

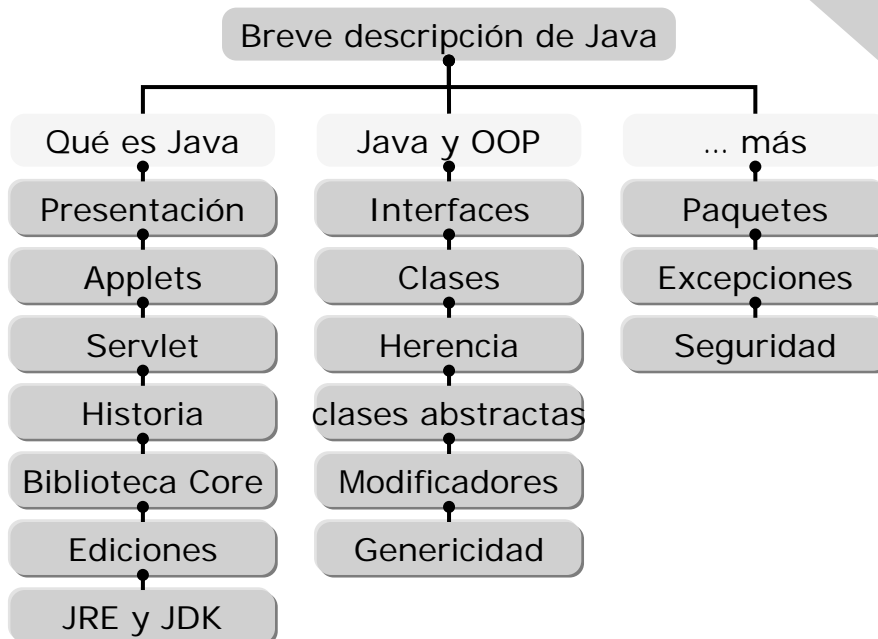
## Una breve descripción de Java

Sistemas Distribuidos – ITInformática (UVA)

César Llamas Bello – © 2003

(¡Si busca un curso sobre Java, no es éste!

Busque en otro sitio porque *seguro* que ya lo ha hecho algún otro.)



## ¿Qué es Java?

- ❑ Un lenguaje de programación con elementos de C, C++ y otros lenguajes,
  - ❑ ... Unido a un **gran** conjunto de bibliotecas,
  - ❑ ... Con un soporte de ejecución estándar y una máquina virtual.
- 
- ❑ Ventajas: simple, oo, distribuido, seguro, neutro, multitenhebrado, dinámico.

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

3

## ¿Qué es Java?

- ❑ Un programa Java es una biblioteca de clases (bytecode) organizada en paquetes jerárquicos.
  - Programar es incrementar la biblioteca clase a clase
- ❑ Un programa Java es una «sopa» de objetos organizada en una jerarquía de herencia.
  - Esta «sopa» es dinámica, y es
  - ... distribuida

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

4

## ¿Qué es Java?

- ❑ La máquina virtual java es una opción que garantiza la portabilidad.
- ❑ Existen tres sabores básicos de aplicaciones.
  - Aplicación tipo “standalone”, sobre una máquina virtual (JVM).
  - Aplicación tipo “applet”, lista para descargarse en su JVM tipo “sandbox”.
  - Aplicación tipo “servlet”, que se ejecuta en un contenedor de servlets (p. ej. Tomcat)

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

5

## ¿Qué es un applet Java?

- ❑ Es una aplicacioncilla en Java con un API limitado (arenero), pues...
  - No puede leer/escribir archivos
  - No pueden hacer cualquier llamada al sistema, ...
- ❑ ... Que se descarga desde el sitio Web para ejecutarse en una máquina virtual en la plataforma del cliente:
  - En el cliente web
  - A parte (*appletviewer*)

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

6

## ¿Qué es un applet?

```
/* applet sencillo, por M. Liu 1/8/02 */
import java.applet.Applet;
import java.awt.*;

public class MiApplet extends Applet {
    public void paint (Graphics g) {
        setBackground(Color.blue);

        font chulo = new Font("Arial", Font.BOLD, 40);
        g.setFont(chulo);
        g.setColor(Color.yellow);
        g.drawString("Hola Mundo", 100, 100);
    } // fin de paint
} // fin de la clase
```

- ❑ Los applets ofrecen un entorno de ejecución seguro.

