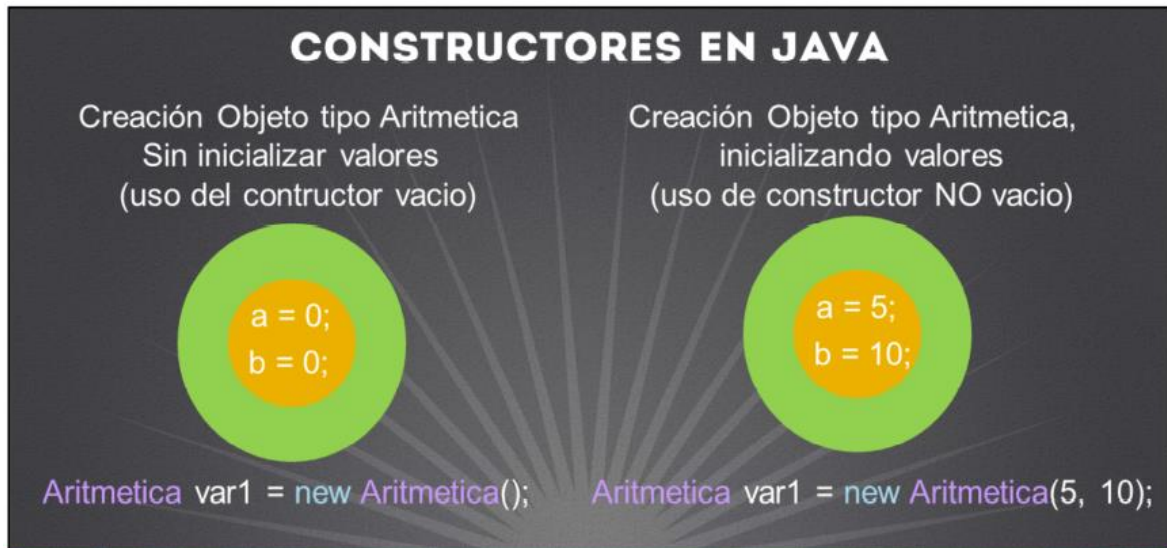


CONSTRUCTORES



Puede ser muy tedioso estar inicializando las variables de una clase cada vez que creamos un objeto, para ello Java permite agregar valores a los objetos desde el mismo momento de su creación. A este concepto se le conoce como Constructor, y básicamente es un método pero que nos permitirá crear un objeto con los valores que proporcionemos al momento de crear nuestro objeto.

Los constructores aunque son muy similares a los métodos, tienen ciertas diferencias y características, tales como:

- Estos métodos sólo se puede ejecutar al momento de la creación de un objeto, no es posible utilizarlos después.
- No devuelven ningún valor.
- El nombre del constructor es idéntico al nombre de la clase, así es como podemos identificar si una clase tiene constructores (siempre respetando las mayúsculas y minúsculas)
- Por defecto en Java, se crea un constructor sin argumentos, conocido como constructor vacío. Este constructor lo agrega en automático el compilador de Java a nuestra clase, sin embargo si nosotros definimos un constructor distinto al constructor vacío, es decir, con argumentos, entonces Java ya no agrega el constructor vacío y es nuestra responsabilidad agregar el constructor vacío a nuestra clase si fuera necesario.
- El constructor vacío es necesario para crear un objeto, recordemos la sintaxis básica general para crear un objeto.
 - `TipoClase objeto = new TipoClase();`
- Si observamos al final de la línea de código estamos abriendo y cerrando paréntesis, ese es precisamente el constructor vacío que el compilador agregó por nosotros siempre y cuando no agreguemos constructores con otros argumentos. Por lo que ahora ya podemos entender que después de la palabra `new` realmente lo que estamos colocando es el nombre del constructor que deseamos llamar, pudiendo tener argumentos o no.

En la figura podemos observar un ejemplo de nuestra clase `Aritmetica`, usando un constructor vacío y otro no vacío para crear nuestros objetos.

Supongamos que tenemos dos atributos en nuestra clase, de tipo entero, llamado `a` y `b`. En el primer caso estamos creando un objeto llamando al constructor vacío, por lo tanto los valores de nuestros atributos se inicializan con los valores por default según el tipo de cada atributo. Si es un tipo primitivo, se inicializan según su valor por default, y si son de tipo `object` su valor inicia es `null`.

Por otro lado creamos un objeto `Aritmetica` pero con valores iniciales desde el momento en que estamos creando el objeto respectivo. Estamos enviando los valores de 5 y 10 respectivamente, y por lo tanto podemos tomar estos valores para inicializar