

Desarrollo de Software VII

Laboratorio # 7 – Prof. Regis Rivera

Objetivo: Introducción a la programación orientada a objetos en PHP (segunda parte).

Laboratorio 7.1: Creación y uso de constantes en una clase

```
class MiClase {  
    const constante = 'valor constante';  
    function mostrarConstante() {  
        echo self::constante . "\n";  
    }  
}
```



```
<?php  
include("class_lib.php");  
echo MiClase::constante . "<br>";  
$clase = new MiClase();  
$clase->mostrarConstante();  
?>
```



Laboratorio 7.2: Abstracción de objetos

```
abstract class ClaseAbstracta {  
    //Se fuerza la herencia de la clase para definir estos métodos  
    abstract protected function tomarValor();  
    abstract protected function prefixValor($prefix);  
    // Método común  
    public function printOut() {  
        print $this->tomarValor() . "<br>";  
    }  
}  
  
class ClaseConcreta1 extends ClaseAbstracta {  
    protected function tomarValor() {  
        return "ClaseConcreta1";  
    }  
    public function prefixValor($prefix) {  
        return "{$prefix}ClaseConcreta1";  
    }  
}
```



```

class ClaseConcreta2 extends ClaseAbstracta {
    protected function tomarValor() {
        return "ClaseConcreta2";
    }
    public function prefixValor($prefix) {
        return "{$prefix}ClaseConcreta2";
    }
}

```



class_lib.php

```

<?php
include("class_lib.php");
$class1 = new ClaseConcreta1;
$class1->printOut();
echo $class1->prefixValor('FOO_') . "<br>";
$class2 = new ClaseConcreta2;
$class2->printOut();
echo $class2->prefixValor('FOO_') . "<br>";
?>

```



Lab72.php

Laboratorio 7.3: Interfaces

```

interface iTemplate {
    public function ponerVariable($nombre, $var);
    public function verHtml($template);
}

```



class_lib.php

```

class Template implements iTemplate {
    private $vars = array();
    public function ponerVariable($nombre, $var) {
        $this->vars[$nombre] = $var;
    }
    public function verHtml($template) {
        foreach($this->vars as $nombre => $value) {
            $template = str_replace('{ ' . $nombre . ' }', $value, $template);
        }
        return $template;
    }
}

```



class_lib.php

```
<?php
include("class_lib.php");
$tpl = new Template;
$prueba = "valor de prueba";

$tpl->ponerVariable("var1", "Valor 1");
$tpl->ponerVariable("var2", "Valor 2");
$tpl->ponerVariable("var3", "Valor 3");

echo $tpl->verHtml($prueba);

?>
```



Lab73.php

Laboratorio 7.4: Auto carga de clases (no incluidas en tiempo de diseño)

```
<?php
class miClase {
    public function __construct() {
        echo "Mi Clase ha sido agregada!!! <br>";
    }
}
?>
```



miClase.php

Nota: Guardar *miClase.php* en la sub carpeta llamada *clases*

```
<?php
class miotraClase {
    public function __construct() {
        echo "Mi segunda Clase ha sido agregada!!! <br>";
    }
}
?>
```



miotraClase.php

Nota: Guardar *miotraClase.php* en la sub carpeta llamada *clases*

```
<?php

spl_autoload_register(function ($nombre_clase) {
    $archivo = "clases/" . $nombre_clase . ".php";
    if(file_exists($archivo)) include_once($archivo);
});

$obj = new miClase();
$obj2 = new miotraClase();

?>
```



Lab74.php

Laboratorio 7.5: Clonación de objetos

```
class SubObject
{
    static $instances = 0;
    public $instance;

    public function __construct() {
        $this->instance = ++self::$instances;
    }

    public function __clone() {
        $this->instance = ++self::$instances;
    }
}
```



```
class MyCloneable
{
    public $object1;
    public $object2;

    function __clone()
    {
        // Forzar una copia de this->object
        $this->object1 = clone $this->object1;
    }
}
```



```
<?php

include("class_lib.php");

$obj = new MyCloneable();

$obj->object1 = new SubObject();
$obj->object2 = new SubObject();

$obj2 = clone $obj;

print("<br>Original Object: ");
print_r($obj);

print("<br>Cloned Object: ");
print_r($obj2);

?>
```



Laboratorio 7.6: Uso de formularios en orientación a objetos I

```
<!DOCTYPE html>
<html>
<head>
<title>Laboratorio 7.6</title>
<meta charset="utf-8">
</head>
<body>
<form name="formularioDatos" method="post" action="lab76.php">
<p> CALCULO DE VOLUMEN y AREA DE UN CILINDRO </p>
<br/>
Introduzca el diametro en metros: <input type="text" name="diam"
value="">
<br/> <br/>
Introduzca la altura en metros: <input type="text" name="altu" value="">
<br/> <br/>
<input value="Calcular" type="submit" />
</form>
</body>
</html>
```



Lab76.html

```
class Cilindro
{
    protected $pi;
    protected $diametro;
    protected $altura;
    protected $radio;

    function __construct($d, $a){
        $this->diametro = $d;
        $this->altura = $a;
        $this->pi = 3.141593;
        $this->radio=$d/2;
    }

    function obtener_radio(){
        return $radio;
    }

    function calc_volumen(){
        return $this->pi*$this->radio*$this->radio*$this->altura;
    }

    function calc_area(){
        return 2*$this->pi*$this->radio*($this->radio+$this->altura);
    }
}
```



class_lib.php

```

<?php
include('class_lib.php');

$diam = $_POST['diam'];
$altu = $_POST['altu'];

$cil = new Cilindro($diam,$altu);

$volumen=$cil->calc_volumen();
$area=$cil->calc_area();

echo "<br/>    El volumen del cilindro es de ". $volumen . " metros cubicos";
echo "<br/>    El area del cilindro es de ". $area . " metros cuadrados";

?>

```



Lab76.php

Laboratorio 7.7: Uso de formularios en orientación a objetos II

```

<?php

if(array_key_exists('enviar', $_POST)){

include('class_lib.php');

$diam = $_POST['diam'];
$altu = $_POST['altu'];

$cil = new Cilindro($diam,$altu);

$volumen=$cil->calc_volumen();
$area=$cil->calc_area();

echo "<br/>    El volumen del cilindro es de ". $volumen . " metros cubicos";
echo "<br/>    El area del cilindro es de ". $area . " metros cuadrados";

}
else{
?>

<!DOCTYPE html>
<html>
<head>
<title>Laboratorio 7.7</title>
<meta charset="utf-8">
</head>
<body>
<form name="formularioDatos" method="post" action="lab77.php">
<p> CALCULO DE VOLUMEN y AREA DE UN CILINDRO </p>

```



Lab77.php

```
<br/>
Introduzca el diametro en metros: <input type="text" name="diam"
value="">
<br/> <br/>
Introduzca la altura en metros: <input type="text" name="altu" value="">
<br/> <br/>
<input value="Calcular" name="enviar" type="submit" />
</form>
</body>
</html>
```

```
<?php
}
?>
```