

# Polimorfismo. Sobrecarga de métodos.

Si una clase tiene varios métodos con el mismo nombre pero con parámetros diferentes, se conoce como **Sobrecarga de métodos**.

Si tenemos que realizar solo una operación, tener el mismo nombre de los métodos aumenta la legibilidad del programa.

Suponga que tiene que realizar la suma de los números dados, pero puede haber cualquier número de argumentos, si escribe el método como `a (int, int)` para dos parámetros y `b (int, int, int)` para tres parámetros, entonces Puede ser difícil programadores comprender el comportamiento del método porque su nombre es diferente.

Por lo tanto, realizamos una sobrecarga de métodos para descubrir el programa rápidamente.

## Ventaja de la sobrecarga de métodos

La sobrecarga de métodos *aumenta la legibilidad del programa*.

## Diferentes formas de sobrecargar el método

Hay dos formas de sobrecargar el método en Java

1. Al cambiar el número de argumentos
2. Al cambiar el tipo de datos

En java, la sobrecarga de métodos no es posible cambiando solo el tipo de retorno del método.

---

## 1) Método de sobrecarga: cambio no. de argumentos

En este ejemplo, hemos creado dos métodos, el primer método `add ()` realiza la suma de dos números y el segundo método `add` suma de tres números.

En este ejemplo, estamos creando métodos estáticos para que no necesitemos crear instancias para llamar a métodos.

---

## 2) Sobrecarga de métodos: cambio del tipo de datos de argumentos

En este ejemplo, hemos creado dos métodos que difieren en el tipo de datos. El primer método `add` recibe dos argumentos enteros y el segundo método `add` recibe dos argumentos dobles.

---

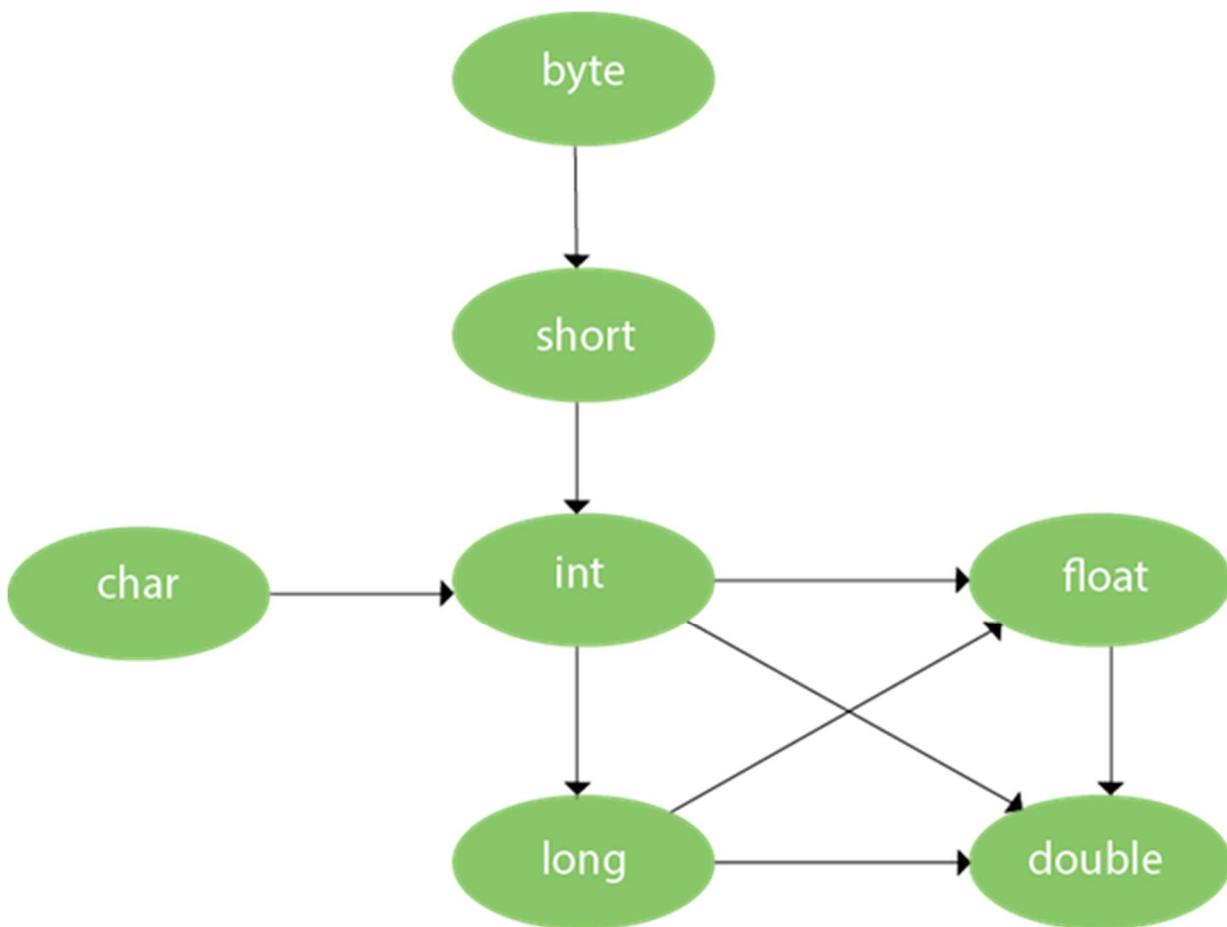
P) ¿Por qué la sobrecarga de métodos no es posible cambiando solo el tipo de método de retorno?

En java, la sobrecarga del método no es posible cambiando el tipo de retorno del método solo debido a la ambigüedad.

**Nota:** El compilador de Java genera un error de tiempo del compilador si declara que el mismo método tiene los mismos parámetros.

## Promoción de tipos primitivos

Un tipo se promueve a otro implícitamente si no se encuentra ningún tipo de datos coincidente. Comprendamos el concepto por la figura que se muestra a continuación:



Como se muestra en el diagrama anterior, el byte puede promoverse a short, int, long, float o double. El tipo de datos corto puede promoverse a int, long, float o double. El tipo de datos char se puede promover a int, long, float o double, etc.