

# Tecnología de la Programación

## Tema 2. Introducción a Java Primera parte

### Historia

- ❑ A finales de los años 80 Sun Microsystems decide introducirse en el mercado de la electrónica de consumo.
  - ❑ Multiplicidad de procesadores en pequeños aparatos y electrodomésticos, con sus consiguientes entornos de programación o APIs (Application Programming Interfaces)
- ❑ (1990) Surge Oak como simplificación de C++ capaz de desarrollar interfaces gráficas sencillas e intuitivas y de adaptarse a situaciones con pocos recursos
- ❑ Oak duerme hasta la aparición de la Web (1995).
  - ❑ Ideal para la red de recursos distribuidos con múltiples tipos de computadoras
- ❑ Oak se rebautiza como Java, se distribuye gratuitamente y se integra con el navegador Netscape.

# Características del lenguaje Java

- ❑ Lenguaje orientado a objetos
  - ❑ Encapsulamiento, Herencia, Polimorfismo
  - ❑ Herencia simple
- ❑ Sintaxis similar a C++
- ❑ Lenguaje fuertemente tipado
  - ❑ No se puede asignar una variable de un tipo a otra de otro tipo distinto.
- ❑ Modelo de objetos más simple
  - ❑ Elimina manipulación directa de punteros
- ❑ Gestión de memoria más sencilla
  - ❑ Recolector de basura automático
- ❑ Arquitectura neutral
  - ❑ Independiente de la arquitectura física y del sistema operativo

Tema 2.1 - 3

# Filosofía de funcionamiento

- ❑ Java es una plataforma software
- ❑ Se basa en
  - ❑ el poder de las redes de computadoras
  - ❑ la idea de que el mismo software debe poderse ejecutar en diferentes clases de computadoras y otros objetos tecnológicos de consumo
    - ❑ Write Once, Run Everywhere

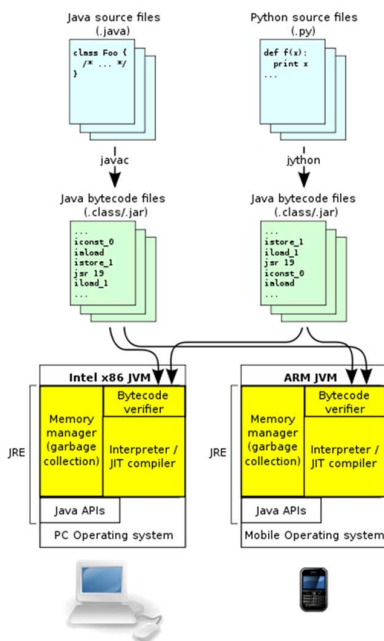
Java funciona en cualquier parte, desde PCs a TVs o teléfonos,  
e incluso tarjetas con chips incorporados (*smart cards*)

