

Constructores en Java

En Java, un constructor es un bloque de códigos similar al método, solo lleva el tipo de la clase y parámetros en caso de ser parametrizado.

Se llama cuando se crea una instancia de la clase. En el momento de llamar al constructor, la memoria para el objeto se asigna en la memoria.

Es un tipo especial de método que se utiliza para inicializar el objeto.

Cada vez que se crea un objeto utilizando la palabra clave `new` (), se llama al menos un constructor.

Llama a un constructor predeterminado si no hay ningún constructor disponible en la clase. En tal caso, el compilador de Java proporciona un constructor predeterminado de forma predeterminada.

Hay dos tipos de constructores en Java: constructor sin argumentos y constructor parametrizado.

Nota: Se llama constructor porque construye los valores en el momento de la creación del objeto. No es necesario escribir un constructor para una clase. Es porque el compilador de Java crea un constructor predeterminado si su clase no tiene ninguno.

Reglas para crear el constructor Java

Hay dos reglas definidas para el constructor.

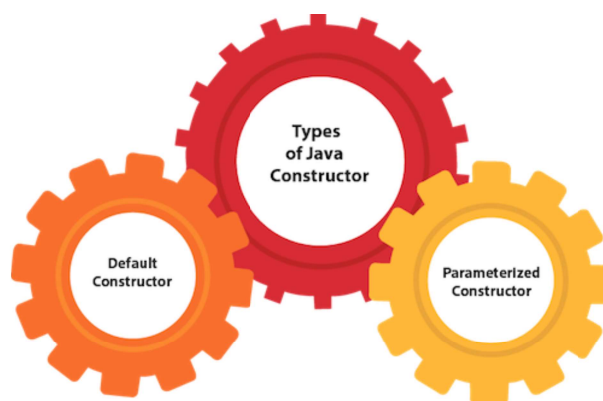
1. El nombre del constructor debe ser el mismo que su nombre de clase
2. Un constructor no debe tener ningún tipo de retorno explícito
3. Un constructor Java no puede ser abstracto, estático, final o sincronizado.

Nota: Podemos usar modificadores de acceso al declarar un constructor. Controla la creación del objeto. En otras palabras, podemos tener un constructor privado, protegido, público o predeterminado en Java.

Tipos de constructores Java

Hay dos tipos de constructores en Java:

1. Constructor predeterminado (constructor sin argumentos)
2. Constructor parametrizado



Constructor predeterminado de Java

Un constructor se llama "Constructor predeterminado" cuando no tiene ningún parámetro.

Regla: si no hay un constructor en una clase, el compilador crea automáticamente un constructor predeterminado.



P) ¿Cuál es el propósito de un constructor predeterminado?

El constructor predeterminado se utiliza para proporcionar los valores predeterminados al objeto como 0, nulo, etc., según el tipo.

Constructor parametrizado Java

Un constructor que tiene un número específico de parámetros se denomina constructor parametrizado.

¿Por qué usar el constructor parametrizado?

El constructor parametrizado se usa para proporcionar valores diferentes a objetos distintos. Sin embargo, también puede proporcionarnos los mismos valores.

Ejemplo: C_ConstructorParametrizado.java

Sobrecarga de constructores en Java

En Java, un constructor es como un método pero sin tipo de retorno. También se puede sobrecargar como los métodos de Java.

La sobrecarga de constructores en Java es una técnica de tener más de un constructor con diferentes listas de parámetros. Están organizados de manera que cada constructor realiza una tarea diferente. El compilador los diferencia por el número de parámetros en la lista y sus tipos.

Diferencia entre constructor y método en Java

Hay muchas diferencias entre constructores y métodos. Se dan a continuación.

Constructor Java	Método Java
Un constructor se usa para inicializar el estado de un objeto.	Se usa un método para exponer el comportamiento de un objeto.
Un constructor no debe tener un tipo de retorno.	Un método debe tener un tipo de retorno.
El constructor se invoca implícitamente.	El método se invoca explícitamente.
El compilador de Java proporciona un constructor predeterminado si no tiene ningún constructor en una clase.	El método no es proporcionado por el compilador en ningún caso.
El nombre del constructor debe ser el mismo que el nombre de la clase.	El nombre del método puede o no ser el mismo que el nombre de la clase.

Constructor por copia

Hay muchas formas de copiar los valores de un objeto en otro en Java. Son:

- Por constructor
- Al asignar los valores de un objeto a otro
- Por el método clone () de la clase Object

En este ejemplo, vamos a copiar los valores de un objeto en otro usando el constructor Java.

Ejemplo: E_ConstructorPorCopia.java

Copiando valores sin constructor

Podemos copiar los valores de un objeto en otro asignando los valores de los objetos a otro objeto. En este caso, no hay necesidad de crear el constructor.