



Herencia

```
padding-top: 8px;";  
fo_bar">  
left;" for="keywords" class="tag-editor-  
information_container" class="field_inform  
ount_info" class="field_inform  
: -3px;"></a>  
og" class="field_inform  
: right; padding-top: 7px;">  
both;"></div>  
ords" class="tag-editor-  
itor ui-sortable">  
h:1px"></li>  
ceholder">  
eywords or paste via Ctrl+  
keywords_for_clipboard" style="width:  
n_keywords_container" style="width:  
s-feedback has-clear" style="clear:  
div > div.keywords_info_bar
```





Herencia

La herencia es una de las características principales de la programación orientada a objetos y con ella podemos reutilizar las clases ya construidas para crear unas nuevas y más poderosas o con capacidades un tanto diferentes a las anteriores. El tener este mecanismo, en lugar de copiar y pegar, confiere un poder para implementar nuevas funcionalidades solo modificando la clase padre y, en consecuencia, todas las clases hijas lo tendrán. ¡Vaya, qué manera de implementar la eficiencia!

Explicación

Componentes de la herencia

¿Cuándo se puede aplicar el concepto de herencia en las clases?

El concepto de herencia se puede aplicar en las clases cuando estas guardan una relación del tipo “es un”.

En este caso, no se trata de que un objeto contenga atributos de otro objeto, sino que el primero en sí mismo puede tratarse como el segundo.

Algunos ejemplos de clases que guardan esta relación son los siguientes:

- Un auto es un vehículo.
- Un triángulo es una figura.
- Un empleado es una persona. Un cliente también es una persona.

Cuando las clases tienen una relación del tipo “es un”, se dice que una de las clases es una Clase base (Superclase) y la otra una Clase derivada (Subclase). Por ejemplo: un empleado es una persona;

también un cliente es una persona. Entonces se dice que la clase *Empleado* es una clase derivada que hereda de la clase *Persona* y la clase *Cliente* también es una clase derivada de la clase *Persona*.

Los empleados y los clientes son un tipo especial de persona, “un empleado es una persona”, “un cliente es una persona”, pero es incorrecto decir que “un empleado es un cliente”.

¿Cuántas clases derivadas pueden tener una clase base?

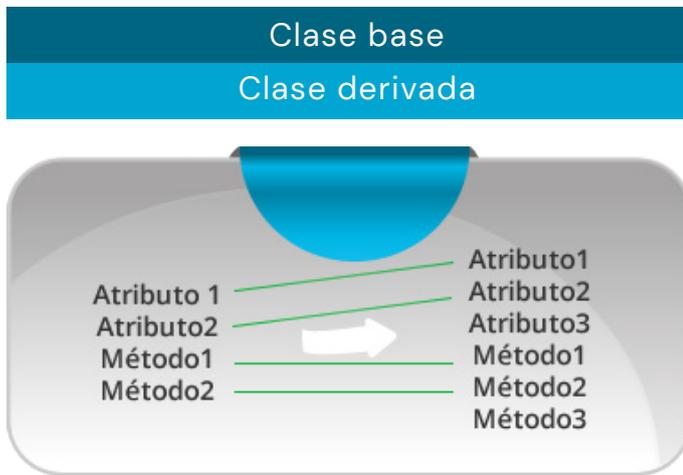
Una clase base puede tener muchas clases derivadas. De hecho, la relación de herencia forma una estructura jerárquica en la que en la parte superior se encuentra la clase base y descienden de ella las clases derivadas.

¿Cómo se debe indicar en Java que se empleará el mecanismo de herencia?

La forma de indicar en Java que se empleará el mecanismo de herencia es empleando la *extends* de la siguiente manera:

```
class nombreClaseBase {
    ...variables de instancia y
    métodos
}
class nombreClaseDerivada extends nombreClaseBase {
    ...variables de instancia y
    métodos
}
```

Al establecer la herencia entre las clases, los atributos y métodos de la clase base se heredan a la clase derivada.



Ejemplo:

Un libro es una publicación.

Entonces la clase derivada *libro* se crea a partir de la clase base *Publicación*.

Un libro tendría tres atributos: editor, edición e ISBN.

```
class Publicacion
{
    private String editor;
    private int edicion;
    public String getEditor() {
return this.editor; }
    public int getEdicion() { return
this.edicion; }
    public void setEditor( String
editor) { this.editor = editor; }
    public void setEdicion( int
edicion) { this.edicion = edicion;
}
}
public class Libro extends
publicacion
{
    private String ISBN;
    public String getISBN() { return
this.ISBN; }
    public void setISBN( String
ISBN) { this.ISBN = ISBN; }
```