

# Ejercicios PHP 6

## 1. Libro

Hay que crear una clase “Libro” con las propiedades “título”, “autor” y “número de páginas”, crear un método llamado “mostrarInfo” que muestre los datos de la clase, crear una instancia de la clase y mostrar los datos de la misma.

```
<?php
class Libro{

    public $titulo; // Título del libro
    public $autor; // Autor del libro
    public $numero_paginas; // Numero de páginas del libro

    public function mostrarInfo(){ //Muestra los datos del libro

        return "Título: ".$this->titulo."\n".
               "Autor: ".$this->autor."\n".
               "Número de páginas: ".$this->numero_paginas."\n";

    }

}

$libro = new Libro();
$libro->titulo = "Viento y verdad";
$libro->autor = "Brandon Sanderson";
$libro->numero_paginas = "1406";

echo $libro->mostrarInfo();

?>
```

### ▼ TERMINAL

```
PS C:\xampp\htdocs\programacion\ejercicio6> php libro.php
Título: Viento y verdad
Autor: Brandon Sanderson
Número de páginas: 1406
PS C:\xampp\htdocs\programacion\ejercicio6> |
```

## 2. Círculo

Hay que crear una clase “Círculo” con la propiedad “radio”, crear un método llamado “calcularArea” que calcule el área del círculo, crear una instancia de radio 5 de la clase y hallar el área del mismo.

```
<?php
class Círculo{

    public $radio; // Radio del círculo

    public function calcularArea(){ // Calcula el área del círculo

        return "Área: " . (($this->radio)*($this->radio))*3.14;

    }

}

$círculo = new Círculo();
$círculo->radio = 5;

echo $círculo->calcularArea();

?>
```

### TERMINAL

```
PS C:\xampp\htdocs\programacion\ejercicio6> php círculo.php
Área: 78.5
PS C:\xampp\htdocs\programacion\ejercicio6> █
```

### 3. Vehículo y coche

Hay que crear una clase “Vehículo” con la propiedad “marca” y el método “encender”, crear una clase hija “Coche” que añade la propiedad “modelo”, crear una instancia de la clase “Coche” y utilizar el método “encender” con la misma.

```
<?php
class Vehiculo{

    public $marca; // Marca del vehículo

    public function encender(){ //Indica que el vehículo esta encendido

        return "El vehículo esta encendido"."\\n";

    }

}

class Coche extends Vehiculo{

    public $modelo; //Modelo del coche

}

$coche = new Coche();
$coche->marca = "Citroen";
$coche->modelo = "D-45";

echo $coche->encender();

?>
```

#### TERMINAL

```
PS C:\xampp\htdocs\programacion\ejercicio6> php vehiculo.php
El vehículo esta encendido
PS C:\xampp\htdocs\programacion\ejercicio6> █
```

## 4. Empleado y Gerente

Hay que crear una clase “Empleado” con las propiedades “nombre” y “sueldo” y el método “mostrarDetalles” que muestre los datos del empleado, crear una clase hija “Gerente” que añade la propiedad “departamento” y modifica el método “mostrarDetalles” para incluir la nueva propiedad, crear una instancia de la clase “Empleado” y otra de “Gerente” y mostrar los datos de ambos.

```
<?php
class Empleado{

    public $nombre; //Nombre del empleado
    public $sueldo; //Sueldo del empleado

    public function mostrarDetalles(){ //Muestra los datos del empleado

        echo "Nombre: ".$this->nombre."\n".
            "Sueldo: ".$this->sueldo."\n";

    }

}

class Gerente extends Empleado{

    public $departamento; //Departamento del gerente

    public function mostrarDetalles(){ //Muestra los datos del gerente

        return "Nombre: ".$this->nombre."\n".
            "Sueldo: ".$this->sueldo."\n".
            "Departamento: ".$this->departamento."\n";

    }

}

$empleado = new Empleado();
$empleado->nombre = "Juan Perales";
$empleado->sueldo = 150.5;

$empleado->mostrarDetalles();

$gerente = new Gerente();
$gerente->nombre = "Agapito Morales";
$gerente->sueldo = 247.3;
$gerente->departamento = "Ventas";

echo $gerente->mostrarDetalles();
```

```
?>
```

#### TERMINAL

```
PS C:\xampp\htdocs\programacion\ejercicio6> php empleado.php
Nombre: Juan Perales
Sueldo: 150.5
Nombre: Agapito Morales
Sueldo: 247.3
Departamento: Ventas
PS C:\xampp\htdocs\programacion\ejercicio6> █
```

## 5. Calculadora

Hay que crear la clase “Calculadora” que utilizando los métodos “sumar”, “restar”, “multiplicar” y “dividir” sea capaz de hacer operaciones básicas. En “dividir” introducimos un try – catch para evitar errores al intentar dividir entre 0, creamos una nueva instancia y un bucle do while que permite introducir nuevos datos y seguir operando hasta que introduzcamos 0 en el operador.

```
<?php
class Calculadora{

    public function sumar($atributo1, $atributo2){ //Hace una suma

        return $atributo1 + $atributo2;

    }

    public function restar($atributo1, $atributo2){ //Hace una resta

        return $atributo1 - $atributo2;

    }

    public function multiplicar($atributo1, $atributo2){ //Hace una
multiplicación

        return $atributo1 * $atributo2;

    }

    public function dividir($atributo1, $atributo2){ //Hace una división

        if ($atributo2 == 0) { //Comprueba si la división es posible
```

```

        throw new Exception("No se puede dividir entre 0");

    } else {

        return $atributo1 / $atributo2;

    }

}

}

$operacion = new Calculadora();

do { //Permite operar hasta que se introduzca 0 en el operador

    $atributo1 = readline("Introduzca el primer número: "); //Primer
número de la operación
    $atributo2 = readline("Introduzca el segundo número: "); //Segundo
número de la operación
    $operador = readline("Introduzca el operador: "); //Operador de la
operación

    try {

        if ($operador == "+"){

            echo "Resultado: ".$operacion->sumar($atributo1,
$atributo2)."\n";

        } elseif ($operador == "-"){

            echo "Resultado: ".$operacion->restar($atributo1,
$atributo2)."\n";

        } elseif ($operador == "*"){

            echo "Resultado: ".$operacion->multiplicar($atributo1,
$atributo2)."\n";

        } elseif ($operador == "/"){

            echo "Resultado: ".$operacion->dividir($atributo1,
$atributo2)."\n";

        } elseif ($operador == 0){

        } else { //Mensaje que sale si no se introduce un operador válido

            echo "Operación no válida". "\n";

```

```

    }
}

    catch (Exception $e) { //Mensaje que sale si se intenta dividir entre
0
        echo "División imposible"."\\n";
    }
} while ($operador != 0)

?>

```

```

TERMINAL
PS C:\xampp\htdocs\programacion\ejercicio6> php calculadora.php
Introduzca el primer número: 5
Introduzca el segundo número: 6
Introduzca el operador: +
Resultado: 11
Introduzca el primer número: 2
Introduzca el segundo número: 3
Introduzca el operador: -
Resultado: -1
Introduzca el primer número: 3
Introduzca el segundo número: 4
Introduzca el operador: *
Resultado: 12
Introduzca el primer número: 3
Introduzca el segundo número: 0
Introduzca el operador: /
División imposible
Introduzca el primer número: 4
Introduzca el segundo número: 3
Introduzca el operador: 0
PS C:\xampp\htdocs\programacion\ejercicio6> 

```

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>