

# Seminario de Java



Programación Orientada a Objetos

## Contenido

1. Introducción
2. Primeros pasos con Java. El entorno JBuilder
3. La sintaxis del lenguaje Java
4. Clases y Objetos
5. Cadenas y Entrada/Salida
6. Herencia, clases abstractas
7. Genericidad e interfaces
8. Colecciones e iteradores
9. Clases anidadas e internas
10. Hilos

# Contenido

1. **Introducción**
2. **Primeros pasos con Java. El entorno JBuilder**
3. La sintaxis del lenguaje Java
4. Clases y Objetos
5. Cadenas y Entrada/Salida
6. Herencia, clases abstractas
7. Genericidad e interfaces
8. Colecciones e iteradores
9. Clases anidadas e internas
10. Hilos

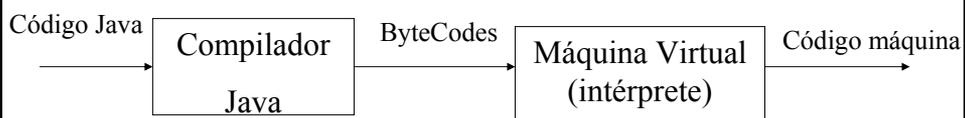
3

## 1. Introducción.

- Sun Microsystems, Agosto 1995
- Navegadores con contenido interactivo
- Tecnología de implementación:
  - ByteCode + Máquina Virtual
  - Compilado e **interpretado**



{ - portable  
- lento



Unix, Linux, Windows, Macintosh, ...

4

- **Orientado a Objetos**
  - encapsulación, herencia, polimorfismo y ligadura dinámica
- **Parecido a C++**
  - elimina las partes confusas ej: punteros
  - añade nuevas propiedades ej: *garbage collector*
- **Robusto**
  - comprobación estática de tipos
  - maneja la memoria por el programador
  - comprobación de límites de arrays y excepciones.
- Proporciona las librerías y herramientas para **programas distribuidos** (protocolos http y ftp, interconexión TCP/IP)
- **Multithreads** (ejecución concurrente)

¿Qué se mueve alrededor de Java?

- **JDBC API** *Java Database Connectivity*
- **Java RMI** *Remote Method Invocation* aplicaciones distribuidas
- **Java IDL** *Interfaz Definition Language*. Puente de compatibilidad con el modelo estándar de objetos CORBA
- **JavaBeans** Especificación de componentes basado en Java

## 2. Primeros pasos con Java

<http://java.sun.com>

- **SDK** (*Software Development Kit*)
  - Necesario para crear (desarrollar y compilar) programas
- **JRE** (*Java Runtime Environment*)
  - Necesario para ejecutar un programa Java
  - SDK = JRE + herramientas de desarrollo
- La tecnología Java se organiza en subáreas:
  - **J2SE** (*Java 2 Platform, Standard Edition*)
  - **J2EE** (*Java 2 Platform, Enterprise Edition*)
    - SDK + soporte para servicios web y componentes
  - **J2ME** (*Java 2 Platform, Micro Edition*): entorno flexible y robusto para aplicaciones que funcionan en dispositivos tales como teléfonos móviles, PDA, ...
  - **Java Web Services**: aplicaciones basadas en la web que utiliza estándar XML y protocolos de transporte para intercambiar datos con los clientes que le invocan.

7

## Compilación y ejecución en la línea de comandos

- Compilación:

```
c:\javac fichero.java   fichero.class
```

- Ejecución:

```
c:\java fichero [argumentos]
```

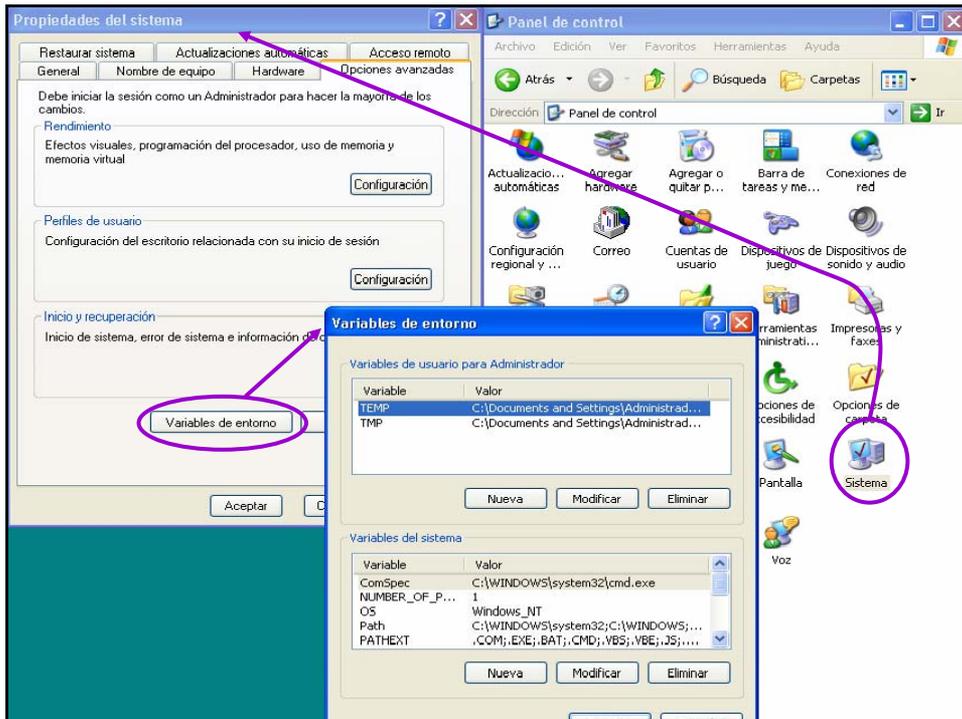
- `fichero.java` es un fichero de texto que contiene la clase principal (clase que contiene el main)
  - `public static void main (String [] args)`