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

7

## ¿Qué es un servlet?

- ❑ Objetos que se ejecutan en una máquina remota, e interactúan mediante un protocolo petición-respuesta.
- ❑ Necesita el soporte de un contenedor de servlets. (De la versión adecuada)
- ❑ Hay dos sabores de servlets:
  - Servlet (*servlet vulgaris*).
  - JSP (Java Server Page).  
Una mezcla excitante de HTML y código Java incrustado.  
Al ser usada la página, el contenedor la compila y genera un *servlet vulgaris*.

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

8

```

/* servlet sencillo, por M. Liu 1/8/02 */
import java.io.*;
import java.text.*;
import java.util.*;
import java.servlet.*;
import java.io.http.*;

public class MiServlet extends HttpServlet {
    public void doGet(HttpServletRequest petition,
        HttpServletResponse respuesta)
        throws ServletException, IOException {
        PrintWriter salida;
        String titulo = "Resultado de MiServlet";
        String algo;
        respuesta.setContentType("text/html");
        salida = respuesta.getWriter();
        algo = "<HTML><HEAD><TITLE>\n" + titulo +
            "</TITLE></HEAD><BODY>\n <H1>" + titulo + "</H1>\n" +
            "<P>Hola Mundo</P>\n" + "</BODY></HTML>";
        salida.println(algo); salida.close();
    } // fin doGet
} // fin de la clase

```

¿Veis que pocas palabras clave aparecen?

## ¿Qué es Java?

## Historia

- ❑ Comienza en 1991 en el Green Project
  - Como un lenguaje para sistemas embebidos,
  - Que substituyera a C++ → Oak
- ❑ Aprovecha el tirón del Web sobre 1995
  - Para dar contenido dinámico
  - ... Y aparece HotJava, con sus applets.
- ❑ Hitos:
  - JDK 1.0, Enero 1996
  - JDK 1.1, Febrero 1997
  - JDK 1.2 (Java 2), Diciembre 1998
  - J2SE 1.3 (Java 2 Standard Edition), Mayo 2000
  - JSE 1.4 , Febrero 2002

## ¿Qué es Java?

## Biblioteca Core

- ❑ java.lang... Clases de muy bajo nivel
- ❑ java.util... Clases útiles (arrays, sets, ...)
- ❑ java.text Texto y formateo
- ❑ java.math Calculo de precisión arbitrario
- ❑ java.io E/S de streams de bytes,...
- ❑ java.nio (idem) pero nueva
- ❑ java.net Red
- ❑ java.rmi... Invocación de métodos remotos
- ❑ org.omg... CORBA
- ❑ java.awt.... Gráficos, ventanas, ...
- ❑ javax.swing... (idem) pero mejor organizado
- ❑ java.beans... Para desarrollar componentes reusables
- ❑ Java.applet Soporte de applets
- ❑ Java.sql Soporte JDBC
- ❑ Java.security... Seguridad, certificados, ...

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

11

## ¿Qué es Java?

## Otras bibliotecas

- ❑ El Java API 4 tiene 135 paquetes y 2738 clases/interfaces
  - JavaBeans Activation Framework
  - InfoBus intercambio entre JavaBeans
  - Java Communications Serie y paralelo...
  - Java Naming and Directory Interface: LDAP
  - JavaMail
  - JavaHelp
  - JavaServlet
  - Java.Cryptography
  - javax. ... Menos (lo ya comentado de Swing)
  - Java 3D Gráficos 3D
  - ...