Tema 2.1 - 4

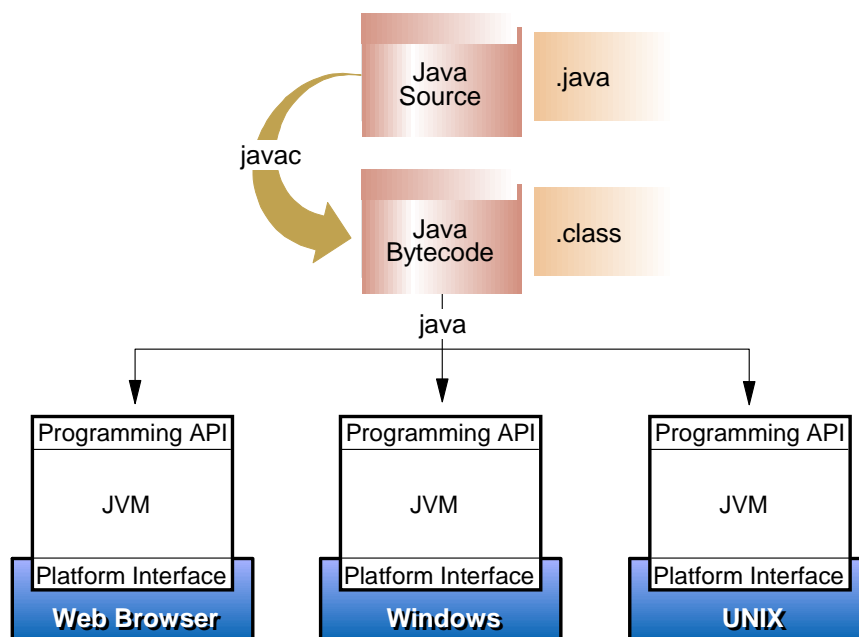
# Filosofía de funcionamiento

## □ Idea

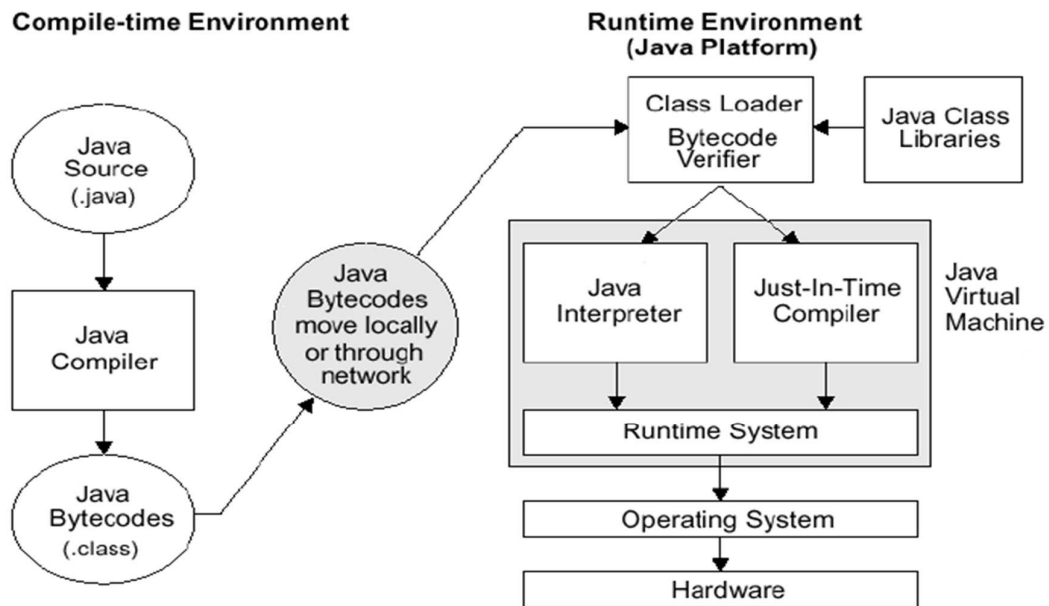
- El software Java se ejecuta en muchas y variadas plataformas gracias a la *Java Virtual Machine* (JVM) o máquina virtual Java (MVJ)
- MVJ es un traductor que transforma las instrucciones Java independientes de la plataforma (*bytecode*) en órdenes que hacen funcionar los dispositivos concretos
- La MVJ es propia de cada plataforma.



# Filosofía de funcionamiento



# Filosofía de funcionamiento



7

## Plataforma Java

### ❑ J2SE (Java Standard Edition)

#### ❑ JRE (Java Runtime Environment) => Entorno de ejecución de Java

- ❑ JVM + APIs estándar (conjunto de bibliotecas)

#### ❑ JDK (Java Development Kit) => Entorno de desarrollo de Java

- ❑ Contienen un compilador y un JRE
- ❑ javac: compilador
- ❑ java: lanza el JRE para ejecutar los bytecodes

### ❑ Existen otras tecnologías Java:

- ❑ J2ME (Java Micro Edition) : móviles
- ❑ J2EE (Java Enterprise Edition): aplicaciones distribuidas
- ❑ Java FX, Java Card, Java TV, ...

## Bibliotecas estándar

- ❑ `java.lang`: Incluye clases como `String`.
- ❑ `java.io`: Para la entrada y salida a través de flujos de datos.
- ❑ `java.util`: Contiene colecciones de datos y clases, el modelo de eventos, facilidades horarias, generación aleatoria de números, ...
- ❑ `java.applet`: Creado para soportar la creación de applet.
- ❑ `java.awt`: La Abstract Window Toolkit contiene rutinas para el desarrollo de interfaces gráficas de usuario. Este paquete también contiene la API de gráficos
- ❑ `javax.swing`: Es una colección de rutinas que se construyen sobre `java.awt` para suministrar un toolkit de widgets independiente de plataforma.

