

# Práctica UD-3 Sesión-3

Objetivos	<ul style="list-style-type: none"><li>• Aprender más en profundidad PHP a través de ejercicios básicos</li></ul>
Instrucciones de entrega	<ul style="list-style-type: none"><li>• No hay que entregar nada.</li></ul>

- Cree un nuevo proyecto en NetBeans (de tipo PHP->PHP Application) o en vuestro editor favorito con nombre "**PHPPRACTICAUD3SESION3**". Añade un nuevo archivo **Main.php** donde vamos a añadir los *include* a los archivos que crearemos y el código de prueba.

## Clases y objetos

### 1. Clase Animal

- 1.1. Cree un archivo **Animal.php** y defina dentro de él una clase llamada *Animal* sin métodos ni propiedades.
- 1.2. En el archivo Main.php crea una instancia de la clase Animal asignada a una variable llamada 'perro'.
- 1.3. Agregue un método llamado *Ladra* en tu clase Animal. El método debe contener lo siguiente: `printf("¡El perro dice guau!");`. El método no debe devolver nada.
- 1.4. Llame al método *Ladra* de la variable 'perro'.
- 1.5. Agregue una propiedad a su clase llamada *sonido* y cambie su método *Ladra* para escribir la oración "El perro dice" y el valor de la propiedad *sonido*.
- 1.6. Llame al método *Ladra* de la variable 'perro'.
- 1.7. Agregue un constructor a la clase *Animal* que acepte un parámetro llamado "sonido" el cual se asignará al atributo *sonido*.
- 1.8. Llame de nuevo al método *Ladra* de la variable 'perro'.

## 2. Clase Persona

- 2.1. Cree un archivo **Persona.php** y defina dentro la clase Persona atendiendo a la siguiente declaración:

```
<?php
class Persona
{
    private $nombre;
    private $apellidos;

    public function __construct($nombre, $apellidos)
    {
        $this->nombre = $nombre;
        $this->apellidos = $apellidos;
    }

    public function getNombre() {
        return $this->nombre;
    }

    public function getApellidos() {
        return $this->apellidos;
    }

    public function getNombreCompleto()
    {
        return $this->nombre." ".$this->apellidos;
    }
}
```

- 2.2. En el archivo *Main.php* crea una nueva instancia y llama a los métodos *getNombreCompleto()* y *GetEdad*.
- 2.3. Agregue una propiedad *fechaNacimiento* a su clase e inicializarla a la fecha actual.
- 2.4. Agregue un método llamado *GetEdad* que devuelva la edad de la persona dependiendo de su edad de nacimiento.
- 2.5. Añade un nuevo constructor que acepte un valor para la propiedad *FechaNacimiento*.
- 2.6. Cree una segunda instancia con el segundo constructor pasándole la fecha.

## Visibilidad y modificadores de acceso

### 3. Clase Student

- 3.1. Cree un archivo ***Student.php*** y defina dentro la clase *Student* atendiendo a la siguiente declaración:

```
class Student {  
  
    public $name;  
    private $graduationYear;  
  
    public function __construct($graduationYear) {  
        $this->graduationYear = $graduationYear;  
    }  
  
    public function getYearsSinceGraduation()  
    {  
  
        $datetime = new DateTime();  
        return $datetime->format("Y") - $this->graduationYear;  
    }  
}
```

- 3.2. En el archivo Main.php añada el siguiente código:

```
$student = new Student(2015);  
$student->name = "Juan";  
printf("Años desde la graduación: ".$student->getYearsSinceGraduation());
```

### 4. Clase Environment

- 4.1. Cree un archivo ***Environment.php*** y defina un atributo llamado *newLine* que sea público y estático.
- 4.2. Añada un método público llamado *getNewLine()* que devuelva el valor de *newLine*.
- 4.3. Añada un método público y estático llamado *getVersion()* que devuelva el número 4.
- 4.4. Pruebe el resultado en el archivo Main.php con el siguiente código:

```
Environment::$newLine = "\r\n";  
$version = Environment::getVersion();  
printf("Environment new line ".Environment::$newLine);  
printf("Environment version es: ".$version);
```

## Herencia y abstracción

5. Defina una clase llamada Vehículo con el siguiente constructor:

```
public function __construct(){  
    printf("__construct Vehiculo\n");  
}
```

- 5.1. Cree otra clase denominada Coche que herede de Vehículo y que llame al constructor padre.
- 5.2. Crea una instancia de Vehículo y otra de Coche en el archivo Main.php para ver el resultado.

6. Clase Message

Atendiendo a la siguiente declaración, siga las instrucciones:

```
class Message{  
  
    private $content;  
  
    public function getContent() {  
        return $this->content;  
    }  
  
    public function setContent($content) {  
        $this->content = $content;  
    }  
}
```

- 6.1. Agregue un método llamado *send* que escriba el valor de la propiedad *content* en la consola.
- 6.2. Implemente una clase derivada denominada *Email* que herede de *Message*.
- 6.3. En archivo Main.php crea una nueva instancia de Email asignada a una variable llamada 'email', asigna un valor a la variable *content* y llame al método *send*.

7. Cree una clase llamada *SpaceStation* que sea abstracta. En esa clase abstracta, agregue un método abstracto llamado *FireLaser*.
  - 7.1. Cree una clase derivada llamada *DeathStar* que implemente el método *FireLaser* para escribir "Pew pew" en la consola seguido de una nueva línea

### Extras

- Probar los ejemplos de la parte teórica que no se hayan visto aquí: interfaces, etc.