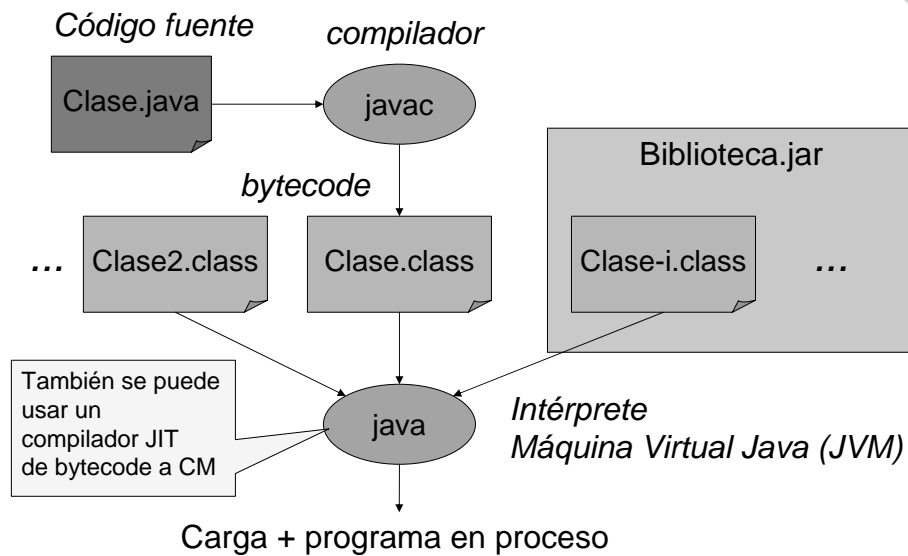
22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

12

## ¿Qué es Java?

## Application Binary Interface



22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

13

## ¿Qué es Java?

## Ediciones

- ❑ Plataforma Java 2, Standard Edition (J2SE)
  - Java ordinario
- ❑ Plataforma Java2, Enterprise Edition (J2EE)
  - Para aplicaciones de servidor de categoría empresarial.
    - Enterprise JavaBeans (Arquitectura de componentes), JavaServer Pages, JMS
- ❑ Plataforma Java2, Micro Edition (J2ME)
  - Para sistemas embebidos (empotrados) y electrónica de consumo
    - Soportan parte del API, incluyen API para desarrollo a bajo nivel.

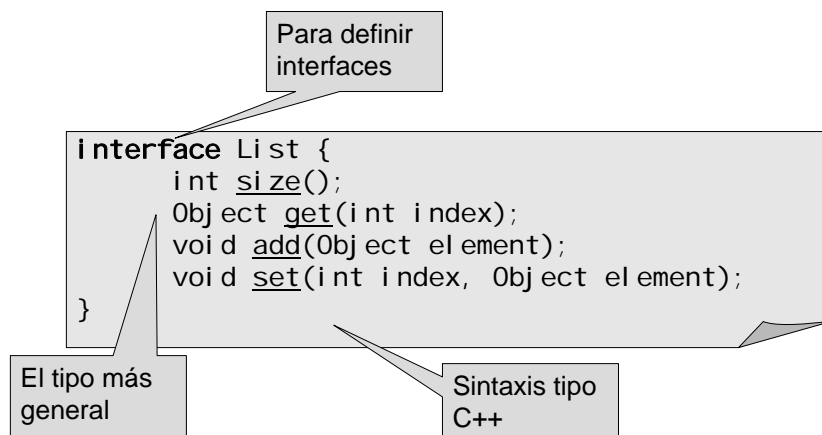
22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

14

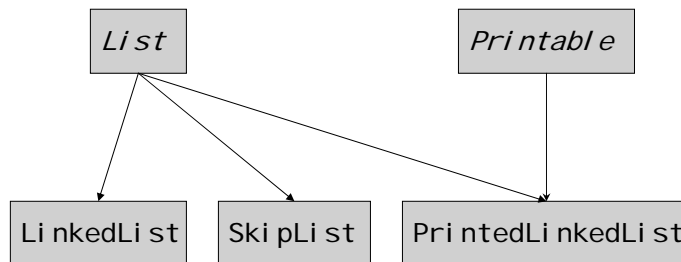
- ❑ Java se distribuye en 3 formatos:
  - JRE (Java Runtime Environment) solo para ejecutar programas Java (tiene las clases Core).
  - JDK (Java Development Kit) contiene **además** compiladores necesarios.
  - Plug-in para instalar en los clientes web.
- ❑ Además es interesantes bajarse la documentación en formato JavaDoc.
- ❑ La dirección es <http://java.sun.com>
  - También IBM tiene su sistema <http://oss.software.ibm.com/developerworks/opensource/jikes/>

- ❑ Interfaces
  - Java permite diseñar interfaces





- ❑ Las interfaces permiten herencia, y se pueden extender
- ❑ *Son tipos de datos*
- ❑ Una clase puede implementar muchas interfaces.



