

SUMARIO

JAVA - PALABRAS CLAVE

También conocidas como **palabras reservadas**, son aquella que están definidas de manera exclusiva por el lenguaje de programación para un **propósito específico**. No se pueden utilizar estas palabras para darles otro comportamiento distinto al definido.

TIPO DE DATOS

boolean dato verdadero/falso.	byte número entero de 8 bits.	char dato valor unicode de 16 bits.	double número real en coma flotante 64 bits.
float número real en coma flotante 32 bits.	int número entero de 32 bits.	long número entero de 64 bits.	short número entero de 16 bits.

ESTRUCTURA DE DATOS

class para definir una clase.	extends permite indicar la clase padre de una clase.	implements para definir la/s interfaces de una clase.	import importa una clase que se encuentra en otro paquete o en la API de Java.
interface para declarar una interfaz.	package te permite agrupar un conjunto de clases e interfaces.		

MODIFICADORES

final indica que una variable no se puede modificar, un método no se puede redefinir o una clase no se puede heredar.	native indica que un método está en un lenguaje de programación dependiente de la plataforma.	protected indica que un elemento es accesible desde la clase donde se ha definido, subclases de ella y otras clases del mismo paquete.	private indica que un elemento es accesible únicamente desde la clase donde se ha definido.
public indica que un elemento es accesible desde cualquier clase.	static indica que un elemento es único en una clase y que se puede acceder al él sin tener que crear una instancia u objeto de la clase.	synchronized indica que un método o bloque de código es atómico.	volatile indica que el valor de un atributo que está siendo utilizado por varios hilos esté sincronizado.
transient para especificar que un atributo no sea persistente.	abstract para definir clases y métodos abstractos.		

ESTRUCTURAS DE CONTROL

CONDICIONALES	BUCLES	SALTO/RETORNO	EXCEPCIONES
if para instrucciones de control condicionales simples (if) o múltiples (else if).	for para escribir bucles que se repiten un número determinado de veces.	break interrumpe la ejecución de un bucle o de una instrucción.	try indica un bloque de código donde se quieren atrapar excepciones.
else instrucción de control condicional doble que indica qué hacer en caso de que no se cumpla la condición del (if) o el (else if).	while para escribir bucles que se repiten mientras una condición sea cierta. La condición se valida al principio.	continue interrumpe la ejecución de un bucle, pero permite seguir realizando interacciones.	catch cláusula de un bloque (try) donde se especifica una excepción.
switch instrucción de control condicional múltiple.	do para escribir bucles (do while) que se repiten mientras una condición sea cierta. La condición se valida al final.	return interrumpe la ejecución del método y puede devolver un valor de retorno.	finally permite especificar un bloque de código que siempre se ejecutará, se produzca o no una excepción.
case para indicar cada caso de una instrucción de control (switch).			throw permite lanzar una excepción.
default indica el caso por defecto de una instrucción de control (switch).			throws indica las excepciones que un método puede lanzar.

OTROS

assert para afirmar que una condición es cierta.	instanceof permite saber si un objeto es una instancia de una clase concreta.	strictfp indica que se tienen que utilizar cálculos en coma flotante estricto.	super permite invocar a un método o constructor de la superclase.
this para referenciar al objeto e invocar al constructor.	new para crear un objeto nuevo de una clase.	enum para definir tipos de datos enumerados.	override sirve para sobrescribir un método.
void dato vacío (sin valor).	const se usaba para la definición de constantes, pero ya no se utiliza, ahora se usa (final).	goto se usaba para moverse de un punto a otro del código, pero ya no se utiliza.	

SUMARIO

JAVA - TIPOS DE DATOS

DATOS PRIMITIVOS O SIMPLES

Son **entidades elementales** como números, caracteres, Verdadero/Falso. **No son objetos**, por tanto, no necesitan ser creados para usarse.

TIPO	NOMBRE	TAMAÑO	RANGO	VALOR DEFECTO
ENTERO	byte	8 bits	-128 a 127	0
	short	16 bits	-2^{15} a $2^{15}-1$	0
	int	32 bits	-2^{31} a $2^{31}-1$	0
	long	64 bits	-2^{63} a $2^{63}-1$	0
DECIMAL	float	32 bits	$\pm 3.40282347e^{+38}$	0
			$\pm 1.40239486e^{-45}$	0
	double	64 bits	$\pm 1.79769313486231570e^{+306}$	0
			$\pm 4.94065645841246544e^{-324}$	0
CARÁCTER	char	16 bits	\u0000 a \uffff	'\u0000'
LÓGICO	boolean	true / false	N.A.	False

Ejemplos de los más usados en Java:

- **int**: números de tipo entero (años, días de la semana)
- **float**: números decimales de hasta 7 dígitos (altura, peso)
- **double**: números decimales muy precisos, hasta 16 dígitos (área, longitud)
- **char**: conjunto de caracteres (códigos con letras y números)
- **boolean**: datos que sólo pueden ser verdadero o falso (par, primero)

DATOS REFERENCIA O COMPLEJOS

Son tipos de datos más complejos, que pueden estar formados por **varias variables, otros datos complejos y/o comportamientos** (funciones). **Son objetos**, por tanto, necesitan ser creados por el programador en caso de usarse

GRUPO	DESCRIPCIÓN	VALOR DEFECTO
BIBLIOTECA ESTÁNDAR JAVA	Son las clases que nos proporcionan la librería estándar de JAVA. (String, Date, ...)	null
PERSONALIZADOS/ DEFINIDOS POR EL DESARROLLADOR	cualquier clase creada por nosotros. clases customizadas/ personalizadas.	
ARRAYS	conjunto de elementos (arreglos). objeto que carece de métodos.	
LÓGICO	Byte Short Integer Long Float Double Character Boolean	

De ellos, el más utilizado es el tipo de dato **String** que permite manejar textos y cadenas de texto. El operador “+” permite concatenar cadenas:

```
String str1 = "Hola";  
String str2 = "Fernando";  
String str3 = str1 + " " + str2; //str3 sería "Hola Fernando"
```

SUMARIO

JAVA - MÉTODOS E INTERFACES

MÉTODOS

Son aquellos que definen el comportamiento de una clase o los objetos de esta.

setter permiten modificar los atributos privados.	getter permiten a los objetos retornar los valores de sus atributos privados.	main sirve para iniciar la ejecución de cualquier programa.	equals sirve para comparar dos objetos de una clase.
toString sirve para mostrar un texto con la información de un objeto.	compare compara dos objetos para la ordenación.	compareTo se usa para comparar dos cadenas.	

INTERFACES

Son una colección de métodos abstractos y atributos constantes en los que se especifica qué se debe de hacer pero no el cómo. Serán las clases hijas quienes definan el comportamiento al implementarlas.

set Define una colección que contiene elementos no repetidos y sin ordenar.	list define una colección de elementos ordenados que pueden estar duplicados.	queue colecciones de tipo cola, que no permiten el acceso aleatorio.	map Define una colección de parejas clave- valor, donde sus valores pueden ser duplicados pero sus claves serán únicas.
comparable permite implementar un criterio de orden natural.	comparator permite usar criterios diferentes al natural para ordenar elementos.		