

Sobrecarga de métodos

La sobrecarga de métodos en Java es una característica que te permite definir **varios métodos en una clase con el mismo nombre pero con conjuntos de parámetros diferentes**. Esto facilita la escritura de código más limpio y legible, ya que puedes utilizar el mismo nombre de método para realizar operaciones similares.

Ejemplo sencillo para ilustrar la sobrecarga de métodos en Java:

```
Java
public class EjemploSobrecarga {

    // Primer método con dos parámetros de tipo int
    public int sumar(int a, int b) {
        return a + b;
    }

    // Segundo método con tres parámetros de tipo double
    public double sumar(double a, double b, double c) {
        return a + b + c;
    }

    // Tercer método con dos parámetros de tipo String
    public String concatenar(String str1, String str2) {
        return str1 + str2;
    }

    public static void main(String[] args) {
        EjemploSobrecarga ejemplo = new EjemploSobrecarga();

        // Llamando al primer método
        int resultado1 = ejemplo.sumar(5, 10);
        System.out.println("Resultado 1: " + resultado1);

        // Llamando al segundo método
        double resultado2 = ejemplo.sumar(2.5, 3.5, 1.5);
        System.out.println("Resultado 2: " + resultado2);

        // Llamando al tercer método
        String resultado3 = ejemplo.concatenar("Hola, ", "mundo!");
        System.out.println("Resultado 3: " + resultado3);
    }
}
```

En este ejemplo, la clase **EjemploSobrecarga** tiene tres métodos llamados sumar, pero cada uno tiene un conjunto diferente de parámetros. La sobrecarga de métodos permite utilizar el mismo nombre de método para operaciones similares, lo que hace que el código sea más legible y fácil de entender. Veamos más detalladamente cada uno de los métodos:

1. Método sumar con dos parámetros de tipo int:

Java

```
public int sumar(int a, int b) {  
    return a + b;  
}
```

Este método toma dos enteros como parámetros y devuelve la suma de ambos.

2. Método sumar con tres parámetros de tipo double:

Java

```
public double sumar(double a, double b, double c) {  
    return a + b + c;  
}
```

En este caso, el método **sumar** acepta tres números de punto flotante (double) como parámetros y devuelve la suma de los tres.

3. Método concatenar con dos parámetros de tipo String:

Java

```
public String concatenar(String str1, String str2) {  
    return str1 + str2;  
}
```

El tercer método, llamado **concatenar**, toma dos cadenas (String) como parámetros y devuelve la concatenación de ambas.

En el método main, se crea una instancia de la clase **EjemploSobrecarga** y se llaman a cada uno de los métodos sumar y concatenar con diferentes conjuntos de argumentos. Esto demuestra cómo la sobrecarga de métodos te permite utilizar el mismo nombre de método para realizar operaciones similares con diferentes tipos de datos.

Sobrecarga de constructores

La sobrecarga de constructores sigue el mismo principio que la sobrecarga de métodos, permitiéndote tener **varios constructores en la misma clase con diferentes conjuntos de parámetros**. Esto es útil cuando deseas crear instancias de una clase de diferentes maneras o con diferentes configuraciones. A continuación mostramos un ejemplo de cómo se realiza la sobrecarga de constructores:

Java

```
public class EjemploSobrecargaConstructores {

    private int valor1;
    private double valor2;
    private String texto;

    // Constructor sin parámetros
    public EjemploSobrecargaConstructores() {
        // Valores por defecto
        this.valor1 = 0;
        this.valor2 = 0.0;
        this.texto = "Sin inicializar";
    }

    // Constructor con un parámetro int
    public EjemploSobrecargaConstructores(int valor) {
        this.valor1 = valor;
        this.valor2 = 0.0;
        this.texto = "Con valor entero";
    }

    // Constructor con dos parámetros (int y double)
    public EjemploSobrecargaConstructores(int valor1, double valor2) {
        this.valor1 = valor1;
        this.valor2 = valor2;
        this.texto = "Con valores entero y double";
    }
}
```

```
// Constructor con un parámetro String
public EjemploSobrecargaConstructores(String texto) {
    this.valor1 = 0;
    this.valor2 = 0.0;
    this.texto = texto;
}

// Otros métodos de la clase, getters y setters...

public static void main(String[] args) {

    // Instancias utilizando diferentes constructores
    EjemploSobrecargaConstructores instancia1 = new
    EjemploSobrecargaConstructores();
    EjemploSobrecargaConstructores instancia2 = new
    EjemploSobrecargaConstructores(10);
    EjemploSobrecargaConstructores instancia3 = new
    EjemploSobrecargaConstructores(5, 3.5);
    EjemploSobrecargaConstructores instancia4 = new
    EjemploSobrecargaConstructores("Hola");
}
}
```

En este ejemplo podemos observar la presencia de:

1. Constructor sin parámetros:

Crea una instancia con valores por defecto.

2. Constructor con un parámetro int:

Crea una instancia con un valor entero proporcionado como argumento.

3. Constructor con dos parámetros (int y double):

Crea una instancia con valores entero y de punto flotante proporcionados como argumentos.

4. Constructor con un parámetro String:

Crea una instancia con un valor de cadena proporcionado como argumento.

En el método main, se crean diferentes instancias de la clase utilizando los diversos constructores. Esto ilustra cómo puedes inicializar objetos de la misma clase de diferentes maneras según los parámetros proporcionados. Por lo tanto, la sobrecarga de constructores en Java no es más que una práctica común que te permite proporcionar flexibilidad al usuario de tu clase al permitirles inicializar objetos de maneras diferentes según sus necesidades.