

MATERIA: PROGRAMACIÓN ORIENTADA A OBJETOS

SEMANA 6

TEMAS SEMANA 6:

- a) Herencia
- b) Interfaces
- c) Polimorfismo

- **¿Qué diferencia existe entre la herencia simple que maneja Java y la herencia múltiple de otros lenguajes como C++?**

La herencia es uno de los 4 pilares fundamentales que ampara la programación orientada a objetos y todo lenguaje de programación que maneje este paradigma deberá poseer la capacidad de brindar esta característica, la cual consiste en heredar funcionalidad de clases existentes a subclases.

Java no es la excepción y es capaz de proveer esta característica, pero bajo el enfoque de herencia simple la cual es capaz de heredar únicamente de una sola clase padre o superclase a las subclases que están heredando. Tal y como se ve en la siguiente imagen.

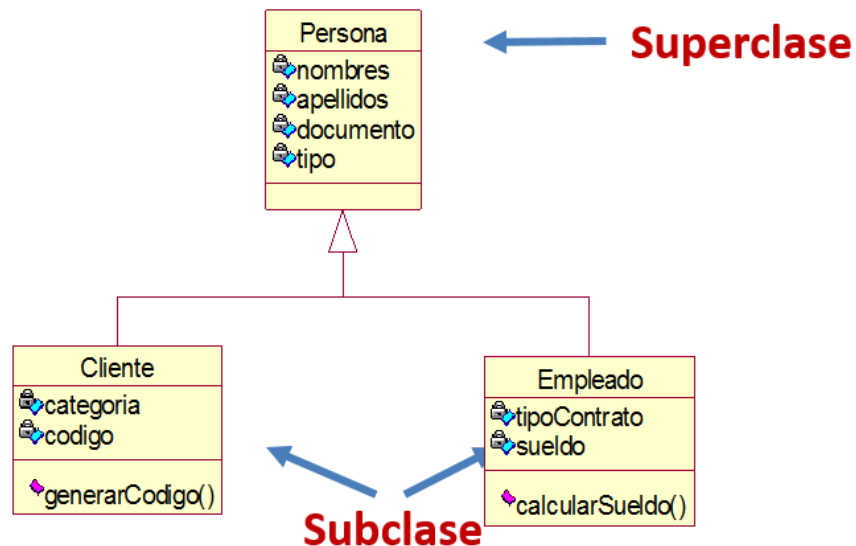


Ilustración 1. Herencia simple

Por otro lado, existe la herencia múltiple que manejan otros lenguajes como C++ solo que se maneja desde otro enfoque, es decir, es posible heredar de dos o más clases padres, como se muestra a continuación.

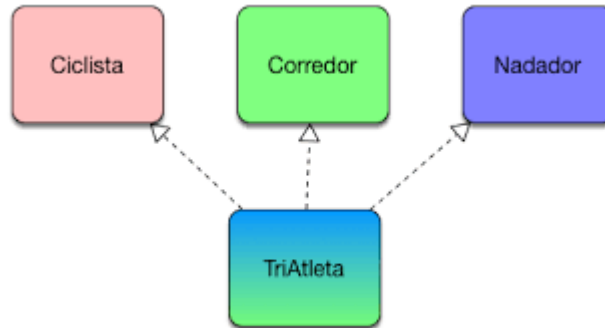


Ilustración 2. Herencia múltiple.

En la imagen anterior se tiene una herencia múltiple de tres clases padre que pasan sus atributos a una subclase.

- **¿Qué es el polimorfismo?**

El polimorfismo forma parte de los 4 pilares del paradigma orientado a objetos que explicado de manera sencilla lo definimos como la capacidad que tienen los métodos de responder de diferente manera a un mismo mensaje, es importante no confundirlo con la sobrecarga de métodos.

A continuación, muestro un ejemplo del esquema básico que maneja el polimorfismo.

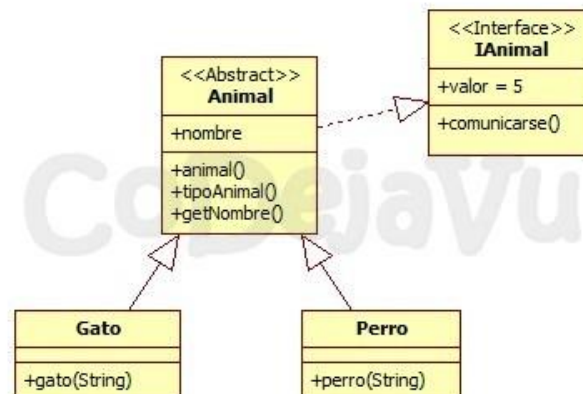


Ilustración 3. Plimorfismo.

En la imagen que observamos nos damos cuenta que se hace uso de la entidad interface que sirve como intermediaria en el manejo del polimorfismo.

- **¿En java es posible manejar la herencia múltiple?**

No, Java solo permite la herencia simple en la cual se entiende que una o mas clase únicamente pueden heredar de una clase padre. Sin embargo, hay una manera indirecta de poder alcanzar una funcionalidad similar por medio de las interfaces. A continuación, te muestro un ejemplo de una inteface.

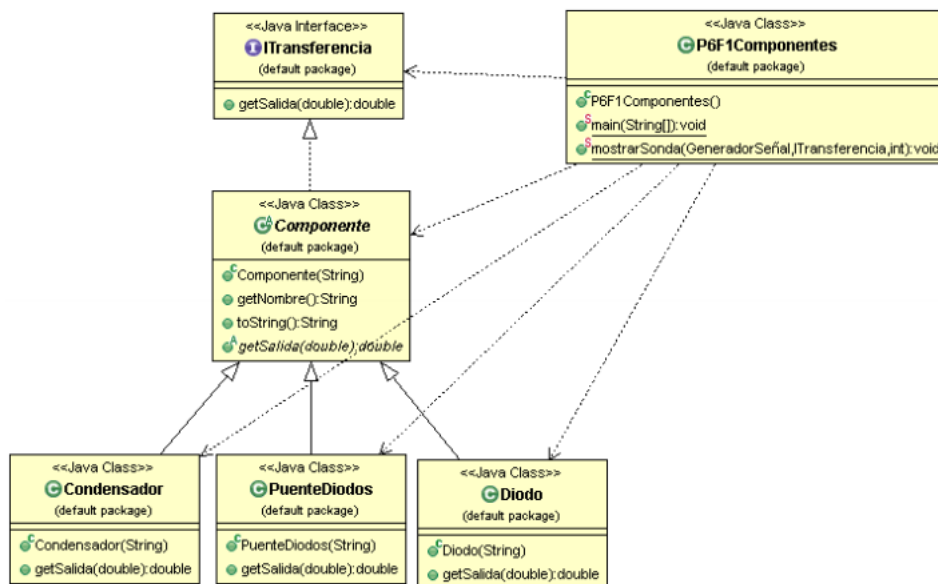


Ilustración 4. Interfaces

En la imagen anterior tenemos el concepto de interfáces que consisten en una colección de datos abstractos y propiedades constantes que serviría de intermediaria para poder tener acceso a métodos y atributos de otras clases, no es como tal herencia múltiple pero si ofrece una alternativa funcional.

- **¿Qué significa la palabra polimorfismo?**

En programación orientada a objetos, el polimorfismo se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía. En la

siguiente imagen podrás Observar que las clases definidas manejan métodos que se llaman exactamente igual pero que en cada objeto, se comportarán de manera diferente.

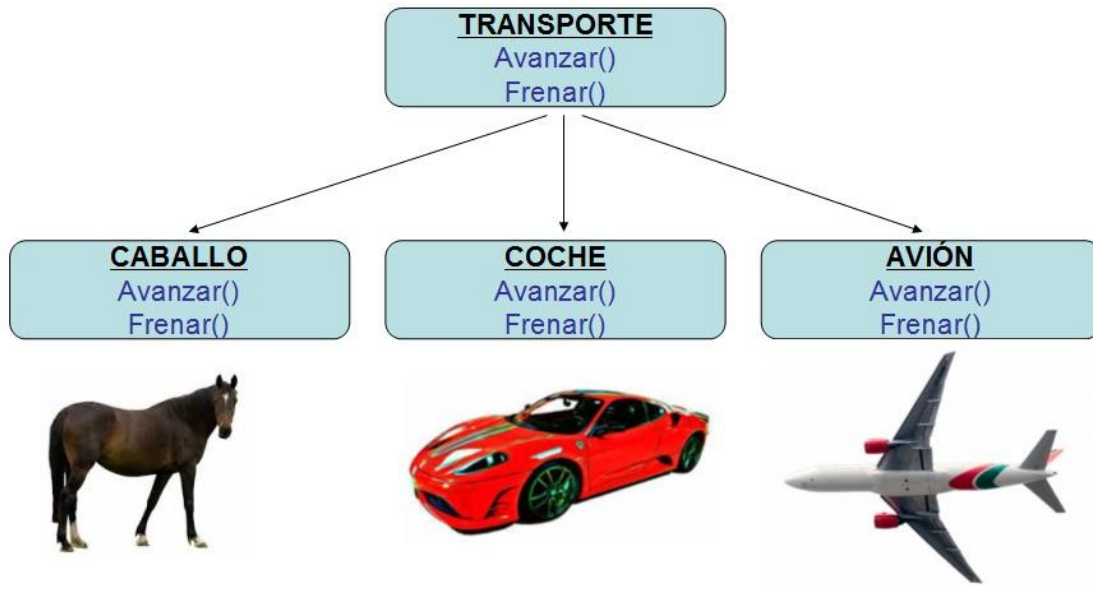


Ilustración 5. Polimorfismo