

Herencia en Java: Reutilizacion de código



Dyl



¿QUÉ ES?

La herencia es la forma en la que yo puedo crear nuevas clases a partir de clases ya existentes.



¿POR QUÉ USAR HERENCIA?

La herencia evita reescribir código, y genera que la lógica del negocio sea más lógica al momento de su implementación,

```
class Vehiculo {  
    void mover() { System.out.println("El vehículo se mueve"); }  
}  
  
class Carro extends Vehiculo {  
    void acelerar() { System.out.println("El carro acelera"); }  
}
```

COMO SE HACE HERENCIA EN JAVA

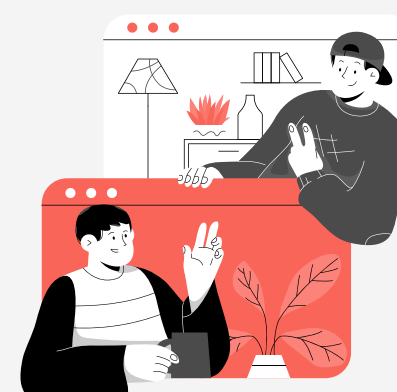
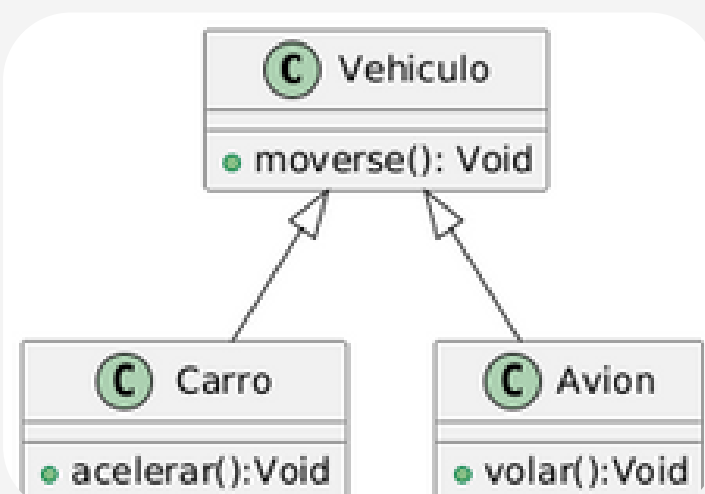
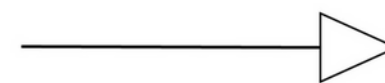
Es bastante sencillo, es necesario una clase base como **Vehiculo**, si deseamos crear una clase que herede de vehículo como **Carro**, se usa la palabra reservada **extends**, seguido del **nombre de la clase padre** al momento de definir la clase hija.

Entonces, ahora Carro hereda los métodos y atributos de Vehículo, se puede hacer con cuantas clases se desee, por ejemplo una nueva clase **Avión**.

```
class Avion extends Vehiculo {  
    void volar() { System.out.println("El avion vuela"); }  
}
```

REPRESENTACIÓN UML

Súper, ya tenemos dos clases hijas y una clase padre, la herencia implica una jerarquía, por lo que en UML se usa una flecha con punta triangular vacía para representar la herencia entre clases.



Es importante usar la representación correcta, nos permite identificar a través del diagrama que relación existe entre las clases.

¿ENTONCES SOLO PUEDO HEREDAR DE UNA CLASE?

En Java no existe la herencia múltiple como en otros lenguajes, se pueden usar interfaces para simular una herencia múltiple, pero es recomendable ajustar el modelo a las reglas de Java.



TIPS

Si en un enunciado se mencionan objetos que comparten características, o pertenecen a una misma categoría, podemos empezar a pensar en aplicar herencia.

