

Java

Introducción y Fundamentos del lenguaje de programación Java

Introducción



Introducción (I)

- Creado en 1991 por Sun Microsystems (James Gosling) para dispositivos electrónicos: calculadoras, microondas y la televisión interactiva.
 - Escasa potencia de cálculo
 - Poca memoria
 - Distintas CPUs
 - Consecuencias:
 - Lenguaje sencillo que genera código reducido
 - Lenguaje que independiente de la plataforma
-
-

Introducción (II)

- Lenguaje de programación para computadores/internet desde 1995.
 - Sun describe Java como “ *un lenguaje simple, orientado a Objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutral, portable, de alto rendimiento, multihilos, y dinámico*”
 - Ejecución de Java como:
 - Aplicación independiente
 - Applet (dentro del navegador al cargar la página Web)
 - Servlet (ejecutado en servidor de internet, sin interfaz gráfica).
-
-

Introducción (III)

- JDK (Java Development Kit): programas y librerías para desarrollar, compilar y ejecutar programas Java.
- Versiones:
 - Java 1.0: 212 clases organizadas en 8 paquetes
 - Java 1.1: 504 clases, 23 paquetes. Mejoras a la JVM.
 - Java 1.2 (Java 2): 1520 clases, 59 paquetes. Plataforma muy madura . Apoyado por grandes empresas: IBM, Oracle, Inprise, Hewlett-Packard, Netscape, Sun.
 - Java 1.3 (Java Enterprise Edition). Java comienza a ser una plataforma de desarrollo profesional
 - ??? Java MicroEdition



