

Desarrollo Web en Entorno Servidor

4.- PHP Objetos / Prácticas

IES Severo Ochoa



Ejercicio 430

- Modifica las operaciones de alquilar, tanto en `Cliente` como en `Videoclub`, para dar soporte al encadenamiento de métodos.
- Posteriormente, modifica el código de prueba para utilizar esta técnica.

Ejercicio 431.1

- Crea un trait similar al `Logger` de los apuntes.
 - Añade un método con nivel `ECHO`, que no muestre ningún nivel y muestre únicamente el mensaje.
 - Añade un método con nivel `CANI`, que tampoco muestre ningún nivel y muestre únicamente el mensaje en `Cani`, con la función que hiciste en el ejercicio 333.

Ejercicio 431.2

- Modifica el proyecto para que todos los mensajes pasen por el log
 - Clasifica los mensajes no esperados en avisos (soporte ya alquilado, cupo lleno) y errores (cuando no existe un cliente, un soporte, etc.)
 - Los métodos `muestraResumen` utilizarán el nivel CANI

Ejercicio 431.3

- Utiliza el trait de *Singleton*, de manera que la clase `Videoclub` sea un *Singleton*.
 - Modifica el constructor de `Videoclub` y renómbralo como `init($nombre)`.
 - Haz también que funcione como método encadenado.
 - En `inicio.php`, tras instanciar `Videoclub` como un *Singleton*, deberás llamar al método `init` para inicializar el `Videoclub`.
 - Para probarlo mejor, tras `listarProductos()`, crea una nueva variable mediante el *trait*.

Ejercicio 432

- Haciendo uso de *Namespaces*:
 - Coloca todas las clases/interfaces/traits en `Dwes\Videoclub`
 - Cada clase debe hacer `include_once` de los recursos que emplea
 - Coloca el/los archivos de prueba en el raíz (sin NS)
 - Desde el archivo de pruebas, utiliza `use` para poder realizar accesos sin cualificar

Ejercicio 433

- Haciendo uso de *Namespaces*:
 - Coloca los *traits* en `Dwes\Videoclub\Util` y las clases en `Dwes\Videoclub\Model`
 - Modifica el código necesario en los archivos de prueba para referenciar a los nuevos NS.

Ejercicio 434

- Reorganiza las carpeta tal como hemos visto en los apuntes: `app`, `test` y `vendor`.
- Crea un fichero `autoload.php` para registrar la ruta donde encontrar las clases
- Modifica todo el código necesario, incluyendo `autoload.php` donde sea necesario y borrando los *includes* previos.

Ejercicio 435

- Crea la excepción de aplicación `VideoclubException` en el *namespace* `Dwes\Videoclub\Util`.
- Posteriormente crea los siguientes hijos:
 - `SoporteYaAlquiladoException`
 - `CupoSuperadoException`
 - `SoporteNoEncontradoException`
- En `Cliente`, modifica los métodos `alquilar` y `devolver`, para que hagan uso de las excepciones y funcionen como métodos encadenados.

Ejercicio 436

- En `Cliente`, modifica los métodos `alquilar` y `devolver`, para que hagan uso de las nuevas excepciones (lanzándolas cuando sea necesario) y funcionen como métodos encadenados.
- En `Videoclub`, modifica `alquilarSocioPelícula` para capturar todas las excepciones e informar al usuario en consecuencia.

Ejercicio 437

- Vamos a modificar el proyecto para que el videoclub sepa qué productos están o no alquilados.
 - En `Soporte`, crea una propiedad `alquilado` que inicialmente estará a `false`.
 - Cuando se alquile, se pondrá a `true`
 - Al devolver, la volveremos a poner a `false`.
 - En `Videoclub`, crea dos nuevas propiedades y sus *getters*:
 - `numProductosAlquilados`
 - `numTotalAlquileres`

Ejercicio 438

- Crea un nuevo método en `Videoclub` llamado `alquilarSocioProductos` (`int numSocio, array numerosProducto`)
 - Debe recibir un array con los productos a alquilar.
 - Antes de alquilarlos, debe comprobar que todos los productos estén disponibles, de manera que si uno no lo está, no se le alquila ninguno.

Ejercicio 439

- **Crea dos nuevos métodos en Videoclub:**
 - `devolverSocioProducto(int numSocio, int numerosProducto)`
 - `devolverSocioProductos(int numSocio, array numerosProducto)`
 - **Deben soportar el encadenamiento de métodos.**
 - **Recuerda actualizar las propiedades:**
 - `alquilado`
 - `numProductosAlquilados`
 - `numTotalAlquileres`

¿Alguna pregunta?