



# Java Básico

## Introducción a Java

## Copyright

- Copyright (c) 2004  
José M. Ordax
- Este documento puede ser distribuido solo bajo los términos y condiciones de la Licencia de Documentación de javaHispano v1.0 o posterior.
- La última versión se encuentra en  
<http://www.javahispano.org/licencias/>

## **Historia del lenguaje**

- Creado por Sun Microsystems.
- Inicialmente orientado a la programación de microsistemas (proyecto OAK).
- Difundido en 1995 con una nueva orientación: Internet.
- Sintaxis muy similar a la de C++.
- En 2010, Oracle compró Sun Microsystems.

## **Características del lenguaje**

- |                        |                        |
|------------------------|------------------------|
| ○ Sencillo.            | ○ Arquitectura neutra. |
| ○ Orientado a Objetos. | ○ Portable.            |
| ○ Distribuido.         | ○ Altas prestaciones.  |
| ○ Interpretado.        | ○ Multithread.         |
| ○ Robusto.             | ○ Dinámico.            |
| ○ Seguro.              |                        |

## Sencillo

- Los creadores de Java se basaron en C++, pero eliminaron la mayoría de sus complejidades.
- No soporta tipos de datos: struct, union, y puntero.
- No soporta typedef ni #define.
- No permite la sobrecarga de operadores.
- No soporta la herencia múltiple.
- Posee una clase String, en vez del array de tipo char[] finalizado con nulo.
- Cuenta con un sistema automático para asignar y liberar memoria: el Garbage Collector.

## Orientado a Objetos

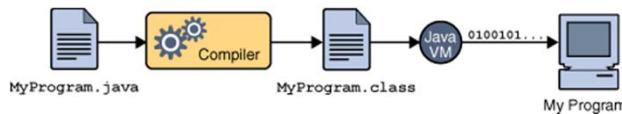
- Java es un lenguaje para el desarrollo de software Orientado a Objetos.
- Implementa los siguientes conceptos O.O.:
  - Herencia.
  - Encapsulación.
  - Abstracción.
  - Reutilización.
  - Etc....

## Distribuido

- Está concebido para trabajar en un entorno conectado en red.
- Cuenta con una amplia biblioteca de clases para comunicarse mediante TCP/IP: HTTP, FTP...
- Permite manipular con gran facilidad recursos vía URL.

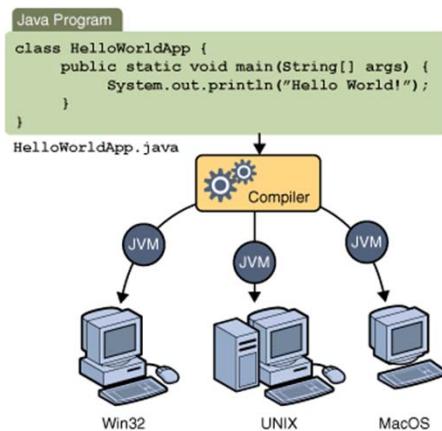
## Interpretado

- El compilador de Java traduce el código fuente a un código intermedio (bytecode).
- Los byetcodes son interpretados (ejecutados) en cualquier entorno donde exista un intérprete de Java.
- El intérprete de Java se llama Máquina Virtual Java o Java Virtual Machine (JVM).



## Interpretado (cont.)

- De ahí el famoso: Write once, run everywhere.



## Robusto

- Un software robusto es aquel que no se 'interrumpe' fácilmente a consecuencia de fallos.
- Un lenguaje de estas características suele tener mas restricciones a la hora de programar.
- No permite sobrescribir memoria y corromper otros datos mediante punteros.
- Facilita el manejo de excepciones.

## Seguro

- Por su naturaleza distribuida, el tema de la seguridad es muy crítico.
- Todos los navegadores poseen una 'sand box'.
- Existen tecnologías de firma digital para confiar en un determinado código Java.
- A su vez, existen políticas de seguridad para controlar de una manera mas precisa que puede o no puede hacer.

## Arquitectura neutra (portable)

- Los bytecodes son interpretados en cualquier plataforma donde exista una JVM.
- El uso de estándares como UNICODE, IEEE 754, etc... permite obtener los mismos resultados en todas las plataformas.

ASCII/8859-1 Text		Unicode Text	
A	0100 0001	A	0000 0000 0100 0001
S	0101 0011	S	0000 0000 0101 0011
C	0100 0011	C	0000 0000 0100 0011
I	0100 1001	I	0000 0000 0100 1001
I	0100 1001	I	0000 0000 0100 1001
/	0010 1111		0000 0000 0010 0000
8	0011 1000		0101 1001 0010 1001
8	0011 1000	天	0101 0111 0011 0000
5	0011 0101	地	0000 0000 0010 0000
9	0011 1001	ج	0000 0110 0011 0011
-	0010 1101	ل	0000 0110 0100 0100
l	0011 0001	ب	0000 0110 0011 0111
	0010 0000	م	0000 0110 0100 0101
t	0111 0100	ا	0000 0000 0010 0000
e	0110 0101	أ	0000 0011 1011 0001
x	0111 1000	ك	0010 0010 0111 0000
t	0111 0100	ي	0000 0011 1011 0011

## Altas prestaciones

- Existen compiladores JIT (Just-in-time) que compilan en el momento de la ejecución.
- A partir de la JVM 1.2.2 se introdujo un nuevo JIT llamado HotSpot.
- A pesar de todo, existe algún compilador 'real' de Java (perdemos la portabilidad y ganamos en rendimiento). Se encuentran en desuso.
- Informe Volano (comparativa de JVM):

<http://www.volano.com/report/>

## Multithread

- Soporta la ejecución de varias tareas a la vez.
- Posee una serie de clases que facilitan su utilización.
- También conocido con el término en castellano: 'multihilo'.

## Dinámico

- El código C++, a menudo requiere una recompilación completa si cambia una clase.
- Java emplea un método de interfaces para evitar estas dependencias y recompilaciones.
- Además, Java permite la indirección.

## La plataforma Java

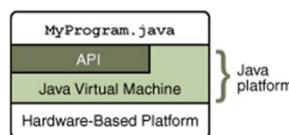
- Una plataforma es el entorno hardware y/o software donde se ejecuta un programa.
- Ejemplos:
  - Plataformas Intel, RISC, SPARC...
  - Plataformas Win32, Linux, AIX, Solaris, HP-UX, z/OS...
  - Plataformas IBM, Sun, HP, Microsoft...

## La plataforma Java (cont.)

- La plataforma Java es una plataforma solo software que se ejecuta sobre otra plataforma hardware/software.
- Tiene dos componentes:
- La Máquina Virtual Java (JVM).
- La Interfaz de Programación de Aplicaciones (API).

## La plataforma Java (cont.)

- La JVM es el intérprete Java.
- El API Java es un conjunto de clases ya desarrolladas que ofrecen un gran abanico de posibilidades al programador.



- Java SE: Java Platform Standard Edition.

## La plataforma Java (cont.)

- Existen distintas ediciones de la plataforma Java:
  - Java ME: Java Micro Edition.
  - Java SE: Java Standard Edition.
  - Java EE: Java Enterprise Edition.

