

PROGRAMACIÓN ORIENTADA A OBJETOS EN PHP (Resumen)

Para crear un sistema puede utilizarse la programación orientada a objetos (POO), esto es el utilizar el extraer un objeto real para resolver un problema. Este paradigma cuenta con cuatro puntos importantes los cuales son: encapsulamiento (tiene en un mismo nivel a todo lo que pueda pertenecer una entidad), herencia (las clases padres heredan sus características a las clases hijas), polimorfismo (puede haber métodos o propiedades con el mismo nombre) y abstracción (asilar ay definir un elemento de su contexto) además de modularidad (se divide la aplicación en módulos más pequeños), aislamiento (los niveles de acceso restringen las modificaciones a quienes no tengan derecho a acceder a estas y permite a quien si los tiene) y recolección de basura (destruye objetos cuando y no se necesitan).

Para la representación en código de un objeto real (abstracción) se utiliza algo denominado clase, esta clase además de crear los objetos contiene comportamientos y propiedades del mismo. Esta característica de crear objetos con atributos y métodos mejora la forma de programar ya que al tener que agregar más características a un objeto, solo es necesario hacerlo una vez y todas las instancias de este lo tendrán.

Para explicar algunos conceptos básicos de POO, podría empezarse con la definición de clase, esto como explica el texto sería una especie de molde del cual se pueden extraer copias u objetos, otra diferencia en POO es que las variables son propiedades, las funciones son métodos, las clases son globales y no se usa la palabra reservada “global” y estas son una agrupación de variables, funciones y constantes. Una propiedad de POO como ya se menciona es la herencia, que no es más que la capacidad de heredar propiedades y métodos a otras clases, esto se hace utilizando la palabra reservada “extends” antes del nombre de la clase que quiera heredar. Otro tipo de clase son las clases finales que solo llevan la palabra “final” antes de la palabra “class” y con esto, estas clases no pueden ser heredadas por otra.

Otro concepto clave son los objetos o instancias que no son más que la instancia de una clase, para esto se usa la palabra clave “new”, las reglas para nombrar a un objeto son igual que para una variable, inicia con una letra o guion bajo y si tiene varias palabras solo va en mayúscula la primera letra del inicio de la siguiente palabra. Para utilizar los niveles de acceso en alguna propiedad solo se antepone la palabra clave del nivel de acceso que desee incorporar a dicha propiedad, existen tres, “public”, “protected” y “private”, la primer palabra “public”, hace que la propiedad pueda ser accedida desde cualquier parte de nuestro programa, por otra parte “private” permite que solo la clase que la definió pueda acceder a ella, “protected” es un nivel de acceso que permite que no solo la clase que la definió pueda acceder a ella sino también las clases que la hereden, pero no de otras partes del programa. Las propiedades estáticas no pueden modificarse a través de los objetos, pero pueden ser accedida sin necesidad de instanciar un objeto mientras que su valor es estático, se accede a esta escribiendo el nombre de la clase seguido de doble dos puntos y el nombre de la variable con el símbolo “\$”. Además para acceder a una propiedad dentro de un objeto se puede usar la palabra reservada “this” (sin embargo la variable no llevara adelante \$), también para acceder a una propiedad de un objeto se usa “->”. Otro tipo de propiedad son las constantes que, como su nombre lo indica, estas mantienen su valor de forma permanente, estas solo pueden tener una visibilidad y es la publica además de no ser creadas dentro de clases, se accede a esta como a las demás propiedades.

Las funciones en la POO se llaman métodos, que pertenecen a una clase, para declarar uno se escribe la palabra clave function antes del nombre del método, seguido por un par de paréntesis de apertura y cierre (en estos puede recibir parámetros si así se requiere) y llaves que encierren el algoritmo. Los métodos pueden llevar los mismos niveles de acceso que las propiedades, también existe un método abstracto que solo puede ser definido en clases abstractas. Los métodos mágicos nos pueden ayudar a ahorrar mucho código, uno de estos es el método mágico “__construct()”, este ejecuta cualquier inicialización que el objeto necesite antes de ser utilizado, se puede mencionar que __destruct() es otro método mágico, este libera la memoria del objeto cuando ya no es referenciado, existen otros métodos

mágicos tales como: `__call`, `__callStatic`, `__get`, `__set`, `__isset`, `__unset`, `__sleep`, `__wakeup`, `__toString`, `__invoke`, `__set_state` y `__clone`.

Para determinar el funcionamiento de una clase, como será esa plantilla, se podría crear un conjunto de métodos abstractos y constantes, a esto se le conoce como interfaz; para implementar una interfaz previamente creada (usando `interface` antes del nombre de la interfaz y en lugar del habitual “`class`”), utilizamos la palabra reservada “`implements`” (recuerda sobrescribir los métodos abstractos).

Una ventaja de la POO es que en PHP se pueden crear las relaciones que se hacen en los diagramas UML, como por ejemplo lo son la agregación (es parte de) y composición (se compone o es parte).