

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE

- ¿Qué es el Polimorfismo?
- ¿Qué es la Herencia?
- Herencia Simple

¿QUÉ ES POLIMORFISMO?

En la Programación Orientada a Objetos, el polimorfismo es uno de los pilares básicos, para poder entenderlo es necesario tener los conceptos claros tanto de la POO, como de la herencia.

El término *polimorfismo* tiene origen en las palabras *poly* (muchos) y *morfo* (formas), y aplicado a la programación, hace referencia a que los objetos pueden tomar diferentes formas. ¿Pero qué significa esto?

Pues bien, se refiere a que objetos de diferentes clases pueden ser accedidos utilizando la misma interfaz, mostrando un comportamiento distinto (tomando diferentes formas) según cómo sean accedidos.

En lenguajes de programación como Python, que tiene tipado dinámico, el polimorfismo va muy relacionado con el *duck typing*.

El polimorfismo es una relajación del sistema de tipos, de tal manera que una referencia a una clase (atributo, parámetro o declaración local, o elemento de un vector) acepta direcciones de objetos de dicha clase, y de sus clases derivadas. El polimorfismo y la herencia son dos conceptos estrechamente ligados. Conseguimos implementar el polimorfismo en jerarquías de clasificación que se dan a través de la herencia. Por ejemplo: tenemos una clase vehículo y de ella dependen varias clases hijas como coche, moto, autobús, entre otros.

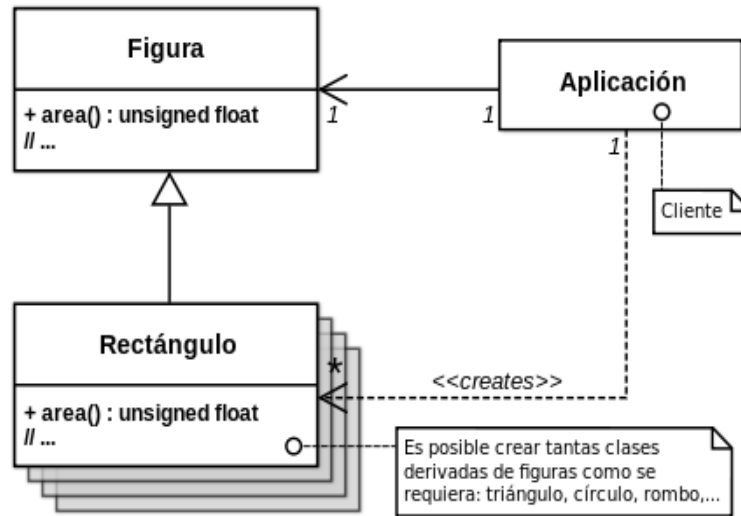


Ilustración 1 Ejemplo de polimorfismo

Existen 3 tipos de polimorfismo:

- **Sobrecarga:** es cuando existen funciones con el mismo nombre, con funcionalidad similar; en clases que son completamente independientes una de la otra.
- **Paramétrico:** existen funciones con el mismo nombre, pero se usan diferentes parámetros (nombre o tipo). Se selecciona el método dependiendo del tipo de datos que se mande.
- **Inclusión:** es cuando se puede llamar a un método sin tener que conocer su tipo, así no se toma en cuenta los detalles de las clases especializadas, utilizando una interfaz común.

¿CUÁL ES LA HERENCIA?

Cuando se diseñan diagramas de clases, en muchas ocasiones se crearán clases que tendrán atributos o métodos similares. Por ejemplo: la clase Perro y la clase Gato. Por lo general, ambas tendrán casi el mismo código, atributos y métodos. En lugar de escribir el mismo código dos veces, usamos un concepto llamado herencia.

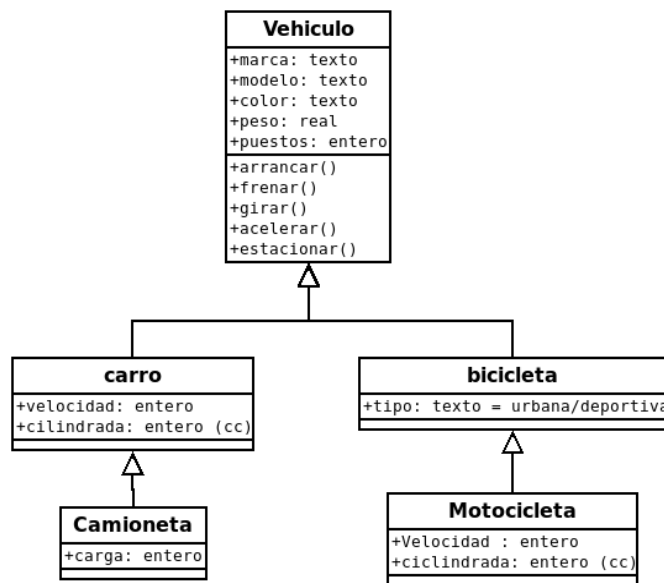
La herencia es uno de los conceptos que permite que las clases tengan reutilización de código dentro de la programación. Cuando se tienen dos o más clases que usan código similar, generalmente quieren crear lo que se llama una "superclase". Las clases que heredarán todo el código dentro de la superclase se conocen como "subclases".

Una excelente manera de pensar en la herencia es imaginar dos clases: una son los padres, y la otra clase que serán sus hijos. Los padres transmiten genes a sus hijos, que son heredados y ayudan a definir los rasgos con los que éste nacerá. En este sentido, la herencia en POO obra de la misma manera, donde la subclase hereda todos los atributos y métodos dentro de la superclase. En lugar de escribir los mismos atributos y métodos dos veces para las clases, podemos heredar una, y sólo necesitamos escribir el código una vez.

HERENCIA SIMPLE

Esta tiene lugar cuando una clase hija hereda los atributos y métodos de una única clase padre.

La herencia es la transmisión del código entre unas clases y otras. Para soportar un mecanismo de herencia tenemos dos clases: la clase padre, y la(s) clase(s) hija(s). La clase padre es la que transmite su código a las clases hijas.





HERENCIA DE CLASES

La diferencia entre herencia y polimorfismo es que la herencia está relacionada con clases, y el polimorfismo con métodos.