



Guía Práctica de Ejercicios N° 6

En nuestra sexta práctica continuaremos con la programación orientada a objetos, definiendo algunos conceptos fundamentales del paradigma POO, mencionando algunas de las características del mismo y luego pasaremos a la implementación de la POO en PHP versión 5.

Objetivo: Comprender y profundizar los conceptos básicos de la programación orientada a objetos, definiendo las clases y objetos con sus atributos y métodos. Definir, desarrollar y utilizar métodos setter, getter y toString. Entender y aplicar las relaciones entre clases utilizando como lenguaje de modelado UML y PHP para codificar las mismas. Entender el uso constantes de clase, su definición y uso, ya sea dentro de la definición de clase o fuera de la misma. Utilizar las palabras reservadas `this` y `self` para referenciar constantes y variables de una clase u objeto.

Introducción: En nuestra actualidad vemos que el desarrollo de Internet ha sido inminente y con ello las aplicaciones Web, por lo tanto, se hace indispensable el uso de un lenguaje que permita desarrollar aplicaciones Web como PHP, entre otros. Teniendo en cuenta la situación anterior es que veremos la como desarrollar aplicaciones Web que se ejecutarán del lado del servidor utilizando uno de los mejores lenguajes del ambiente del Software Libre. Teniendo en cuenta la magnitud del proyecto a desarrollar veremos la necesidad de emplear el paradigma orientado a objetos con PHP; que nos dará la posibilidad de diseñar, desarrollar y mantener el Software de manera profesional utilizando un paradigma antes mencionado.

Construcciones (palabras clave), variables globales y funciones que estudiaremos en esta unidad.

Construcciones del lenguaje y operadores con los que trabajaremos.

class nos permite crear una clase en PHP que es una colección de variables (atributos) y funciones (métodos) que trabajan con estas variables.

public nos permite definir atributos y métodos para que sean accedidos desde cualquier lugar.

private nos permite definir atributos y métodos para que sean accedidos desde la clase donde fueron definidos.

new nos permite instanciar un objeto de acuerdo a una clase.

this nos permite hacer referencia a una instancia (objeto) particular de la clase. Dejando la posibilidad de acceder a comportamientos y atributos de un objeto dentro de la definición de una clase.

self nos permite acceder a variables estáticas y constantes de una clase.

:: el operador de ámbito nos permite acceder a elementos estáticos, constantes y sobrescribir métodos de una clase.



Ejercicios

Utilizar los conceptos de programación Web y la POO

1) – Escriba los scripts PHP para definir en primer lugar una clase de nombre Alumno (**Alumno.php**) que implemente métodos para establecer y obtener los valores para todos los atributos. En segundo lugar deberá definir una clase denominada Comision (**Comision.php**) que implemente los métodos necesarios para establecer y obtener los valores de todos sus atributos. Y en tercer lugar se deberá crear una clase de nombre Asignatura (**Asignatura.php**) que también implemente y desarrolle los métodos necesarios para establecer y obtener los datos de los distintos atributos. Finalmente, en este ejercicio se le pedirá que utilice una clase Index (**index.php**) para crear los alumnos, las comisiones y las asignaturas.

Index.php

Métodos

(público) ejecutar()

Asignatura.php

Atributos

(privado) _codigo
(privado) _nombre
(privado) _observaciones
(privado) _comisiones

Métodos

(público) __construct()
(público) setCodigo(\$codigo)
(público) getCodigo()
(público) setNombre(\$nombre)
(público) getNombre()
...

Comision.php

Atributos

(privado) _codigo
(privado) _lugar
(privado) _dia
(privado) _horaInicio
(privado) _horaFin
(privado) _cupos
(privado) _alumnos



Métodos

```
(público) __construct()  
(público) setCodigo($codigo)  
(público) getCodigo()  
(público) setLugar($lugar)  
(público) getLugar()  
...
```

Alumno.php

Atributos

```
(privado) _codigo  
(privado) _apellidos  
(privado) _nombres  
(privado) _numeroDocumento  
(privado) _tipoDocumento  
(privado) _tipoAlumno
```

