

Portfolio Programación

UD 4 - Objetos

Jhonal Jesús Roca Hernandez
Programación
1º DAW
2024 - 2025

Tabla de palabras reservadas de java

Elemento	Descripción	Valor por Defecto
INT	Representa un número entero (sin decimales). Usado para almacenar valores numéricos enteros.	0
DOUBLE	Representa un número decimal de doble precisión. Usado para almacenar valores numéricos con decimales.	0.0
FLOAT	Similar a double, pero con precisión simple. Menos preciso, pero usa menos memoria.	0.0f
STRING	Representa una cadena de caracteres (texto). Usado para almacenar textos o secuencias de caracteres.	null
BOOLEAN	Representa un valor verdadero (true) o falso (false). Usado para decisiones lógicas.	false
STATIC	Indica que un miembro (variable o método) pertenece a la clase, no a una instancia de la clase.	N/A (es un modificador, no tiene valor por defecto)
PUBLIC	Especifica que un método o variable es accesible desde cualquier clase. Modificador de acceso.	N/A (no tiene valor)
PRIVATE	Especifica que un método o variable sólo es accesible dentro de la misma clase. Modificador de acceso.	N/A (no tiene valor)
VOID	Especifica que un método no devuelve ningún valor. Usado en métodos que no retornan nada.	N/A (no tiene valor)
PROTECTED	Especifica que un método o variable es accesible dentro de la misma clase, clases derivadas y en el mismo paquete.	N/A (es un modificador)
CLASS	Define una clase. Es la plantilla para crear objetos (instancias) de ese tipo.	N/A (es una palabra clave, no tiene valor)

EXTENDS	Se utiliza para indicar que una clase hereda de otra clase (herencia).	N/A (es una palabra clave, no tiene valor)
NEW	Se utiliza para crear una nueva instancia de un objeto.	N/A (es una palabra clave, no tiene valor)
THIS	Hace referencia a la instancia actual de la clase. Se usa para diferenciar entre las variables de instancia y los parámetros del constructor.	N/A (es una palabra clave, no tiene valor)
SUPER	Hace referencia a la clase base o clase padre. Se usa para acceder a miembros de la clase padre.	N/A (es una palabra clave, no tiene valor)
RETURN	Se usa para devolver un valor desde un método.	N/A (dependiendo del tipo de retorno)

Tabla de los distintos métodos de java

Tipo de método	Descripción	Valor por defecto
Método estático (static)	Un método estático pertenece a la clase, no a las instancias de la clase. Se puede llamar sin crear un objeto. Se utiliza para operaciones que no dependen del estado de un objeto en particular.	N/A (no tiene valor por defecto)
Método de instancia	Un método de instancia pertenece a un objeto (una instancia de la clase). Necesita un objeto para ser llamado. Se usa para operaciones que dependen del estado del objeto.	N/A (se debe inicializar el objeto)
Método void	Un método que no devuelve ningún valor. Su propósito es realizar alguna acción o modificar el estado de un objeto sin retornar datos.	<code>void</code> (no tiene valor por defecto, no retorna nada)
Método con retorno	Un método que devuelve un valor de un tipo de dato específico (por ejemplo, <code>int</code> , <code>String</code> , <code>double</code>). Se usa cuando se necesita que el método realice cálculos y devuelva un resultado.	Depende del tipo de retorno (por ejemplo, <code>0</code> para <code>int</code> , <code>null</code> para <code>String</code>)
Metodo Main	Es el punto de entrada principal de una aplicación Java. Este método es siempre estático (<code>static</code>) y se utiliza para ejecutar el programa cuando se inicia la aplicación. Tiene la firma <code>public static void main(String[] args)</code> .	No tiene un valor por defecto (es el punto de inicio del programa)

método recursivo	Un método que se llama a sí mismo durante su ejecución. Se utiliza para resolver problemas que pueden dividirse en subproblemas más pequeños, como el cálculo de factoriales o la búsqueda en estructuras de datos como árboles.	N/A (depende de cómo se invoque la recursividad)
Método constructor	Es un tipo especial de método que se utiliza para inicializar un objeto cuando se crea. El nombre del constructor debe coincidir con el nombre de la clase y no tiene ningún tipo de retorno (ni siquiera <code>void</code>).	No tiene un valor por defecto (se ejecuta al crear el objeto)
Método get y set	Son métodos convencionales utilizados para obtener (get) y modificar (set) el valor de una propiedad o atributo de una clase. Los métodos get retorna el valor del atributo, mientras que los set modifican el valor del atributo.	N/A (depende de la implementación del atributo)
Método toString	Es un método especial heredado de la clase <code>Object</code> . Su propósito es devolver una representación en forma de cadena de texto del objeto. Se usa para imprimir o visualizar un objeto de forma comprensible.	Depende de la implementación del objeto. Si no se sobrescribe, retorna algo como
Método equals	Un método heredado de la clase <code>Object</code> que se utiliza para comparar si dos objetos son "iguales". Por defecto, compara las referencias de los objetos, pero se puede sobrescribir para comparar el contenido de los objetos	Compara las referencias por defecto (en otras palabras, compara si son el mismo objeto).
Método hashCode	Un método heredado de la clase <code>Object</code> que se utiliza para generar un valor hash único para un objeto, que es útil cuando el objeto se usa como clave en colecciones como <code>HashMap</code> o <code>HashSet</code> .	Por defecto, devuelve una representación de la dirección de memoria del objeto, lo que puede no ser único si no se sobrescribe.
Método Abstractos	No tiene implementación, debe ser implementado.	No tiene valor por defecto.

Ejercicios realizados en clase

<https://github.com/jrocher526/-Programacion>

