

PRÁCTICA 3: Trabajando con clases.

Hitos

A) Genera una **clase** que represente la entidad de tu **hobby** favorito incluyendo todos los atributos, con sus getters y setters.

B) Instancia tres objetos de dicha clase en un script individual.

C) Crea una **superclase abstracta** denominada **Hobby** de la que herede tu entidad y que implemente la **interfaz Acciones**.

C.1) Hobby definirá los métodos **setNombre(string \$nombre)**, **getNombre(): string**

C.2) La **interfaz** contendrá los métodos **iniciar()**, **detener()**, **actualizar(array \$a)**

C.3) **Implementa** los **métodos de la interfaz** en la subclase, mostrando mensajes de texto descriptivos. Ej: "Iniciando lectura del libro \$titulo de \$n páginas...", etc,

D) Recupera la práctica 2 y crea una clase de ayuda que te devuelva los valores aleatorios adecuados para cada tipo de dato que componen los atributos de tu clase/entidad.

E) Genera N actividades distintas de tu hobby instanciando N objetos de tu clase de forma aleatoria en un programa principal, usando la clase creada en D), donde N es una constante PHP definida en el script principal.

*F) Incluye un autoload que dinamice la carga de clases de tu proyecto.

**G) Genera un array global en el programa principal que almacene una ampliación de los atributos de tu hobby (solamente claves). Después, agrega "mágicamente" dichos nuevos atributos, modificando lo realizado en E, de modo que pueda verificarse que se generan al vuelo/en memoria los nuevos atributos de tu hobby.

H) Crea el constructor y destructor de tu hobby e implementa el método mágico **__toString()**

I) Crea métodos **public**, **private** y **protected** en tu hobby que tengan sentido. Para ello genera una subclase de la clase si es necesario.

J) Genera una constante en tu clase (con sentido) y un atributo estático de tipo **int**. Comprueba que es estático con dos referencias que manipulen y muestren el valor de dicha variable estática (referencia desde clase y/o desde un método estático)

L) Resulta que tienes un segundo hobby. Crea su clase asociada, heredará de la clase abstracta. Necesitas **contabilizar** el tiempo dedicado a cada uno de ellos con métodos que te permitan establecer un tiempo máximo y mínimo. Emplea alguna estructura que te permita modularizarlo e implementarlo.

***LL) Crea un nuevo hobby, el tercero, mediante una clase anónima, sin perder la herencia, interfaces y estructuras esperadas. Es decir, instancia un objeto asignándole una clase que contenga al vuelo todas las estructuras.