Métodos

```
(público) __construct()  
(público) setCodigo($codigo)  
(público) getCodigo()  
(público) setApellidos($apellidos)  
(público) getApellidos()  
...
```

index.php

```
<?php  
require_once 'Alumno.php';  
require_once 'Comision.php';  
require_once 'Asignatura.php';  
  
class Index  
{  
    public static function ejecutar()  
    {  
        $alumnoGarcia = new Alumno();  
        $alumnoGarcia->setApellidos('Garcia');  
        $alumnoGarcia->setNombres('Juan');  
        $alumnoGarcia->setTipoAlumno('regular');  
  
        $alumnoGonzalez = new Alumno();  
        $alumnoGonzalez->setApellidos('Gonzalez');  
        $alumnoGonzalez->setNombres('Juan');  
        $alumnoGonzalez->setTipoAlumno('regular');  
  
        $comisionViernes = new Comision();  
        $comisionViernes->setDia('Viernes');
```



```
$comisionViernes->setAlumno($alumnoGarcia);  
$comisionSabados = new Comision();  
$comisionSabados->setDia('Sabados');  
$comisionSabados->setAlumno($alumnoGonzalez);
```

```
$oAsignatura = new Asignatura();  
$oAsignatura->getNombre('Programación I');  
$oAsignatura->setComision($comisionViernes);  
$oAsignatura->setComision($comisionSabados);
```

```
var_dump($oAsignatura);
```

```
}
```

```
}
```

```
Index::ejecutar();
```

Asignatura.php

```
<?php
```

```
class Asignatura
```

```
{
```

```
    private $_codigo;  
    private $_nombre;  
    private $_observacion;  
    private $_comisiones = array();
```

```
    public function getCodigo()
```

```
    {
```

```
        return $this->_codigo;
```

```
    }
```

```
    public function setCodigo($codigo)
```

```
    {
```

```
        $this->_codigo = $codigo;
```

```
    }
```

```
    public function getNombre()
```

```
    {
```

```
        return $this->_nombre;
```

```
    }
```

```
    public function setNombre($nombre)
```

```
    {
```

```
        $this->_nombre = $nombre;
```

```
    }
```

```
    public function getObservacion()
```

```
    {
```

```
        return $this->_observacion;
```

```
    }
```

```
    public function setObservacion($observacion)
```

```
    {
```

```
        $this->_observacion = $observacion;
```

```
    }
```



```
        public function getComisiones()
        {
            return $this->_comisiones;
        }
        public function setComision(Comision $oComision)
        {
            $this->_comisiones[] = $oComision;
        }
    }
```

Comision.php

<?php

require_once 'Alumno.php';

class Comision

```
{
    private $_codigo;
    private $_lugar;
    private $_dia;
    private $_horaInicio;
    private $_horaFin;
    private $_cupo;
    private $_alumnos = array();

    public function getCodigo()
    {
        return $this->_codigo;
    }
    public function setCodigo($codigo)
    {
        $this->_codigo = $codigo;
    }
    public function getLugar()
    {
        return $this->_lugar;
    }
    public function setLugar($lugar)
    {
        $this->_lugar = $lugar;
    }
    public function getDia()
    {
        return $this->_dia;
    }
    public function setDia($dia)
    {
        $this->_dia = $dia;
    }
}
```



```
public function getHoraInicio()
{
    return $this->_horaInicio;
}
public function setHoraInicio($horaInicio)
{
    $this->_horaInicio = $horaInicio;
}
public function getHoraFin()
{
    return $this->_horaFin;
}
public function setHoraFin($horaFin)
{
    $this->_horaFin = $horaFin;
}
public function getCupo()
{
    return $this->_cupo;
}
public function setCupo($cupo)
{
    $this->_cupo = $cupo;
}
public function getAlumnos()
{
    return $this->_alumnos;
}
public function setAlumno(Alumno $oAlumno)
{
    $this->_alumnos[] = $oAlumno;
}
}
```

