

# Interfaz en Java

Una **interfaz en java** es un plano de una clase. Tiene constantes estáticas y métodos abstractos.

Solo puede haber métodos, no el cuerpo del método. Se utiliza para lograr abstracción y herencia múltiple en Java.

En otras palabras, puede decir que las interfaces pueden tener métodos y variables abstractos. No puede tener un cuerpo de método.

La interfaz Java también **representa la relación IS-A**.

No puede ser instanciado como la clase abstracta.

## ¿Por qué usar interfaces?

Hay principalmente tres razones para usar la interfaz. Se dan a continuación.

- Se utiliza para lograr la abstracción.
- Por interfaz, podemos admitir la funcionalidad de herencia múltiple.
- Se puede utilizar para lograr un bajo acoplamiento .



## ¿Cómo declarar una interfaz?

Una interfaz se declara utilizando la palabra clave de interfaz. Proporciona abstracción total; significa que todos los métodos en una interfaz se declaran con el cuerpo vacío, y todos los campos son públicos, estáticos y finales por defecto. Una clase que implementa una interfaz debe implementar todos los métodos declarados en la interfaz.

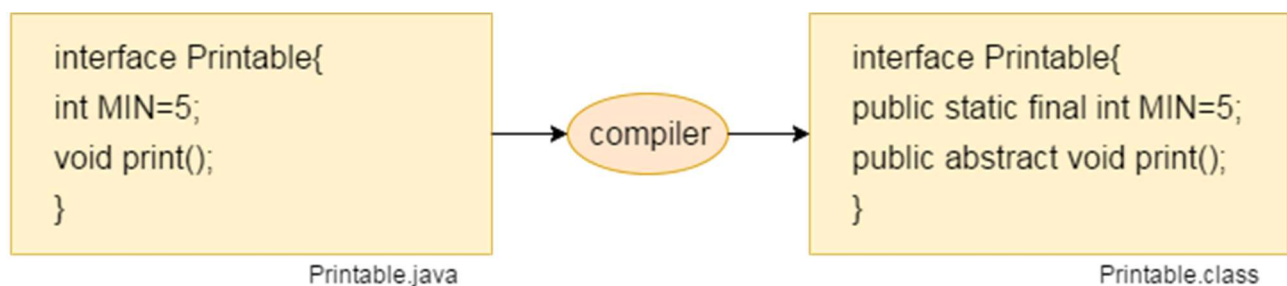
### Sintaxis:

1. **interfaz** <nombre de interfaz> {
- 2.
3.     // declara campos constantes
4.     // declara métodos que resumen
5.     // por defecto.
6. }
- 7.

### Adición interna por parte del compilador.

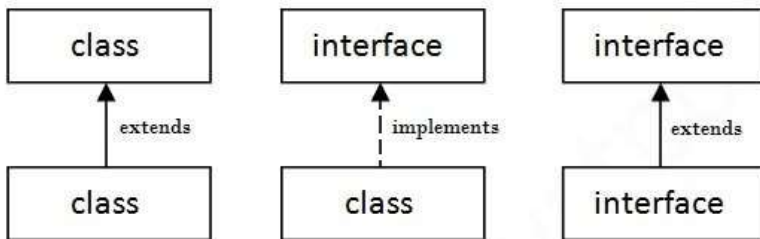
El compilador de Java agrega palabras clave **public** y **abstract** antes del método de interfaz. Además, agrega palabras clave **static** y **final** antes de los miembros de datos, constantes o variables

En otras palabras, los campos de la interfaz son públicos, estáticos y finales por defecto, y los métodos son públicos y abstractos.



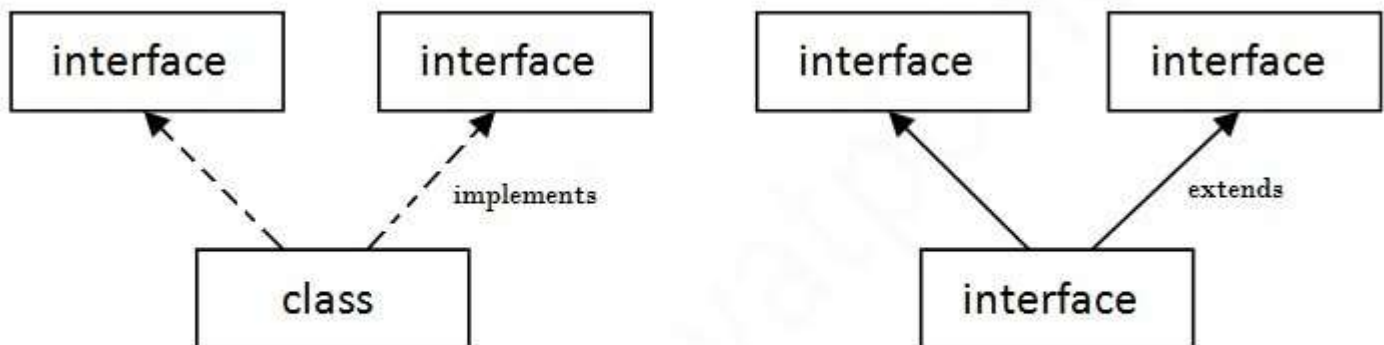
## La relación entre clases e interfaces.

Como se muestra en la figura siguiente, una clase extiende otra clase, una interfaz extiende otra interfaz, pero una **clase implementa una interfaz**.



## Herencia múltiple en Java por interfaz

Si una clase implementa múltiples interfaces, o una interfaz extiende múltiples interfaces, se conoce como herencia múltiple.



**Multiple Inheritance in Java**