Características de Java



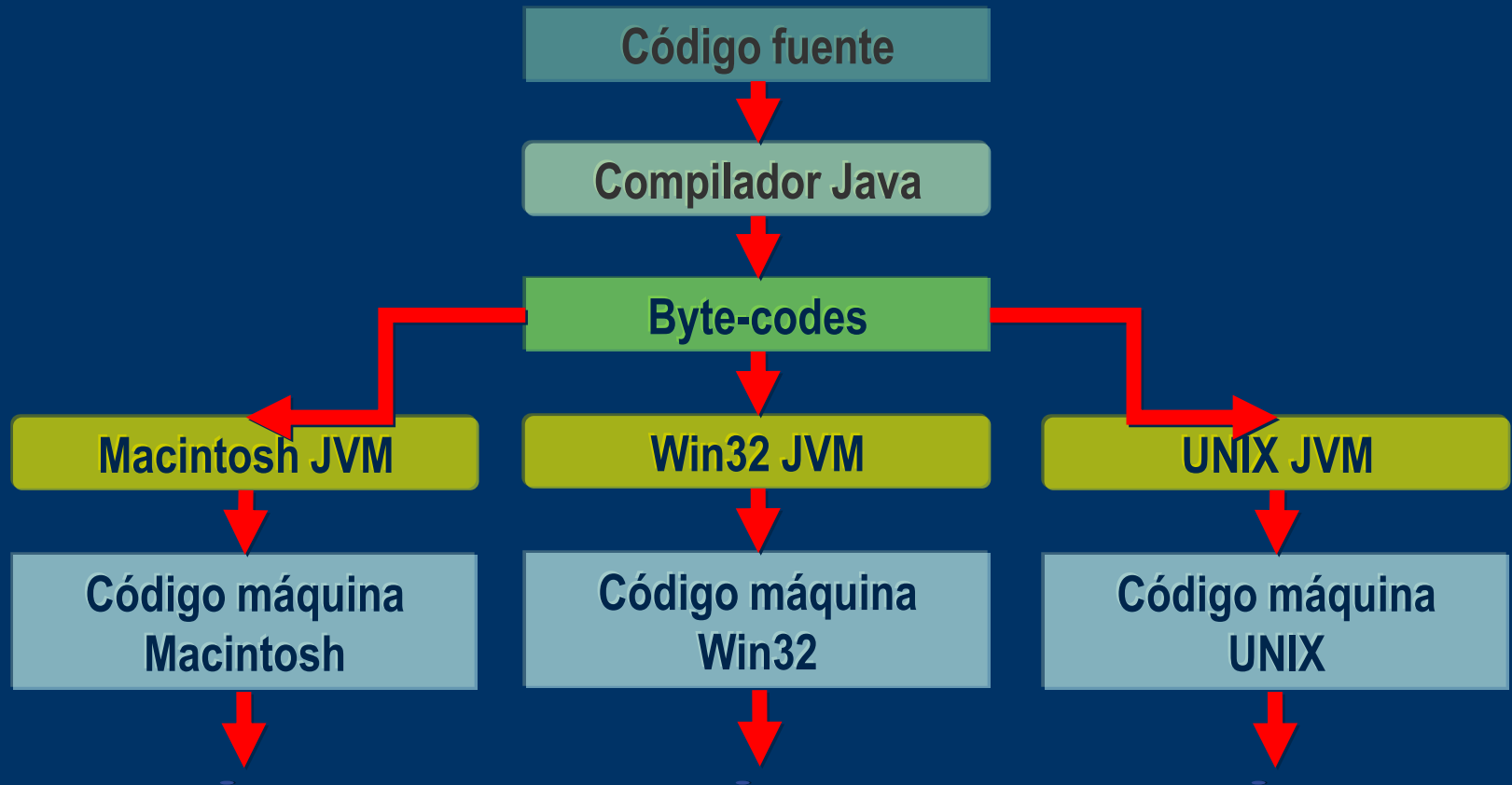
Características de Java I

- Lenguaje de fácil uso orientado a objetos
 - Lenguaje compilado e interpretado
 - Facilita un entorno interpretado:
 - Velocidad de desarrollo (no de ejecución)
 - Portabilidad del código
 - Ejecución multitareas
 - Cambios dinámicos en tiempo de ejecución
 - Seguridad del código.
-
-

La máquina Virtual Java - JVM (I)

- Es una máquina hipotética que emula por software una máquina real. Es definida por una especificación.
 - Conjunto de instrucciones de máquina
(C.O.) + Operandos
 - Pila
 - Memoria
 - ...
 - El compilador genera bytecodes (instrucciones de código máquina para la JVM)
 - El interprete traduce y ejecuta los bytecodes para cada máquina específica.
-
-