Alumno.php

<?php

class Alumno

```
{
    private $_codigo;
    private $_apellidos;
    private $_nombres;
    private $_numeroDocumento;
    private $_tipoDocumento;
    private $_tipoAlumno;

    public function getCodigo()
    {
        return $this->_codigo;
    }
}
```



```
}  
public function setCodigo($codigo)  
{  
    $this->_codigo = $codigo;  
}  
public function getApellidos()  
{  
    return $this->_apellidos;  
}  
public function setApellidos($apellidos)  
{  
    $this->_apellidos = $apellidos;  
}  
public function getNombres()  
{  
    return $this->_nombres;  
}  
public function setNombres($nombres)  
{  
    $this->_nombres = $nombres;  
}  
public function getNumeroDocumento()  
{  
    return $this->_numeroDocumento;  
}  
public function setNumeroDocumento($numeroDocumento)  
{  
    $this->_numeroDocumento = $numeroDocumento;  
}  
public function getTipoDocumento()  
{  
    return $this->_tipoDocumento;  
}  
public function setTipoDocumento($tipoDocumento)  
{  
    $this->_tipoDocumento = $tipoDocumento;  
}  
public function getTipoAlumno()  
{  
    return $this->_tipoAlumno;  
}  
public function setTipoAlumno($tipoAlumno)  
{  
    $this->_tipoAlumno = $tipoAlumno;  
}  
}
```

2) – Teniendo en cuenta el ejemplo de clases y objetos de PHP (Index, Facultad, Persona, Mascota) de la clase de teoría se le solicitará que termine de implementar el mismo, creando y definiendo las clases que corresponde.



```
index.php
<?php
require_once 'Facultad.php';
require_once 'Mascota.php';
require_once 'Persona.php';
class Index
{
    public function ejecutar()
    {
        $oPersona = new Persona('Garcia', 'Juan', 23);
        $oPerro = new Mascota('Kirby', 'blanco, negro y marron');

        echo $oPersona->tocar($oPerro, 'cabeza');
    }
}
$oIndex = new Index();
$oIndex->ejecutar();
```

```
Persona.php
<?php
require_once 'Mascota.php';
class Persona
{
    private $_apellidos;
    private $_nombres;
    private $_edad;
    private $_hermanos = array();
    private $_mascotas = array();
    public function __construct($apellidos, $nombres, $edad)
    {
        $this->_apellidos = $apellidos;
        $this->_nombres = $nombres;
        $this->_edad = $edad;
    }
    public function agregarHermano(Persona $persona)
    {
        $this->_hermanos[] = $persona;
    }
    public function agregarMascota(Mascota $mascota)
    {
        $this->_mascotas[] = $mascota;
    }
    public function tocar(Mascota $mascota, $lugar)
    {
        return $mascota->tocar($lugar);
    }
}
```

Mascota.php



```
<?php
class Mascota
{
    private $_nombre;
    private $_color;
    public function __construct($nombre, $color)
    {
        $this->_nombre = $nombre;
        $this->_color = $color;
    }
    public function tocar($lugar)
    {
        if ($lugar = 'cabeza') {
            return $this->_moverCola();
        }
    }
    private function _moverCola()
    {
        return 'muevo la cola!!!';
    }
}
```

Facultad.php

```
<?php
require_once 'Persona.php';
class Facultad
{
    private $_nombre;
    private $_alumnos;
    public function __construct($nombre)
    {
        $this->_nombre = $nombre;
    }
    public function agregarAlumno(Persona $persona)
    {
        $this->_alumnos[] = $persona;
    }
}
```

3) – Realice un script de nombre Persona.php que defina y use una clase (Persona) con los siguientes atributos y métodos:

Persona.php

Atributos

- (privado) tipoDocumento
- (privado) numeroDocumento
- (privado) apellidos
- (privado) nombres



Métodos

- (privado) telefono
- (privado) correoElectronico
- (público) __construct()
- (público) cambiarNumeroTelefono()
- (público) cambiarCorreoElectronico()
- (público) obtenerApellidos()
- (público) obtenerNombres()

index.php
<?php

require_once 'Persona.php';

```
class Index
{
    public function ejecutar()
    {
        $oPersona = new Persona('DNI', 22444222, 'Garcia', 'Juan');
        echo $oPersona->obtenerApellidos();
        echo "<br />";
        echo $oPersona->obtenerNombres();
    }
}
```

```
$oIndex = new Index();
$oIndex->ejecutar();
```

