

7.C. Herencia.

1. Herencia.

1.11. Clases y métodos finales.

En unidades anteriores has visto el **modificador final**, aunque **sólo** lo has utilizado por ahora **para atributos y variables** (por ejemplo para declarar **atributos constantes**, que una vez que toman un valor ya **no pueden ser modificados**). Pero este modificador también puede ser **utilizado con clases y con métodos** (con un comportamiento que no es exactamente igual, aunque puede encontrarse cierta analogía: **no se permite heredar o no se permite redefinir**).

Una clase declarada como final no puede ser heredada, es decir, **no puede tener clases derivadas**. La jerarquía de clases a la que pertenece acaba en ella (**no tendrá clases hijas**):

```
[modificador_acceso] final class nombreClase [herencia] [interfaces]
```

Un **método** también puede ser declarado como **final**, en tal caso, **ese método no podrá ser redefinido en una clase derivada**:

```
[modificador_acceso] final <tipo> <nombreMetodo> ([parámetros]) [excepciones]
```

Si **intentas redefinir un método final** en una subclase se producirá un **error de compilación**.

Autoevaluación

Los modificadores **final** y **abstract** son excluyentes en la declaración de un **método**. ¿Verdadero o Falso?

- ☐ Verdadero
☐ Falso

Además de en la declaración de atributos, clases y métodos, el modificador **final** también podría aparecer acompañando a un método de un **parámetro**. En tal caso **no se podrá modificar el valor del parámetro dentro del código del método**. Por ejemplo: **public final metodoEscribir (int par1, final int par2)**.