**Conocimiento: Clases Finales**

Algunas veces es necesario evitar que una clase sea heredada. Para hacerlo hay que utilizar el modificador **final** en la cáscara, antes del nombre de la clase. Declarando una clase como final, implícitamente declaramos también todos sus métodos finales, ya que al no haber herencia no hay sobrescritura de métodos.

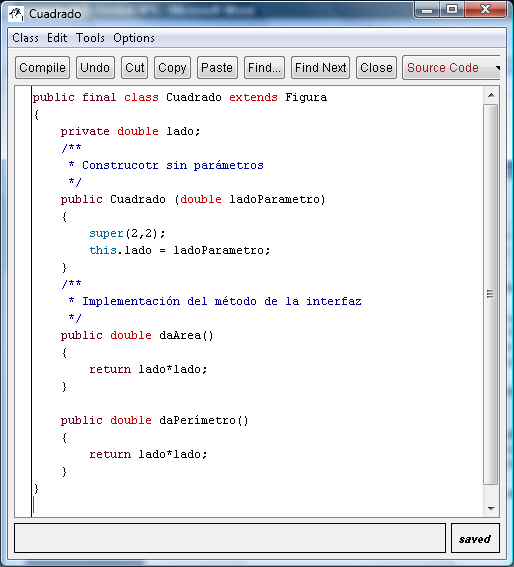
Por tanto, una clase final es una clase que no se puede derivar o heredar. Es decir no posee subclases.

Cabe destacar que las palabras reservadas **final** y **abstract** no pueden ir en la misma cáscara, ya que una clase abstracta es incompleta por definición y se basa en sus subclases para proporcionar una implementación completa.

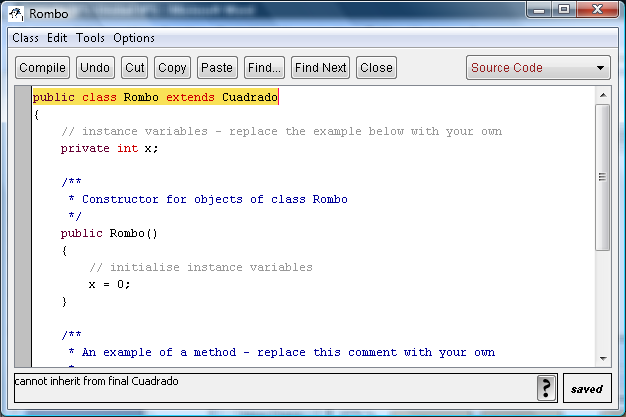
En java existen varias clase String, las que no permiten herencia solo colaboración con ellas.

Por ejemplo:

Tenemos una clase final:



Si queremos tener una subclase de Cuadrado, veamos que sucede al compilar



El error que nos da es que no se puede heredar de la clase final Cuadrado.

**Métodos Finales**

La sobrescritura de métodos es una de las características más importantes del lenguaje Java, pero puede haber ocasiones en las que sea necesario evitarla. Para impedir que un método se sobrescriba, hay que especificar el modificador final en la firma. Los métodos declarados como final no pueden sobrescribirse.

Los métodos declarados como final a veces pueden proporcionar una mejora del rendimiento porque el compilador puede realizar llamadas en línea a esos métodos, ya que sabe que no serán sobrescritos por una subclase.

En las clases ya implementadas en java se han declarado ciertos métodos como finales para evitar que el comportamiento esperado cambie en las subclases, por ejemplo en la clase Object los métodos **getClass(), notify(), notifyAll() y wait()** han sido declarados como final.

Sintaxis:

[modificador de acceso] **final** tipo\_retorno nombreMetodo(parámetros)

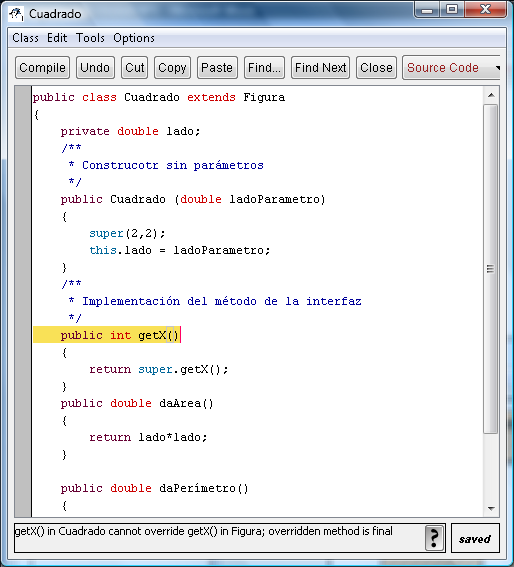
{

Bloque de código

}

Por ejemplo: tenemos la clase Figura con un método final:

Entonces en la subclase Cuadrado intentamos sobrescribir este método, veamos qué sucede:



El error claramente nos informa que no podemos sobrescribir getX() ya que es final en la clase padre.