

**UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”**



**NOMBRE:** Luis Calle

**NRC:** 16419

**FECHA:** 30/06/24

## **RESUMEN:**

En el siguiente informe se redactará lo aprendido al leer sobre los temas correspondientes al silabo de la materia Programación Orientada A Objetos (POO). Los temas a tratar serán Herencia y Polimorfismo, temas importantes para programación.

## **INTRODUCCION:**

Para entender bien lo que es POO se deben entender varias cosas, entre ellas se debe tener conocimiento de lo que es Herencia y Polimorfismo, temas bastante importantes ya que ambas van conectadas y se puede decir que tienen bastante relación entre si.

Por ello se va a dar la definición de cada uno de estos términos y como se aplica en la programación orientada a objetos.

## **DESARROLLO:**

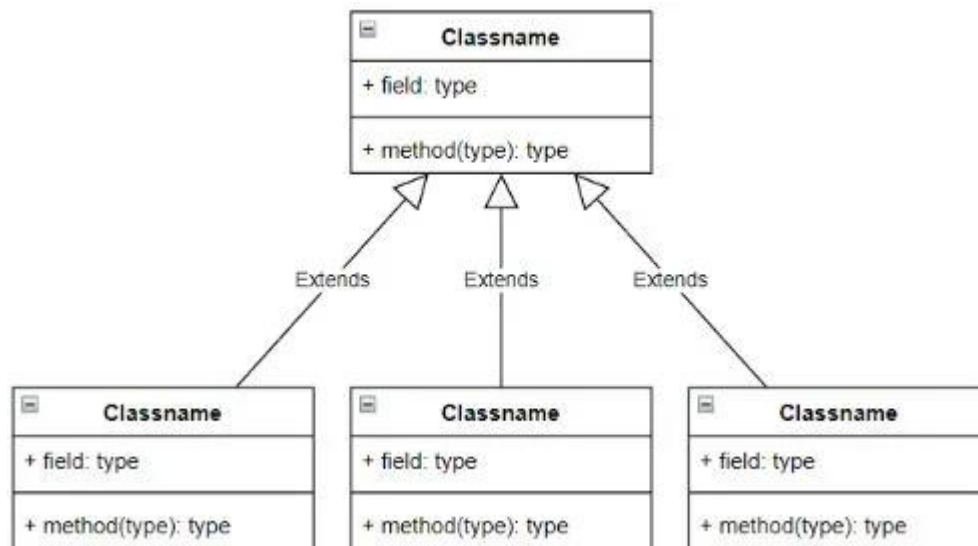
### **HERENCIA:**

Cuando hablamos de herencia en programación no nos referimos precisamente a que algún familiar lejano nos ha podido dejar una fortuna, ya nos gustaría. En realidad se trata de uno de los pilares fundamentales de la programación orientada a objetos. Es el mecanismo por el cual una clase permite heredar las características (atributos y métodos) de otra clase.

La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.

En Java, cada clase solo puede derivarse de otra clase. Esa clase se llama superclase, o clase padre. La clase derivada se llama subclase o clase secundaria.

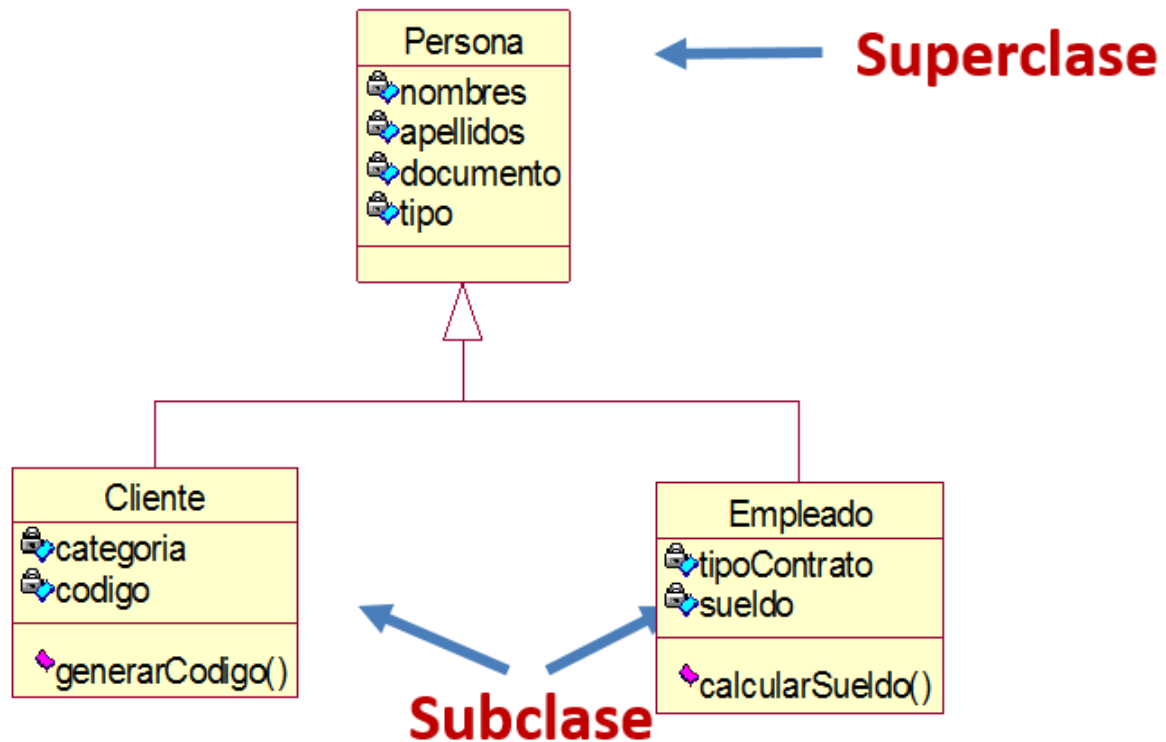
Utiliza la palabra clave extends para identificar la clase que extiende su subclase. Si no declara una superclase, su clase amplía implícitamente la clase Object. El objeto es la raíz de todas las jerarquías de herencia; Es la única clase en Java que no se extiende de otra clase.



## POLIMORFISMO:

En programación orientada a objetos, polimorfismo es la capacidad que tienen los objetos de una clase en ofrecer respuesta distinta e independiente en función de los parámetros (diferentes implementaciones) utilizados durante su invocación. Dicho de otro modo el objeto como entidad puede contener valores de diferentes tipos durante la ejecución del programa.

En JAVA el término polimorfismo también suele definirse como 'Sobrecarga de parámetros', que así de pronto no suena tan divertido pero como veremos más adelante induce a cierta confusión. En realidad suele confundirse con el tipo de poliformismo más común, pero no es del todo exacto usar esta denominación.



## CONCLUSIONES:

Se concluyó que ambos conceptos van relacionados entre sí y es fundamental que un programador tenga estos conocimientos para poder trabajar de manera correcta y óptima sin ningún problema.

Es interesante saber cómo ambos conceptos tienen que ver uno con el otro y además se entendió cómo y cuándo aplicar herencia y polimorfismo.

## BIBLIOGRAFÍA:

Blog IfGeekThen. (2024). IfgeekthenNTTDATA.

<https://ifgeekthen.nttdata.com/s/post/herencia-en-programacion-orientada-objetos-MCPV3PCZDNBFHSROCCU3JMI7UIJQ?language=es>

Blog IfGeekThen. (2024). IfgeekthenNTTDATA.

<https://ifgeekthen.nttdata.com/s/post/polimorfismo-en-java-programacion-orientada-objetos-MCIU2TZFKR6FFIJMDQQASC7CU75I?language=es>