Persona.php
<?php

```
class Persona
{
    private $_tipoDocumento;
    private $_numeroDocumento;
    private $_apellidos;
    private $_nombres;
    private $_telefono;
    private $_correoElectronico;
    public function __construct($tipoDocu, $numDocu, $apellidos, $nombres)
    {
        $this->_tipoDocumento = $tipoDocu;
        $this->_numeroDocumento = $numDocu;
        $this->_apellidos = $apellidos;
        $this->_nombres = $nombres;
    }
    private function _establecerTelefono($telefono)
    {
        $this->_telefono = $telefono;
    }
}
```



```
}  
public function cambiarTelefono($telefono)  
{  
    $this->_establecerTelefono($telefono);  
}  
private function establecerCorreoElectronico($email)  
{  
    $this->_correoElectronico = $email;  
}  
public function cambiarCorreoElectronico($email)  
{  
    $this->establecerCorreoElectronico($email);  
}  
public function obtenerApellidos()  
{  
    return $this->_apellidos;  
}  
public function obtenerNombres()  
{  
    return $this->_nombres;  
}  
}
```

4) – Mejore la clase Alumno (Alumno.php) del primer ejercicio de la practica empleando constantes para determinar el tipo de alumno. La mejora se puede agregar en el constructor y el método getter de la clase.

Alumno.php

```
<?php
```

```
class Alumno
```

```
{  
    const ALUMNO_LIBRE = "L";  
    const ALUMNO_REGULAR = 'R';  
    const ALUMNO_PASIVO = 'P';  
    const ALUMNO_ACTIVO = 'A';  
  
    private $_codigo;  
    private $_apellidos;  
    private $_nombres;  
    private $_numeroDocumento;  
    private $_tipoDocumento;  
    private $_tipoAlumno;  
  
    public function __construct($tipoAlumno = self::ALUMNO_ACTIVO)  
    {  
        $this->_tipoAlumno = $tipoAlumno;  
    }  
    public function getCodigo()  
    {
```



```
        return $this->_codigo;
    }
    public function setCodigo($codigo)
    {
        $this->_codigo = $codigo;
    }
    public function getApellidos()
    {
        return $this->_apellidos;
    }
    public function setApellidos($apellidos)
    {
        $this->_apellidos = $apellidos;
    }
    public function getNombres()
    {
        return $this->_nombres;
    }
    public function setNombres($nombres)
    {
        $this->_nombres = $nombres;
    }
    public function getNumeroDocumento()
    {
        return $this->_numeroDocumento;
    }
    public function setNumeroDocumento($numeroDocumento)
    {
        $this->_numeroDocumento = $numeroDocumento;
    }
    public function getTipoDocumento()
    {
        return $this->_tipoDocumento;
    }
    public function setTipoDocumento($tipoDocumento)
    {
        $this->_tipoDocumento = $tipoDocumento;
    }
    public function getTipoAlumno()
    {
        return $this->_tipoAlumno;
    }
    public function setTipoAlumno($tipoAlumno)
    {
        switch($tipoAlumno)
        {
            case self::ALUMNO_ACTIVO:
                $this->_tipoAlumno = self::ALUMNO_ACTIVO;
                break;
        }
    }
}
```



```
        case self::ALUMNO_PASIVO:
            $this->_tipoAlumno = self::ALUMNO_PASIVO;
            break;
        case self::ALUMNO_REGULAR:
            $this->_tipoAlumno = self::ALUMNO_REGULAR;
            break;
        case self::ALUMNO_LIBRE:
            $this->_tipoAlumno = self::ALUMNO_LIBRE;
            break;
        default:
            $this->_tipoAlumno = self::ALUMNO_ACTIVO;
    }
}
```

5) – Para los puntos 2, 3 y 4 agregar el método `__toString()` mostrando información que identifique el objeto.

Facultad.php (punto 2)

```
public function __toString()
{
    return $this->_nombre;
}
```

Persona.php (punto 2)

```
public function __toString()
{
    return $this->_apellidos.", ".$this->_nombres;
}
```

Mascota.php (punto 2)

```
public function __toString()
{
    return $this->_nombre;
}
```

Persona.php (punto 3)

```
public function __toString()
{
    return $this->_tipoDocumento. " ".$this->_numeroDocumento;
}
```

Alumno.php (punto 4)

```
public function __toString()
{
    return $this->_codigo. ' - ' . $this->_apellidos. ', ' . $this->_nombres;
}
```

