

Los Constructores

Toda clase tiene un constructor. Los constructores son operaciones que son invocadas al instanciar una clase. Los constructores se encargan de inicializar atributos y configurar todo lo que sea necesario antes de que comience a existir el objeto.

Al tratarse de operaciones, los constructores pueden contener cualquier comportamiento, aunque éstas no tengan que ver exactamente con “construir” el objeto. Es posible tener más de un constructor en una misma clase; en este caso, cada constructor debe aceptar parámetros distintos para poder así distinguirlos.

Los constructores suelen llamarse con el mismo nombre de las clases, como se muestra a continuación:

Automovil
- color: int - modelo: string - precio: int - usado: boolean
+ acelerar(int, int): void + Automovil(): void + frenar(int, int): void

CuentaBancaria
- saldo: int
+ CuentaBancaria(): void + depositar(int): void + extraer(int): void

Por lo tanto la clase “Automóvil” tendrá un constructor llamado “Automovil()”, y la clase “CuentaBancaria” tendrá un constructor llamado “CuentaBancaria()”. En los ejemplos anteriores los constructores no tienen parámetros.

En resumen, un constructor es una operación que es invocada al instanciar una clase.

Los Destructores

Los destructores son funciones miembro especiales que sirven para eliminar un objeto de una determinada clase. El destructor realizará procesos necesarios cuando un objeto termine su ámbito temporal, por ejemplo liberando la memoria dinámica utilizada por dicho objeto o liberando recursos usados, como archivos, dispositivos, etc.

Los destructores tienen ciertas características:

- También tienen el mismo nombre que la clase a la que pertenecen, pero tienen el símbolo ~ delante.
- No tienen tipo de retorno, y por lo tanto no retornan ningún valor.
- No tienen parámetros.
- No pueden ser heredados.
- Deben ser públicos, no tendría ningún sentido declarar un destructor como privado, ya que siempre se usan desde el exterior de la clase, ni tampoco como protegido, ya que no puede ser heredado.
- No pueden ser sobrecargados, lo cual es lógico, puesto que no tienen valor de retorno ni parámetros, no hay posibilidad de sobrecarga.
- Una clase no puede tener más de un destructor. Además, si se omite el destructor, el compilador lo crea automáticamente.

El sistema en tiempo de ejecución llama automáticamente a un destructor de la clase cuando la instancia de esa clase cae fuera de alcance o cuando la instancia se borra explícitamente.