Tema 2.1 - 9

## El entorno de programación en Java

### El entorno mínimo

- ❑ El entorno mínimo consta de compilador y máquina virtual, y funciona en línea de órdenes del sistema operativo
- ❑ El compilador se invoca por medio de la instrucción `"javac MiClase.java"`
- ❑ El compilador produce un archivo en bytecode con el nombre `"MiClase.class"`
- ❑ La máquina virtual se invoca por medio de la instrucción `"java MiClase"` e interpreta el bytecode
- ❑ Si se trata de un applet, éste se incrusta en una página Web (`"página.html"`) y se usa un navegador o el `appletviewer` por medio de la instrucción `"appletviewer página.html"`

Tema 2.1 - 10

## Estructura de un programa Java (1)

```
package nombre.del.paquete;

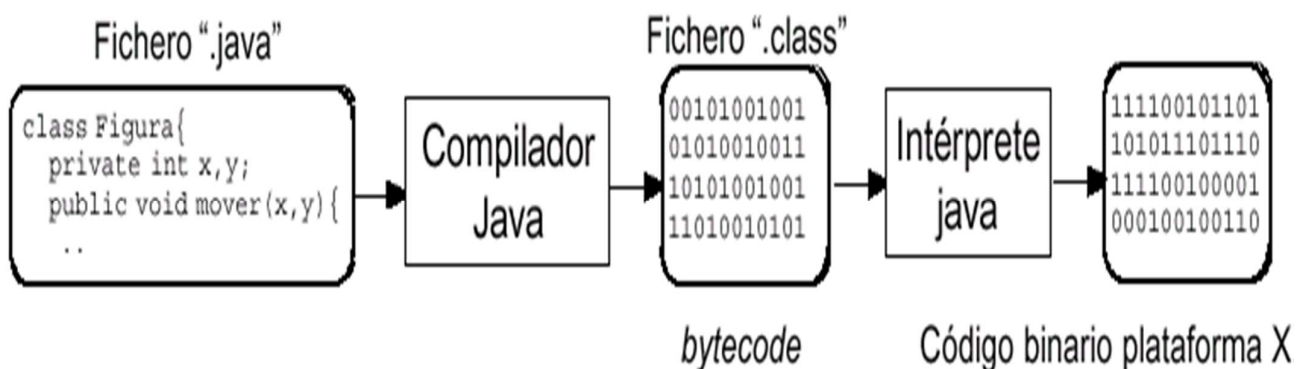
// Sección de importación de otras clases o paquetes
import java.util.Vector;
import java.util.Date;

// Declaración e implementación de la clase
public class NombreClase
{
    ...
}
```

Tema 2.1 - 11

## Estructura de un programa Java (2)

- ❑ Un programa Java consiste en una o más definiciones de clase, pero no pueden definirse funciones sueltas. Deben pertenecer a una clase.
- ❑ Cada clase genera un fichero compilado .class
- ❑ Un fichero fuente .java es una unidad independiente de compilación y puede contener una o más definiciones de clase



Tema 2.1 - 12

## Estructura de un programa Java (3)

- ❑ En un fichero fuente, al menos una de las clases public debe llamarse igual que el fichero fuente
- ❑ Aunque varias definiciones de clase se encuentren en un mismo fichero, se generará un .class compilado para cada clase.
- ❑ Si hay 8 clases en el fichero, se generar 8 ficheros .class

## Estructura de un programa Java (4)

- ❑ Ficheros .class
- ❑ Los ficheros .class pueden equipararse a los .obj generados en otros lenguajes compilados, pero además incluyen otras funcionalidades.
- ❑ En otros lenguajes como C y C++ se precisa de ficheros de cabecera con las declaraciones de las funciones y las definiciones de tipos. En Java no son necesarios. Toda la información de tipos está dentro del propio .class.
- ❑ Formas de ejecución de programas Java
  - ❑ Como aplicación tipo consola (entrada/salida standard)
  - ❑ Como aplicación tipo ventanas (Windows, Mac...)
  - ❑ Como Applet incrustado en un navegador Web

## Primer programa (1)

HolaMundo.java

```
// Este es mi primer programa en Java

public class HolaMundo
{
    public static void main(String[] args)
    {
        System.out.println("Hola mundo");
    }
}
```

Tema 2.1 - 15

## Primer programa (2)

❑ Lo que tenemos es una definición de la clase HolaMundo

❑ Es una clase pública (accesible desde fuera del paquete).

❑ Dentro de la clase se define un único método:

```
public static void main(String[] args)
{
    System.out.println("Hola mundo");
}
```

❑ También el método es público.

❑ Es un método de clase (static) y no devuelve nada (void).

❑ El objeto estático out de la clase System recibe el mensaje println() con la cadena que se quiere mostrar.

❑ El objeto out representa la pantalla, lugar donde se imprime la cadena.

