros.

Comprueba su funcionamiento a través de una página web que nos permita realizar dichas funcionalidades.