

Palabras reservadas

Elemento	Descripción	Ejemplo	Valor por Defecto
CLASS	Define la estructura básica para crear objetos. Es el "molde" de la POO.	<code>class Coche { ... }</code>	N/A
NEW	Operador que asigna memoria para crear una nueva instancia de una clase.	<code>Coche miCoche = new Coche();</code>	N/A
THIS	Referencia autorreferencial para acceder a atributos/métodos del objeto actual.	<code>this.modelo = "Tesla";</code>	N/A
STATIC	Vincula un miembro (método/variable) directamente a la clase.	<code>static int contador = 0;</code>	N/A
PUBLIC	Modificador de acceso sin restricciones.	<code>public void acelerar() { ... }</code>	N/A
PRIVATE	Restringe el acceso solo a la clase que lo contiene.	<code>private String serial;</code>	N/A
PROTECTED	Permite acceso a clases hijas y clases del mismo paquete.	<code>protected int añoFabricacion;</code>	N/A
VOID	Indica que un método no devuelve ningún valor.	<code>public void imprimir() { System.out.println("Hola"); }</code>	N/A
RETURN	Finaliza un método y opcionalmente devuelve un valor.	<code>return precio * 1.21;</code>	N/A
EXTENDS	Establece herencia entre clases.	<code>class Eléctrico extends Coche { ... }</code>	N/A
SUPER	Referencia a la clase padre.	<code>super();</code> o <code>super.metodoPadre();</code>	N/A

INT	Tipo primitivo para números enteros (32 bits).	int edad = 25;	0
DOUBLE	Tipo primitivo para números decimales de alta precisión (64 bits).	double pi = 3.1416;	0.0
FLOAT	Tipo primitivo para números decimales de 32 bits (menor precisión).	float temperatura = 23.5f;	0.0f
BOOLEAN	Tipo lógico que solo admite true o false.	boolean esNuevo = true;	false
STRING	Clase para almacenar secuencias de caracteres inmutables.	String nombre = "DeepSeek";	null

Métodos

Tipo de método	Descripción	Ejemplo	Valor por Defecto
Método estático	Pertenece a la clase (no a instancias). Se llama sin crear objeto.	static int sumar(int a, int b) { return a + b; }	N/A
Método de instancia	Requiere un objeto para ser invocado. Accede al estado del objeto.	public void frenar() { velocidad = 0; }	N/A
Método void	No devuelve ningún valor.	public void imprimir() { System.out.println("Hola"); }	N/A
Método con retorno	Devuelve un valor de un tipo específico.	public double calcularIVA(double precio) { return precio * 0.21; }	Depende del tipo
Método Main	Punto de entrada de una aplicación Java.	public static void main(String[] args) { ... }	N/A

Método constructor	Inicializa un objeto al crearlo. Mismo nombre que la clase.	public Coche(String modelo) { this.modelo = modelo; }	N/A
Método get/set	Accede (get) o modifica (set) atributos privados.	public String getModelo() { return modelo; }	N/A
Método toString()	Devuelve una representación en String del objeto.	public String toString() { return "Coche: " + modelo; }	NombreClase@hashCode
Método equals()	Compara si dos objetos son equivalentes.	public boolean equals(Coche c) { return this.modelo.equals(c.modelo); }	false