

Se llama **sobrecarga de operadores** cuando reutilizando el mismo **operador** con un número de usos diferentes, y el compilador decide como usar ese **operador** dependiendo sobre qué opera. La **sobrecarga de operadores** solo se puede utilizar con clases, no se pueden redefinir los **operadores** para los tipos simples predefinidos.

al igual que las funciones, pueden ser sobrecargados (**overloaded**). Este es uno de los aspectos más característicos de este lenguaje. La **sobrecarga de operadores** quiere decir que se pueden **redefinir algunos de los operadores existentes** en C++ para que actúen de una determinada manera, definida por el programador, con los objetos de una clase determinada. Esto puede ser muy útil por ejemplo, para definir operaciones matemáticas con elementos tales como **vectores y matrices**. Así, sobrecargando adecuadamente los operadores **suma (+) y asignación (=)**, se puede llegar a sumar dos matrices con una sentencia tan simple como:

**C = A + B;**

Otra capacidad muy utilizada es la de sobrecargar los **operadores de inserción y extracción** en los flujos de entrada y salida (>> y <<), de manera que puedan imprimir o leer estructuras o clases complejas con una sentencia estándar.

Los únicos operadores de C que no se pueden sobrecargar son el operador punto (.), el if aritmético (? :) y el operador sizeof. C++ añade otros 2 a esta lista: el scope resolution operator (::) y puntero a miembro de un objeto (. \*).

¿Me puede decir si es posible sobrecargar operadores en Java? Si se usa en algún lugar de Java, ¿podría contarme al respecto?

No, Java no admite la sobrecarga del operador definido por el usuario. El único aspecto de Java que se acerca a la sobrecarga del operador "personalizado" es el manejo de + para cadenas, que da como resultado la concatenación de constantes en tiempo de compilación o la concatenación en tiempo de ejecución usando StringBuilder / StringBuffer. Sin embargo, no puede definir sus propios operadores que actúen de la misma manera.