SINGULAR Vs PLURAL

SINGULAR:

* Son clases q se refieren a un ejemplar del conjunto
* Se llaman Clases de ENTIDAD.
* NO SABEN que existen otros obj como ellos.
* Se usan SIEMPRE INSTANCIADAS. Ej: $UnaPeli=new película();
* Solo tienen métodos o funciones q se refieran al ppio objeto, NUNCA al conjunto de objetos de la misma clase.
* Los **()** significan que es una **FUNCION**. Y no un atributo

PLURAL (en el audio empieza en 11:43)

* Se refieren al UNIVERSO de objetos de esa clase
* Se llaman clases de UNIVERSO o clases de control.
* Nunca se instancian. O sea que sus métodos y funciones son siempre estáticas (significa que nunca se instancia)
* (Audio14:42)Devuelve datos de tipos nativos (int, string, boolean) o bien objetos del tipo de la clase de entidad que manejan. Ej: la clase alumnos puede tener un método llamado “el alumno más alto” que me devuelve un objeto de tipo alumno; puede tener un método llamado “cantidad” y me devuelve un int, puede tener un método “apellido del más alto” y me devuelve un string.

La clase UNIVERSO tiene acceso a revisar cada uno de los elementos de la clase entidad.

* Administra los objetos de la clase **ENTIDAD**

(17:08) EJEMPLO PARA QUE LA CLASE UNIVERSO ME DEVUELVA DATOS TIPO **BOOLEAN** (True or False): ¿Hay reprobados? Se debe fijar si hay o si no hay.

EJ: alumnos. Hay reprobados. (se fija de todo el universo de alumnos y se fija uno por uno y si NO hay devuelve FALSE, pero si hay reprobados devuelve TRUE, devuelve los reprobados)

* (17:50) EJEMPLO PARA QUE LA CLASE UNIVERSO ME DEVUELVA DATOS TIPO **OBJETO** Clase UNIVERSO Docente**s**. ME **DEVOLVIÓ** UN OBJETO del tipo docente, de la clase entidad que está manejando.

Ej. Clase UNIVERSO Docente**s**. MasCopado() ME DEVOLVIÓ UN OBJETO del tipo docente. Me doy cuenta porque al final del método hay **()**

* Sus métodos, func y atributos deben llevar la palabra **static**

*Public static $Cantidad;*

(atributo)

*Este atributo al llevar la palabra* **static** *puede ser utilizado directamente desde la clase, sin necesidad de instanciar.*

*Public static $TodasLasPeliculas;*

(atributo de una clase static)

*Public static function ObtenerTodas(){}*

(función de una clase static)

Un objeto de una clase no es lo mismo que una tabla. (30:08)

* ¿cómo me refiero a un atributo? Uso Self::

Ej: if (!isset (self:: $TodasLasPeliculas)) { }

(Variable de la clase Todas las películas)

Isset significa si alguien le dio un valor, si fue declarada, si existe. Si tiene algún valor me devolverá TRUE. Pero al estar el ! significa si NO existe tal variable.

Ej. De cómo se asigna un valor a una clase UNIVERSO: self::$Cantidad=count($PeliculasADevolver); guardé en el atributo $Cantidad la consulta de count.

Desde el exterior, **desde** el código PHP que usa la clase ENTIDAD, tb se usa el :: en lugar de la flechita

* Se usa el :: para referirse a un atributo o método de la clase

clase

1. *$MisPeliculas=Peliculas::ObtenerTodas();*

Me devuelve una función que supuestamente es un array lleno de objetos Película

De UNIVERSO

Guarda en $MisPeliculas

En este ejemplo me devuelve una **función** de la clase universo Películas.

1. *$CantidadPelículas=Películas::$Cantidad;*

En este ejemplo llamo a un **atributo** de la clase universo Películas. Y ese atributo lo guardo en una variable.

1. *$PeliDeMejorRating=Peliculas::getLaDeMejorRating();*

En este ejemplo me devuelve un **OBJETO** de la clase entidad

La clase universo Películas recorre un array de objetos película una por una, busca la de mejor rating y esa es la que devuelve y la que guarda en $ *PeliDeMejorRating*

Todo esto lo explica en el archivo hola.php. Explicación del archivo hola. php en el audio\_0227, desde 35:18

Luego explica el archivo película.php. Explicación del archivo película. php en el audio\_0227, desde 42:04

**ATRIBUTOS PUBLIC**

Se pueden:

1. Ver
2. Leer
3. Y asignarle valor

desde el exterior de la clase.

|  |  |
| --- | --- |
| **Private** | **Protected** |
| No se pueden ver ni modificar desde afuera | |
| La diferencia está con la herencia. | |
| No se pueden usar los atributos de la herencia | Estos atributos los puedo usar en la clase HEREDADA.  O sea, puedo usar los atributos de la clase heredada. |

Con este ej:

Public function get ID(){

Return $this-->

Logró que sea SOLO LECTURA a id. Pues le añadió getId