Universidad Católica de Oriente

Palabras Reservadas en Java

Las palabras reservadas en Java tienen un propósito específico para el lenguaje y por ende para el compilador, y por lo tanto no pueden usarse para algo distinto, como nombrar variables o métodos propios.

A continuación, se presenta la lista de las 50 palabras reservadas en Java:

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

abstract: Se usa para declarar clases o métodos abstractos, los cuales se implementan más tarde en una subclase.

assert:

boolean: Tipo de dato primitivo que puede tener valores booleanos de true o false.

break: Sentencia utilizada para romper el flujo normal del bloque de código actual, es comúnmente usado para salir de bucles.

byte: Tipo de dato primitivo que puede almacenar un número entero de hasta 8 bits.

case: Se utiliza en las sentencias switch para especificar los valores que se van a evaluar.

catch: Se utiliza para capturar las excepciones generadas en una sentencia try.

char: Tipo de dato primitivo que puede almacenar un carácter UNICODE.

class: Se utiliza para declarar una clase.

const: Palabra reservada que no es utilizada dentro de Java.

continue: Sentencia que se usa para ir a la siguiente iteración del bucle que está siendo ejecutado.

default: Indica el bloque de código que se ejecuta por defecto sí ninguno de los case aplica para el valor del switch.

do: Se usa para declarar el inicio de un ciclo do-while.

double: Tipo de dato primitivo que puede almacenar números con punto flotante de hasta 64 bits.

else: Indica la opción que se ejecuta sí la sentencia if que lo acompaña resulta falsa.

enum:

extends: Indica que una clase extiende o hereda de otra que se denomina "superclase".

final: Se utiliza para definir clases, métodos o atributos que no se pueden reemplazar o sobrescribir. Y en el caso de las clases no se pueden extender de ellas.

finally: Determina el bloque de código que se ejecuta siempre luego de un try-catch, independientemente de si ocurrió una excepción o no.

float: Tipo de dato primitivo que puede almacenar números con punto flotante de hasta 32 bits.

for: Se usa para declarar una estructura de control cíclica que lleva su mismo nombre.

goto: Palabra reservada que no es utilizada dentro de Java.

if: Se utiliza para definir una estructura de control condicional que evalúa si una expresión es verdadera o no.

implements: Se usa para indicar que una clase implementa una o varias interfaces.

import: Indica la ruta o los paquetes en los que se encuentran las clases o interfaces utilizadas dentro de otra clase.

instanceof: Es un operador que determina sí un objeto es una instancia de una clase.

int: Tipo de dato primitivo que puede almacenar un número entero de hasta 32 bits.

interface: Se utiliza para declarar una interface en Java.

long: Tipo de dato primitivo que puede almacenar un número entero de hasta 64 bits.

native: Se usa para indicar que un método es implementado en código nativo usando JNI (Java Native Interface).

new: Se utiliza para crear un objeto de una clase (instanciar una clase).

package: Especifica el nombre del paquete al que pertenece una clase o interface.

private: Especifica que el atributo o método sólo puede ser accesible desde la clase en la que está declarado.

protected: Especifica que el atributo o método sólo puede ser accesible desde la propia clase, clases de su paquete y subclases.

public: Especifica que el atributo o método es visible y accesible desde cualquier lugar, incluso otros proyectos.

return: Retorna o devuelve (normalmente un valor) desde el método actual.

short: Tipo de dato primitivo que puede almacenar un número entero de hasta 16 bits.

static: Significa que el atributo, variable o método pertenece a la clase y no a la instancia (objeto). Es decir, su valor se comparte para todas las instancias.

strictfp: Garantiza que los cálculos de punto flotante sean exactamente los mismos en cada plataforma donde se ejecute la JVM.

super: Se utiliza para hacer referencia a la clase padre o al constructor de la clase padre del objeto actual.

switch: Se utiliza para definir una estructura de control de condiciones múltiples.

synchronized: Se usa para indicar que el método, o bloque de código deberá prevenir que no sean cambiados los objetos o recursos en procesos multihilo.

this: Se utiliza para hacer referencia al objeto actual o a su constructor.

throw: Se usa para lanzar una excepción directamente desde nuestro código.

throws: Especifica la o las excepciones que podría lanzar un método.

transient: Se utiliza para especificar que un atributo no se puede serializar y no hace parte del estado persistente de un objeto.

try: Declara un bloque de código que puede lanzar una excepción.

void: Indica que el método no retornará ningún valor.

volatile: Se utiliza para indicar que una variable puede cambiar de forma asíncrona muy frecuentemente y que debería ser leída sin ninguna optimización.

while: Se usa para definir una estructura de control cíclica que lleva su mismo nombe.