8

# VARIABLES DE ENTORNO

- **JAVA\_HOME**
  - Directorio en el que está instalado el J2SE SDK
  - Por ejemplo, `c:\set JAVA_HOME = c:\JBuilderX\jdk1.4`
- **CLASSPATH**
  - Ruta de acceso a las API's de Java
  - `c:\set CLASSPATH=.;%JAVA_HOME%\lib\tools.jar;  
%JAVA_HOME%\lib\dt.jar;`
  - `..;` Carpeta o directorio actual de trabajo
- **PATH del sistema**
  - Directorio donde se encuentran los ejecutables de java
  - `c:\set PATH=%JAVA_HOME%\bin;%PATH%`
- Cambiar las variables de entorno en Windows
  - Panel de Control >> Sistema >> Opciones avanzadas >> Variables de entorno

9



## Ejercicio: Compila y ejecuta desde la línea de comando

```
//Fichero de texto Hola.java

public class Hola {
    public static void main(String[] args) {
        System.out.println("HOLA JAVA AMIGO "+args[0]);
    }
}
```

11

## Ejecutables en Java

- Existen algunos compiladores pero se pierde la portabilidad
- Soluciones:
  - Crear un **.bat** ejecutable que contenga la llamada:

```
java nombre_clase_prinpal
```
  - Crear un **.jar** ejecutable
    - Comprimir ficheros de la aplicación en un JAR
    - Modificar la metainformación indicando la clase principal:
      - META-INF/MANIFEST.MF
      - añadir la entrada: `Main-Class: clasePrincipal`
    - Sintaxis creación: **jar** cvf destino.jar \*.class  
c = nuevo; v = verbose; f = indicamos el nombre destino
    - Ejecutar: `java -jar destino.jar`

12

# Ficheros jar y Manifest

- Definir un fichero con la nueva entrada de Manifest (llámese `myManifest.mf`)

```
Main-Class: clasePrinpal
```

- Crear el fichero jar utilizando un fichero Manifest existente y comprimiendo todos los ficheros del directorio `classes`:

```
jar cvfm destino.jar myManifest.mf -C classes/ .
```

- Los ficheros se deben colocar en el mismo orden en el que se ponen los modificadores "f" y "m"
- Ejecutar: `java -jar destino.jar [argumentos]`

13

Barra de herramientas del Proyecto

Compilar y Ejecutar

Generar el proyecto

Panel de Proyecto

Errores previos a la compilación

Panel de Estructura

Panel de Contenido

```
package prueba;
public class Hola {
    public static void main (String [] args) {
        System.out.println(" HOLA JAVA AMIGO "+ args[0]);
    }
}
```

Mensajes

```
C:\Programas\NetBeans\jdk1.4\bin\javaw -classpath "C:\pruebas\Prueba\classes;C:\JBuilderX\jdk1.4\demo\jfc\Java2D\Java2Demo.jar;C:\JBuilderX\jdk1.4\demo\p
HOLA JAVA AMIGO Bego
```

Proceso finalizado.

# Elección de la clase principal en JBuilder

1 Menú Proyecto

2

3

4

Clase que contiene el main

Parámetros de la aplicación

15

# Recopilatorios en JBuilder

Equipo Asistentes Herramientas Ventana Ayuda Comprar

- Implementar interfaz...
- Modificar métodos...
- EJB
- Usar módulo de datos...
- Usar interfaz CORBA...
- Recursos de cadenas...
- Configuración de Cactus...
- Creador de compiladores...**
- Creador de ejecutables nativos...
- Javadoc...
- Tarea externa de generación
- Ant
- Exportar a Ant
- Importar código fuente...

16

**Creador de recopilatorios - Paso 1 de 5**

**Seleccione un tipo de recopilatorio**

A continuación se muestran las características del recopilatorio seleccionado.

Tipo de recopilatorio: **Básico**

Descripción:  
Los recopilatorios de propósito general están descomprimidos por defecto y habitualmente no incluyen bibliotecas.

< Atrás    **Siguiente >**    Finalizar    Cancelar    Ayuda

**Creador de recopilatorios - Paso 2 de 5**

**Indiquen el archivo que deberá crearse**

El asistente utilizará los parámetros de crear un nuevo nodo de proyecto para el recopilatorio. Puede cambiar los parámetros accediendo al cuadro de diálogo contextual del nodo del recopilatorio.

Tipo: Básico

Nombre: **Recopilatorio**

Archivo: **U:\SeminarOS-ObjBuilder\_projects\SeminarOS\Hola.jar**

Documentación

Incluir la documentación del proyecto en el recopilatorio

Nombre del archivo: **documentación**

Comprimir el contenido del recopilatorio

Crear siempre un recopilatorio al generar el proyecto

< Atrás    **Siguiente >**    Finalizar    Cancelar    Ayuda

**Creador de recopilatorios - Paso 5 de 5**

**Establecer las opciones del descriptor del recopilatorio**

La mayoría del tiempo será suficiente el proceso de creación automática de descripciones, pero puede que desee guardar una copia del descriptor generado. Sólo los usuarios avanzados tendrán necesidad en alguna ocasión de modificar el contenido de un descriptor.

Incluir un descriptor en el archivo

Configuración del descriptor

Crear un descriptor

Crear un descriptor y guardar una copia en el archivo

Vía de acceso:

Usar el archivo especificado como archivo descriptor

Vía de acceso: **main0src\META-INF\MANIFEST.MF**

< Atrás    **Siguiente >**    Finalizar    Cancelar    Ayuda

**Nombre dentro del JBuilder**

**Mantener la etiqueta: Main-Class**