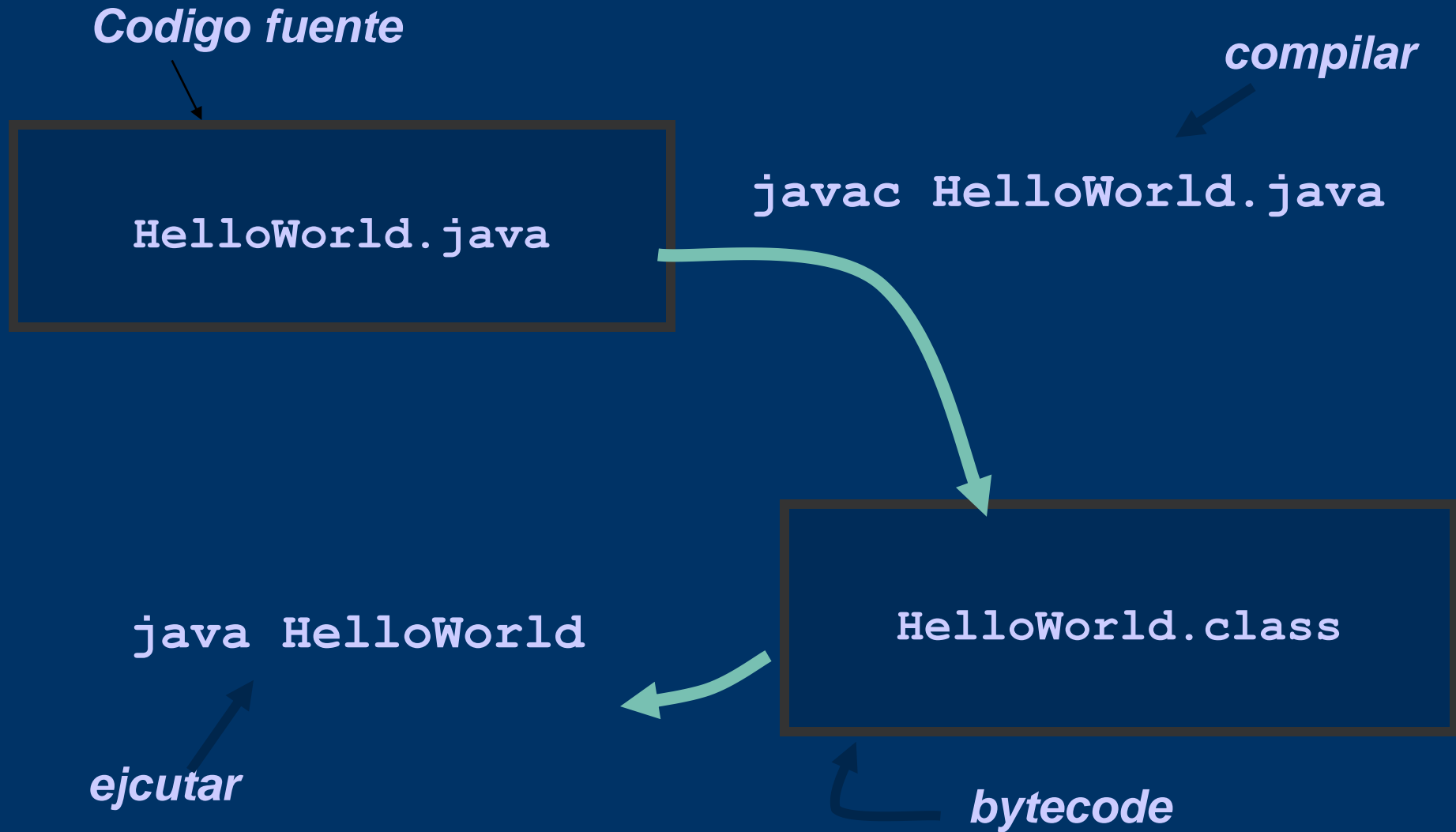
La máquina Virtual Java (II)



Compilador e intérprete de Java (I)

- El compilador analiza la sintaxis del código fuente (con extensión **.java**). Si no hay errores, genera *bytecodes*
 - > **javac** <Nombre>.java
- El intérprete es la Máquina Virtual de Java que ejecuta los bytecodes (*.class) creados por el compilador.
 - > **java** <Nombre> (sin extensión)
 - > **java** <Nombre> arg1 arg2 ...

Compilador e intérprete de Java (II)



Garbage Collection

- La memoria reservada dinámicamente que No se vaya a usar más debe ser liberada.
 - En otros lenguajes, esta liberación debe realizarla el propio programador.
 - La JVM dispone de un proceso que rastrea las operaciones de memoria: el *Garbage Collector*.
 - Verifica y libera la memoria que no se necesita
 - Se ejecuta automáticamente
 - Puede variar según la implementación de la JVM.
-
-

Seguridad del Código

- La JVM verifica los bytecodes asegurando que:
 - El código se ajusta a las especificaciones de la JVM.
 - No hay violaciones de acceso restringido.
 - Los tipos de los parámetros son correctos para todo el código.
 - No existen conversiones ilegales de datos (ej. de enteros a apuntadores).
-
-

Formato de los archivos fuente

- Declaración de paquete (opcional)
 - package <nombrePaquete>
- Instrucciones de importación(opcional)
 - import <nombrePaquetes>
- Declaración de clase o interfaz
 - [<mod>] class <NombreClase> {.....}



Un programa Simple

```
public class ProgramaSimple
{
    public static void main(String[] args)
    {
        System.out.println("Este es" + "un programa simple");
        System.exit(0);
    }
}
```