23/08/2018

Grupo: 800

Nombres: Daniel Ángel Franco – Arley Antonio Cabrera – Diego Alberto Sánchez

Taller: Interfaz

1. Como definir y declarar los siguientes elementos:

* Interfaz: este termino se puede entender como una plantilla con propiedades, métodos que deben asumir las clases que heredan la interface; igual número de propiedades y métodos debe declararse en la clase que hereda, la clase que hereda mínimo debe tener los de esa clase.



* Métodos: es un conjunto de instrucciones que van a realizar una acción, este puede retornar valores o no o recibir parámetros, estos se pueden declarar como públicos o privados.

function Verdura($comestible, $color="verde")

{

$this->comestible = $comestible;

$this->color = $color;

}

* Propiedades: las propiedades son valores (vacíos o no) que son propias de la clase y estos pueden ser públicos o privados, un ejemplo puede ser la clase (carro) la cual puede tener propiedades como (marca, modelo, color etc)

class MyClass

{

private $foo = FALSE;

public function \_\_construct()

{

$this->foo = TRUE;

echo($this->foo);

}

}

* Eventos: este es un suceso que se coloca en el código el cual se puede usar para comunicación en el momento que esta compilando el programa, este se declara como si fuera una propiedad.

abstract class mspwObject {  
 public function EventHandler( $event, &$object, $method ) {  
 $this->{'\_\_event\_' . $event}[] = array( $object, $method );  
 }  
 protected function EventFire( $event, Array $args ) {  
 foreach( $this->{'\_\_event\_' . $event} as $w )  
 $w[0]->{$w[1]}( $args );  
 return;  
 }  
}

1. Como derivar a partir de una interfaz base: para derivar una clase de interfaz se debe primero haber creado la clase (que va a heredar la interfaz) y al igual que en herencia se colocan los (:) para tomar como interfaz hija, una vez realizado esto se deben colocar las mismas propiedades que la clase interface (también se pueden colocar más propiedades) y los mismos métodos.
2. Cómo utilizar la palabra new para reutilizar identificadores: estos new se pueden usar para asignar valores a las propiedades de la clase interfaz, donde por ejemplo en nuestra clase program creamos el arreglo y damos los valores puede ser usando la clase que heredó la interface.

$instancia = new ClaseSencilla();

1. Cómo implementar interfaz en clases y estructuras:

Para implementar una interfaz, se utiliza el operador implements. Todos los métodos en una interfaz deben ser implementados dentro de la clase; el no cumplir con esta regla resultará en un error fatal. Las clases pueden implementar más de una interfaz si se deseara, separándolas cada una por una coma.

interface iTemplate

{

public function setVariable($name, $var);

public function getHtml($template);

}

class Template implements iTemplate

{

private $vars = array();

public function setVariable($name, $var)

{

$this->vars[$name] = $var;

}

public function getHtml($template)

{

foreach($this->vars as $name => $value) {

$template = str\_replace('{' . $name . '}', $value, $template);

}

return $template;

}

}

1. Cómo implementar métodos de interfaz con el mismo:

en las interfaces PHP solo pueden heredar de otras interfaces y las clases sólo pueden implementar interfaces. Una vez que una clase ha implementado una interfaz, todas las clases derivadas (secundarias) implementan automáticamente esa interfaz.

interface IOutput

{

public function output();

}

class Test1 implements IOutput

{

public function output()

{

return "test1";

}

}

class Test2 extends Test1

{

public function output()

{

return "test2";

}

}