



DigitalHouse >
Coding School

Composición y herencia



**Certified Tech
Developer**
The Ultimate Degree

“

De manera general, una definición para composición y herencia sería que son dos mecanismos para reutilizar la funcionalidad, es decir, no ser repetitivo ni escribir código innecesario.

”



Herencia

La herencia siempre se ha considerado una herramienta básica para extender y reutilizar la funcionalidad basada en los atributos y métodos de una clase. Por ejemplo:

```
public class Animal{  
}  
public class Cachorro extends Animal{  
}  
public class Gato extends Animal{  
}
```

Composición

En la composición, en lugar de codificar un comportamiento de forma estática, como se hace con la herencia, definimos pequeños comportamientos predeterminados y la usamos para declarar comportamientos más complejos. Por ejemplo:

```
public class Sistema {  
  
    Persona persona = new Persona();  
  
}  
public class Persona{  
}
```

Ventajas y desventajas de la herencia

Ventajas	Desventajas
<p>Captura lo común y lo aísla de lo diferente. La herencia se ve directamente en el código, incluso debido a su naturaleza estática.</p>	<p>Encapsulación débil y acoplamiento estrecho, donde el cambio de una superclase puede afectar a todas las subclases.</p>
<p>Permite crear una estructura jerárquica de clases cada vez más especializada. Por lo tanto, no hay que empezar desde cero para especializar una clase existente.</p>	<p>A veces un objeto debe ser de una clase diferente en diferentes momentos, lo que no es posible con la herencia, ya que tiene una relación estática.</p>

Ventajas	Desventajas
<p>El comportamiento se puede elegir en tiempo de ejecución en lugar de estar vinculado en tiempo de compilación.</p>	<p>El software es muy dinámico y parametrizado y es más difícil de entender que el software más estático.</p>
<p>Los objetos que fueron instanciados y que están contenidos en la clase que los instancia, se accede sólo a través de su interfaz, siguiendo así el principio de programación para una interfaz y no para una implementación.</p>	

¿Cuándo usarlos?

En general, siempre se prefiere usar la composición sobre la herencia, sin embargo, podemos definir algunas reglas para identificar cuándo podemos usar la herencia para que no tengamos los problemas que conlleva.

- 1** La herencia se usa si una instancia de una clase Hija nunca necesitará convertirse en un objeto de otra clase.
- 2** Si la jerarquía de herencia representa una relación "Es una" y no una relación "Tiene una".
- 3** Si desea o necesita realizar cambios globales en sus clases secundarias cambiando una clase principal.
- 4** Cuando la clase secundaria se extiende en lugar de reemplazar total o parcialmente las responsabilidades de la clase principal.

DigitalHouse>
Coding School