# Java Polimorfismo: Entendiendo herencia e interfaces

## Conclusión 1: Introducción a herencia

En esta aula comenzamos a hablar de herencia y aprendimos:

* Cuáles son los problemas que la herencia resuelve.
* Cómo usar la herencia en Java a través de la palabra llave extends
* Al heredar la clase hija gana todas las características (atributos) y todas las funcionalidades (métodos) de la clase madre.
* Conocimos el primer beneficio de la herencia: La reutilización de código.

En la próxima clase veremos más detalles sobre la herencia como la palabra llave super, protected y la sobreescritura de métodos.

|  |
| --- |
|  |

## Conclusión 2: Super y reescrita de Métodos (intercambiar modulo 5 con 3)

* que la clase madre es llamada de super o base class.
* que la clase hija también es llamada de sub class.
* como aumentar la visibilidad de un miembro (atributo, método) a través de protected.
* cómo acceder o llamar un miembro (atributo, método) a través de super.
* cómo redefinir un método a través de la sobreescritura. En la próxima clase veremos un nuevo beneficio de la herencia, el Polimorfismo. ¡Aguarda!

|  |
| --- |
|  |

## Conclusión 3: Entendiendo Polimorfismo

En esta clase aprendimos que:

* los objetos no cambian de tipo;
* la referencia puede cambiar, y ahí es donde entra el polimorfismo;
* el polimorfismo permite utilizar referencias más genéricas para comunicarse con un objeto;
* el uso de referencias más genéricas permite desacoplar sistemas.

En el siguiente vídeo, hablaremos sobre cómo se comportan los constructores en la herencia.

|  |
| --- |
|  |

## Conclusión 4: Herencia y el uso de constructores

## Conclusión 5: Clases y métodos abstractos

## Conclusión 6: Interfaces

## Conclusión 7: Practicando herencia e interfaces

## Anexos:

## Código:

* Extends: extiende clases a otros grupos de datos
* Super: da acceso a los métodos de la clase padre

## Vocabulario:

* Sobre-escritura de método:
* Paquete
* Polimorfismos: