PHP POO

PROGRAMACIÓN ORIENTADA A OBJETOS EN PHP

La POO es un paradigma de programación (o técnica de programación) que utiliza objetos e interacciones en el diseño de un sistema.

Como tal, nos enseña un método -probado y estudiado- el cual se basa en las interacciones de objetos (todo lo descrito en el título anterior, Pensar en objetos) para resolver las necesidades de un sistema informático. Básicamente, este paradigma se compone de 4 pilares y diferentes características.

**CARACTERÍSTICAS CONCEPTUALES DE LA POO**

La POO debe guardar ciertas características que la identifican y diferencian de otros paradigmas de programación. Dichas características se describen a continuación: - Abstracción Aislación de un elemento de su contexto. Define las características esenciales de un objeto.

- Encapsulamiento Reúne al mismo nivel de abstracción, a todos los elementos que puedan considerarse pertenecientes a una misma entidad.

- Modularidad Característica que permite dividir una aplicación en varias partes más pequeñas (denominadas módulos), independientes unas de otras.

- Ocultación (aislamiento) Los objetos están aislados del exterior, protegiendo a sus propiedades para no ser modificadas por aquellos que no tengan derecho a acceder a las mismas.

- Polimorfismo Es la capacidad que da a diferentes objetos, la posibilidad de contar con métodos, propiedades y atributos de igual nombre, sin que los de un objeto interfieran con el de otro.

- Herencia Es la relación existente entre dos o más clases, donde una es la principal (padre) y otras son secundarias y dependen (heredan) de ellas (clases “hijas”), donde a la vez, los objetos heredan las características de los objetos de los cuales heredan.

- Recolección de basura Es la técnica que consiste en destruir aquellos objetos cuando ya no son necesarios, liberándolos de la memoria.

**CLASES O CLASES CONCRETAS**

Una clase es un modelo que se utiliza para crear objetos que comparten un mismo comportamiento, estado e identidad.

**Objeto**

Es una entidad provista de métodos o mensajes a los cuales responde (comportamiento); atributos con valores concretos (estado); y propiedades (identidad).

**Método**

Es el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer.

**Propiedades y atributos**

Las propiedades y atributos, son variables que contienen datos asociados a un objeto.

**EMPECEMOS CON POO BÁSICA**

Toda clase consta de la palabra clave class seguido del nombre de la clase y un bloque de código entre llaves. Dentro del bloque de código se pueden crear tres tipos de bloques básicos:

· Constantes

· Variables

· Métodos

Una vez creadas dentro de las llaves, tanto la constante, como la variable, como la función pertenecen a la clase, y para ser utilizadas hay que acceder a través de la clase.

**Reglas de Estilo sugeridas**

Utilizar CamelCase para el nombre de las clases. La llave de apertura en la misma línea que el nombre de la clase, permite una mejor legibilidad del código.

**Herencia de Clases**

Los objetos pueden heredar propiedades y métodos de otros objetos. Para ello, PHP permite la “extensión” (herencia) de clases, cuya característica representa la relación existente entre diferentes objetos. Para definir una clase como extensión de una clase “padre” se utiliza la palabra clave extends.

**Declaración de clases abstractas**

Las clases abstractas son aquellas que no necesitan ser instanciadas pero sin embargo, serán heredadas en algún momento. Se definen anteponiendo la palabra clave abstract.

Este tipo de clases, será la que contenga métodos abstractos (que veremos más adelante) y generalmente, su finalidad, es la de declarar clases “genéricas” que necesitan ser declaradas, pero a las cuales, no se puede otorgar una definición precisa (No se pueden instanciar), de eso, se encargarán las clases que la hereden).

**Declaración de Clases finales En PHP**

PHP desde su versión 5.1 incorpora clases finales que no pueden ser heredadas por otra. Se definen anteponiendo la palabra clave final.

**OBJETOS E INSTANCIAS**

Una vez que las clases han sido declaradas, será necesario crear los objetos y utilizarlos, aunque hemos visto que algunas clases, como las clases abstractas son solo modelos para otras, y por lo tanto no necesitan instanciar al objeto.

**Instanciar una clase**

Para instanciar una clase, solo es necesario utilizar la palabra clave new. El objeto será creado, asignando esta instancia a una variable (la cual, adoptará la forma de objeto). Lógicamente, la clase debe haber sido declarada antes de ser instanciada.

**Reglas para la instanciación de los objetos**

Para una mejor legibilidad y manejo de las clases, se recomienda utilizar nombres de variables (objetos) descriptivos, siempre con guion bajo al comenzar, la primera letra debe ser en minúscula, y la siguiente palabra en mayúscula. Por ejemplo, si el nombre de la clase es nombreClasecomo variable utilizar $\_nombreClase. Esto permitirá una mayor legibilidad del código.

**Definición de atributos o propiedades en PHP**

Las propiedades representan ciertas características del objeto en sí mismo. Se definen anteponiendo la palabra clave var al nombre de la variable (propiedad). No es necesario utilizar la palabra reservada varpara la definición de la variable, pues PHP la reconoce por defecto.

Las propiedades pueden gozar de diferentes características como, por ejemplo, la visibilidad: pueden ser públicas, privadas o protegidas.

**Niveles de acceso**

1. Propiedades públicas

Las propiedades públicas se definen anteponiendo la palabra clave public al nombre de la variable. Éstas, pueden ser accedidas desde cualquier parte de la aplicación, sin restricción.

2. Propiedades privadas

Las propiedades privadas se definen anteponiendo la palabra clave private al nombre de la variable. Éstas solo pueden ser accedidas por la clase que las definió.

3. Propiedades protegidas

Las propiedades protegidas pueden ser accedidas por la propia clase que la definió, así como por las clases que la heredan, pero no, desde otras partes de la aplicación. Éstas, se definen anteponiendo la palabra clave protected al nombre de la variable

4. Propiedades estáticas Las propiedades estáticas representan una característica de “variabilidad” de sus datos, de gran importancia en PHP. Una propiedad declarada como estática, puede ser accedida sin necesidad de instanciar un objeto y su valor es estático (es decir, no puede ser modificada para cada objeto, es como una variable global para todas las instancias que se crean de ese objeto). Ésta, se define anteponiendo la palabra clave static al nombre de la variable.