Unidad 6

Programación Orientada a Objetos

La programación orientada a objetos (POO, u OOP en lenguaje inglés), es una metodología de programación basada en objetos. Un objeto es una estructura que contiene datos y el código que los maneja.

La estructura de los objetos se define en las clases. En ellas se escribe el código que define el comportamiento de los objetos y se indican los miembros que formarán parte de los objetos de dicha clase. Entre los miembros de una clase puede haber:

- **Métodos**. Son los miembros de la clase que contienen el código de la misma. Un método es como una función. Puede recibir parámetros y devolver valores.
- **Atributos o propiedades**. Almacenan información acerca del estado del objeto al que pertenecen (y por tanto, su valor puede ser distinto para cada uno de los objetos de la misma clase).

A la creación de un objeto basado en una clase se le llama instanciar una clase y al objeto obtenido también se le conoce como **instancia de esa clase**.

Los pilares fundamentales de la POO son:

- **Herencia**. Es el proceso de crear una clase a partir de otra, heredando su comportamiento y características y pudiendo redefinirlos.
- **Abstracción**. Hace referencia a que cada clase oculta en su interior las peculiaridades de su implementación, y presenta al exterior una serie de métodos (interface) cuyo comportamiento está bien definido. Visto desde el exterior, cada objeto es un ente abstracto que realiza un trabajo.
- **Polimorfismo**. Un mismo método puede tener comportamientos distintos en función del objeto con que se utilice.
- **Encapsulación**. El encapsulamiento consiste en definir todas las propiedades y el comportamiento de una clase dentro de esa clase
- .

Clase

Concepto abstracto que denota una serie de cualidades, por ejemplo coche.

Instancia

Objeto palpable, que se deriva de la concreción de una clase, por ejemplo mi coche.

Atributos

Conjunto de características que comparten los objetos de una clase, por ejemplo para la clase coche tendríamos matrícula, marca, modelo, color y número de plazas.

- Nombre de la clase *CamelCase*.
- Nombre de la clase es igual que el nombre del fichero.
- Atributos y funciones con nombres *camelCase*

```php
class ClaseSencilla
{
    // Declaración de una propiedad
    public $var = 'un valor predeterminado';

    // Declaración de un método
    public function mostrarVar() {
        echo $this->var;
    }
    //NOTA: $this hace referenca a este objeto
    //"->" se usa para acceder a métodos y atributos
}
```

Métodos mágicos

- Reciben este nombre los que se ejecutan de forma "mágica". Sin petición explícita.
- Siempre empiezan por doble guión bajo. Los más destacables:
  - `__construct()`: constructor. Se ejecuta al crear un objeto.
  - `__destruct()`: destructor. Se ejecuta al eliminar un objeto.
  - `__toString()`: se ejecuta cuando imprimimos un objeto. Lo convierte a string.

Particularidades

- Para acceder a métodos y atributos usamos **"->"**.
- Para acceder a constantes de clase y métodos estáticos usamos **"::"**
- Para referirnos al propio objeto usamos **$this**

Crear una clase Persona con los siguientes atributos:

- nombre
- apellidos
- edad

Añade una función saludar que escriba Hola

Redefine el constructor y la funcion __toString()

Herencia

La herencia es una de las características más importantes de la POO. Si definimos una serie de atributos y métodos para una clase, al crear una subclase, todos estos atributos y métodos siguen siendo válidos.

A continuación se muestra la implementación de la clase Animal. Uno de los métodos de esta clase es duerme.

Luego crearemos las clases Gato y Ave como subclases de Animal.

Hacer relación ejercicios