

RESUMEN: PROGRAMACIÓN ORIENTADA A OBJETOS EN PHP

La POO presenta características que la identifican de otros paradigmas de programación, como la abstracción, encapsulamiento, modularidad, ocultamiento, polimorfismo, herencia, y la recolección de basura. La POO es un paradigma de programación basado en el uso de objetos, instancias, métodos y clases para las interacciones en el diseño de un sistema.

Las clases son utilizadas para crear objetos que comparten un mismo comportamiento. Para crear una clase se utiliza la palabra clave `class`, seguida de un nombre y un bloque de código delimitado por llaves.

Dentro de las clases se crean objetos, métodos y sus propiedades. Los objetos son entidades provistas de métodos o mensajes a los cuales responden, atributos con valores concretos y propiedades. El objeto será creado, asignando esta instancia a una variable

Un método es un procedimiento asociado con un objeto, es un fragmento de código que puede tomar un parámetro de entrada, procesarlo y devolver un valor. Para definir un método se utiliza las palabras claves según el nivel de seguridad deseado: `public`, `protected` o `private`, seguido de una declaración de nombre.

Los atributos y propiedades son variables que contienen datos asociados a un objeto, al igual que los métodos las propiedades tienen cuatro niveles de acceso, los cuales se diferencian por permitir al acceso a los datos de las propiedades de distintas formas, los niveles son: `public`, `private`, `protected` y `static`. Para acceder a las propiedades de un objeto dependerá del ámbito y su nivel de acceso. Para acceder a una propiedad no estática dentro de la clase se utiliza `$this->variable;`. Cuando la variable es estática se utiliza `self` o `parent` seguido de `::`. Para acceder a una variable desde el exterior de la clase: `$objeto->variable`. Para acceder a una propiedad pública y estática el objeto no necesita ser instanciado, utilizando la siguiente sintaxis: `Clase::$variable_estatica`. Las constantes son otro tipo de propiedades de una clase, estas contienen su valor de forma permanente y sin cambiarlo, el acceso a las constantes es igual que las otras propiedades.

Otra de las características de POO es la herencia. Los objetos pueden heredar propiedades y métodos de otros objetos. Para ello, PHP permite la herencia de clases, cuya característica representa la relación existente entre diferentes objetos. La herencia define una clase como padre y otra como hijo, para definir la clase padre se utiliza la palabra clave `extends`. Cuando se hereda de una clase abstracta, todos los métodos definidos como `abstract` en la definición de la clase Padre deben ser redefinidos en la clase hijo.

Dentro de los métodos tenemos los métodos mágicos, estos métodos tienen una funcional pre-definida por PHP, lo que ahorra una gran cantidad de código.

Por ejemplo:

`__construct()` se invoca de manera automática al instanciar un objeto, su función es la de ejecutar cualquier inicialización que el objeto necesite antes de ser utilizado.

`__destruct()` libera la memoria al objeto cuando ya no es referenciado.

Otros métodos mágicos que nos ofrece PHP son: `__call`, `__callStatic`, `__get`, `__set`, `__isset`, `__unset`, `__sleep`, `__wakeup`, `__toString`, `__invoke`, `__set_state` y `__clone`.

Una interfaz es un conjunto de métodos abstractos y de constantes cuya funcionalidad es la de determinar el funcionamiento de una clase, es decir, funciona como un molde o como una plantilla. Al ser sus métodos abstractos estos no tienen funcionalidad alguna, sólo se definen su tipo, argumento y tipo de retorno.

Para implementar una interface es necesario que la clase que quiera hacer uso de sus métodos utilice la palabra reservada `implements`. La clase que la implemente, de igual modo debe sobrescribir los métodos y añadir su funcionalidad.

La agregación y la composición en PHP se utiliza para mapear las relaciones que utilizamos en los diagramas UML. La Agregación es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye y la Composición es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye.