

Tarea 1

Programación Orientada a Objetos en PHP

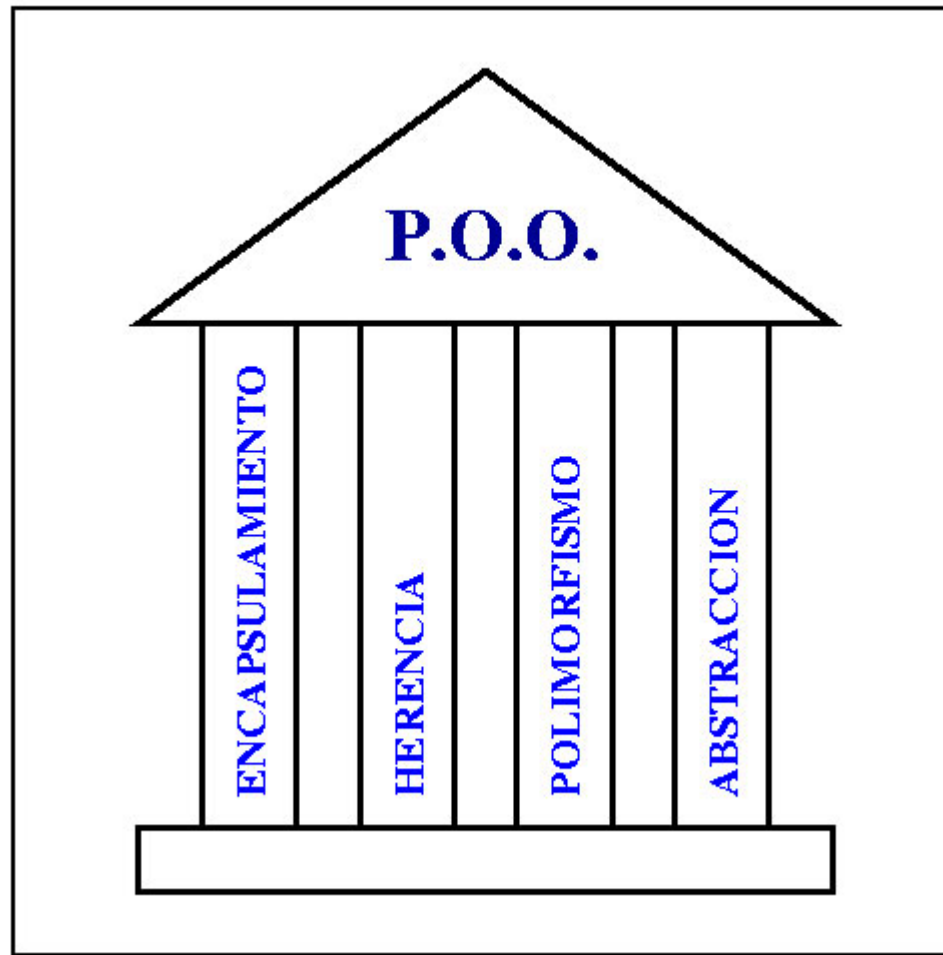
**Universidad Politécnica de Victoria
Ingeniería en Tecnologías de la Información
Tecnología y Aplicaciones Web**

Héctor Alán De La Fuente Anaya

¿Qué es POO?

La POO es un paradigma de programación (o técnica de programación) que utiliza objetos e interacciones en el diseño de un sistema. Básicamente, este paradigma se compone de 4 pilares y diferentes características que veremos a continuación.

Pilares de P.O.O.



Pilares de la POO

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.

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 y otras son secundarias y dependen de ellas, donde a la vez, los objetos heredan las características de los objetos de los cuales heredan.

Otras características de POO

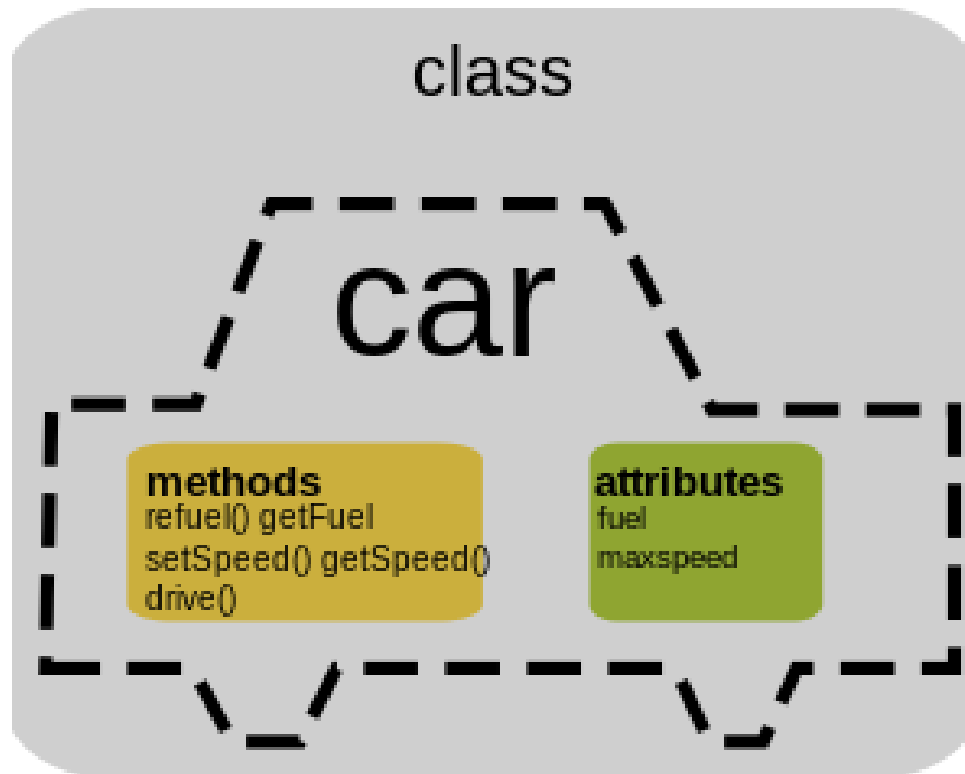
Recolección de basura: Es la técnica que consiste en destruir aquellos objetos cuando ya no son necesarios, liberándolos de la memoria.

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

Ocultación: 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.

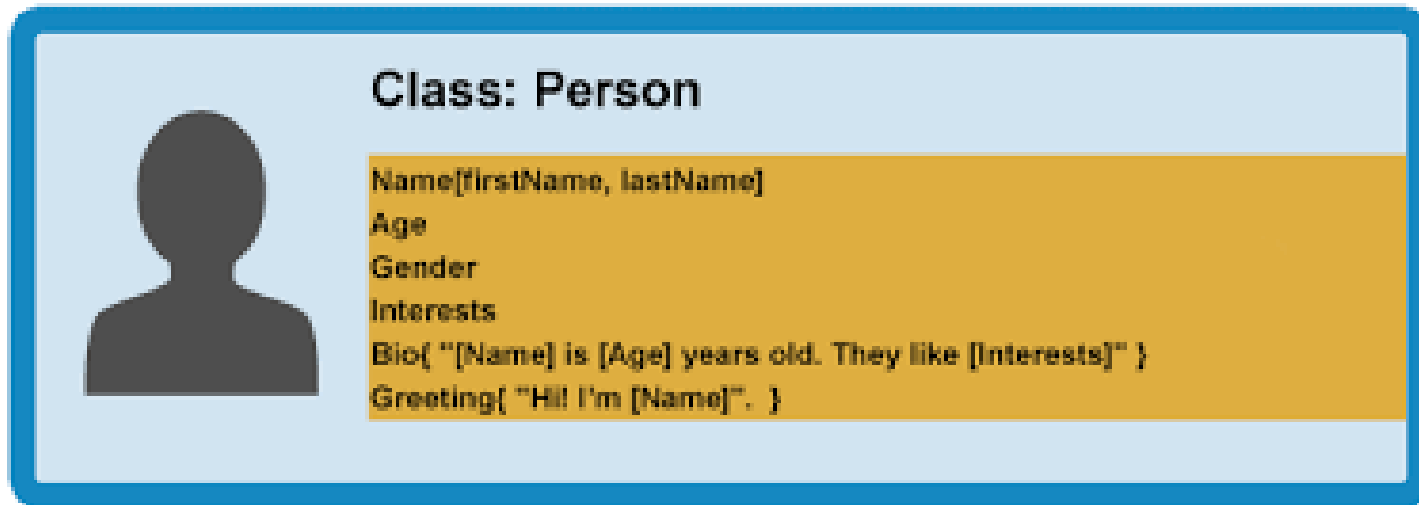
Clases

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.

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`.

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).

Clases finales

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

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 nombreClase como variable utilizar `$_nombreClase`. Esto permitirá una mayor legibilidad del código.

Niveles de acceso

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.

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.

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.

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.

Acceder a las propiedades de un objeto

Acceso a variables desde el ámbito de la clase

Se accede a una propiedad no estática dentro de la clase, utilizando la pseudo-variable `$this` siendo esta pseudo-variable una referencia al objeto mismo, se debe tener en cuenta que la variable que se llamara no llevara adelante el \$.

Acceso a variables desde el exterior de la clase

Se accede a una propiedad no estática con la siguiente sintaxis: `$objeto → variable` Nótese además, que este acceso dependerá de la visibilidad de la variable.

Constantes

Otro tipo de “propiedad” de una clase, son las constantes, aquellas que mantienen su valor de forma permanente y sin cambios. A diferencia de las propiedades estáticas que pueden ser declaradas dentro de una clase, las constantes solo pueden tener una visibilidad pública y no deben ser creadas dentro de las clases. El acceso a constantes es exactamente igual que al de otras propiedades.

Métodos

La forma de declarar un método es anteponiendo la palabra clave `function` al nombre del método, seguido por un par paréntesis de apertura y cierre y llaves que encierren el algoritmo. Al igual que cualquier otra función en PHP, los métodos recibirán los parámetros necesarios indicando aquellos requeridos, dentro de los paréntesis.

Métodos públicos, privados, protegidos y estáticos

Los métodos, al igual que las propiedades, pueden ser públicos, privados, protegidos o estáticos. La forma de declarar su visibilidad tanto como las características de ésta, es exactamente la misma que para las propiedades.

Interfaces

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 tiene 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.

Gracias por su atención