

OPERADOR FINAL

PALABRA FINAL EN JAVA

- **USOS DE LA PALABRA FINAL.**
- **EN VARIABLES:** Evita cambiar el valor que almacena la variable
- **EN MÉTODOS:** Evita que se modifique la definición de un método desde una subclase
- **EN CLASES:** Evita que se cree una subclase



Ej. `Math.PI`

La palabra final afecta no solamente a variables, sino también es posible utilizarlo en métodos y en clases. En cada uno tiene un significado distinto que es lo que estudiaremos en esta lección.

En primer lugar veremos lo que implica utilizar la palabra final al definir una variable.

Es posible inicializar una variable final, pero una vez que se ha inicializado su valor, ya no es posible modificarlo, sin embargo recordemos que las variables que almacenan referencias de objetos no contienen el valor en si del objeto, sino solo de la referencia donde se ubica el mismo. Esto quiere decir que es posible cambiar el estado del objeto, pero no la referencia almacenada en la variable que se creo indicando que es final. En pocas palabras se dice que la palabra final es como crear una constante en Java, sin embargo debemos tomar en cuenta lo mencionado respecto a los objetos.

Debido a que la palabra final es como crear una constante, es muy común que se combine con la palabra static para poderla acceder directamente en lugar de crear una instancia de la clase para poder usar la variable final, en cambio al definir una variable como public static y final podemos accederla directamente indicando el nombre de la clase y posteriormente el nombre de la variable, y más aún, si agregamos el import static podemos utilizar directamente el nombre de la constante como veremos más adelante. Un ejemplo muy común es el uso de la constante PI, la cual se encuentra definida en la clase Math. Esta variable si viéramos su código está definida como public final y static, por lo tanto podemos acceder escribiendo: `Math.PI` y esto nos regresará el valor de 3.1416...

Cuanto estamos hablando de utilizar la palabra final en la definición de un método, básicamente estamos diciendo que una subclase no puede sobrescribir el comportamiento del método de la clase padre, es decir, que así como lo hereda debe utilizarlo y no puede modificarlo. El tema de sobrescritura lo veremos a detalle en el siguiente curso.

Y finalmente, cuando hablamos de clases, cuando definimos una clase como final, lo que indicamos es que no se puede crear clase que extienda de esta clase, es decir, no tendrá clases hijas. Es como indicar que esta será la última clase en la jerarquía de clases para esta clase en particular.