Tema 2.1 - 16



## Gestión de ficheros y directorios

- ❑ Java distingue minúsculas y mayúsculas
- ❑ Cada fichero “.java” sólo puede contener una clase public
- ❑ Cada fichero “.java” debe corresponder al nombre de la clase public dentro del fichero
- ❑ El compilador genera un fichero .class para cada clase del fichero fuente
- ❑ El compilador y el intérprete deben poder encontrar las librerías de clases utilizadas (import)
- ❑ Para ello es necesario establecer el valor de la variable de entorno CLASSPATH o indicarlo al compilar y ejecutar

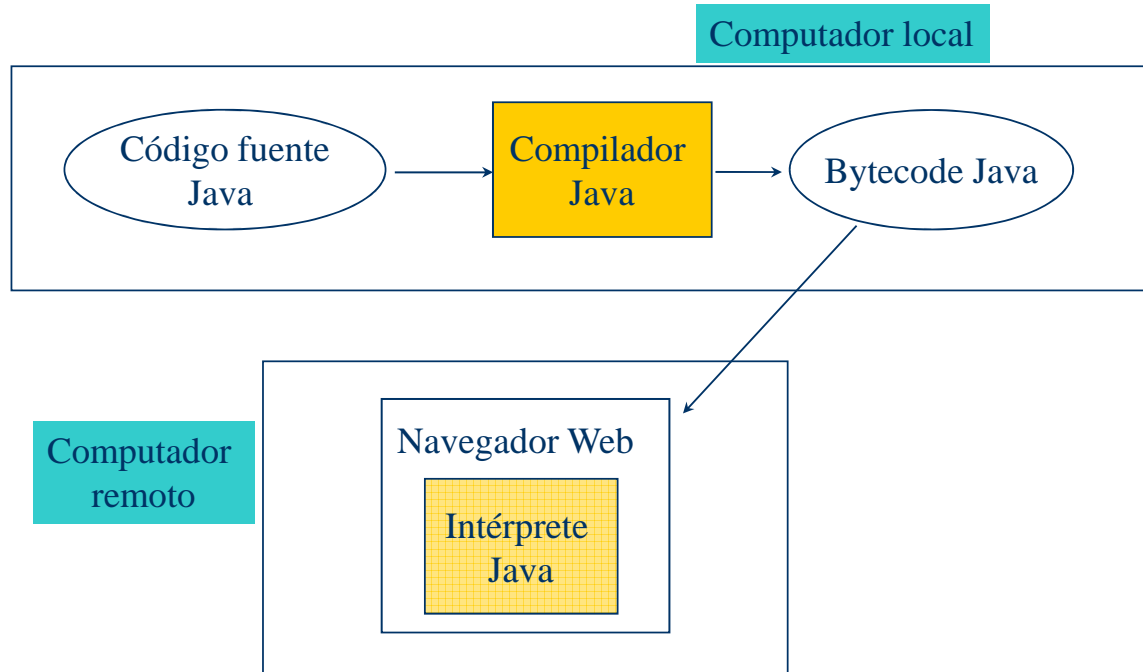
Tema 2.1 - 17

## Ejecución de programas: Aplicaciones

- ❑ Código autosuficiente interpretable por la máquina virtual
- ❑ No precisan, pero pueden incluir interfaces gráficas de usuario
- ❑ Compilar
  - `javac -d DIRECTORIO_DE_DESTINO_DE_BINARIOS \`  
`-cp PATH1 ; PATH2 ;...; PATH_N \`  
`NOMBRE_DEL_FICHERO_JAVA`
- ❑ Ejecutar
  - `java -cp DIRECTORIO DE CLASES mipaquete.MiClase`
    - ❑ MiClase debe contener el método main que se ejecutará
    - ❑ Otras clases pueden tener método main
      - ❑ útil para probar clases por separado

Tema 2.1 - 18

## Ejecución de programas: Applet



Tema 2.1 - 19

## Ejecución de programas: Applet

### ❑ Código fuente Java

- ❑ Código diseñado para ser incluido dentro de páginas Web
- ❑ Imprescindible incluir interfaz gráfica de usuario

```
import java.awt.*;
import javax.swing.*;
public class HolaMundoApplet extends JApplet
{
    public void paint(Graphics g)
    {
        g.drawString("Hola, mundo", 25, 50);
    }
}
```

Tema 2.1 - 20

## Página web para Applet

### **holaMundo.html**

```
<HTML>
<HEAD>
  <TITLE>Ejemplo de Hola, Mundo</TITLE>
</HEAD>
<BODY>
  <APPLET CODE="HolaMundoApplet.class" WIDTH=100   HEIGHT=100>
</APPLET>
</BODY>
</HTML>
```

Tema 2.1 - 21

## Ejecución de programas: Applet

- ❑ Compilación y ejecución en línea de comandos:
  - ❑ C:\java\ejemplos\Holamundo> javac HolaMundoApplet.java
  - ❑ C:\java\ejemplos\Holamundo> appletviewer holaMundo.html (Navegador)

Tema 2.1 - 22