- ❑ Clases
  - Con sus campos y métodos

```

class ArrayList implements List {
    Object[] elements;
    int length;

    int size(){ return length; }
    Object get(final int index) { return elements[index]; }
    void add(final Object element); { ... }
    void set(final int index, final Object element) { ... }

    ArrayList() { ... }
    ArrayList(int capacity) { ... }
}
    
```

## Java y OOP

- ❑ Permite cualificar los campos, clases y métodos: `public`, `protected`, `[nada]`, `private`
- ❑ Los objetos se crean instanciando (`new`)
  - Que devuelve una referencia (no un puntero)
- ❑ Las llamadas son al estilo:  
`lista.add(...)`
  - Las variables de tipo de clase son referencias a objetos
  - `"=="` compara punteros

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

19

## Java y OOP

### *Cosas sobre clases*

- ❑ Tiene polimorfismo
- ❑ Tiene la referencia `"this"`
- ❑ Los constructores se denominan igual que la clase (`void`)
- ❑ No hay destructores, sólo `"finalize()"`
  - el espacio se toma del "montón" y
  - se dispone del espacio de un objeto cuando desaparecen sus referencias.

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

20

- ❑ Java permite herencia de implementación,
  - Extendiendo una clase
  - Permite la re-implementación aunque puede prohibirse
  - Y la adición de métodos y campos

```
public class Graphi cCi rcl e extends Ci rcl e {
    private Color outl i ne, fi ll ;
    Graphi cCi rcl e( . . . . ) {
        super( . . . );
        thi s. outl i ne= . . . ;
    }
    . . .
}
```

- ❑ Se permiten clases abstractas “abstract”
  - Existen algunos métodos implementados, y Otros métodos abstractos *no*.
  - Pueden tener datos
  - No pueden instanciarse directamente, Pero pueden extenderse.
- ❑ **Java no admite herencia múltiple.**
  - Solo las interfaces tienen herencia múltiple.

- ❑ Modificadores de campos:
  - (visibilidad) public, protected, private
  - (estatus) final, static, transient, volatile
- ❑ Modificadores de Métodos:
  - (visibilidad) public, protected, private
  - (estatus) final, static, abstract
  - (conurrencia) synchronized
  - ... strictfp (para RT)

- ❑ Modificadores de clases:
  - (visibilidad) public
    - (solo para clases internas) protected, private
  - (status) final, abstract, static
  - (...) strictfp.

*Modificadores class nombre*

*[extends nombre] [implements i1 [, i2 [, ...]]]*

*cuerpo*

## Genericidad y demás

### □ Genericidad (En Java 1.4):

- No es lo mismo que plantillas en C++
- Elimina la necesidad de moldeados (cast).
- Ejemplo: Utilización de `java.util.List` con Strings.
- A la antigua:  

```
List ids = new ArrayList();  
...  
String id = (String) ids.get(index);
```
- Con genéricos:  

```
List<String> ids = new ArrayList<String>();  
...  
String id = ids.get(index);
```

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

25

## Genericidad y demás

- Bucle **for** mejorado
- La clase `Iterator` permite recorrer un `Collection`; ahora puede ser substituido por la extensión de **for**.
- Antes:  

```
ArrayList<Integer> lista = new ArrayList<Integer>();  
for (Iterator i = lista.iterator(); i.hasNext();) {  
    Integer valor = (Integer) i.next();  
    ...  
}
```
- Ahora:  

```
ArrayList<Integer> lista = new ArrayList<Integer>();  
for (Integer valor : lista) {  
    ...  
}
```

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

26

## Paquetes

- ❑ Los paquetes definen espacios de nombres
  - para jerarquizar
  - evitar colisiones
- ❑ Los nombres están totalmente cualificados cara al exterior del paquete  
`nombre[. nombre[. . . . ] ]`
- ❑ Se declara al principio del archivo:  
`package mi codi go. grafi cos;`

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

27

## Paquetes

- ❑ Al importar un paquete se entra en su espacio de nombres:  
`i mport mi codi go. *;`
- ❑ `j ava. l ang. *` y el paquete “actual” están importados automáticamente.
- ❑ Cuando se programa debe reproducirse la estructura del paquete en los directorios.

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

28

## Excepciones

- ❑ Una excepción es un objeto de una subclase de `java.lang.Throwable`.
- ❑ Sirve para **señalizar** errores mediante el lanzamiento (`throw`) de excepciones (entre otras cosas).

```
public FileInputStream(String nombre)
    throws FileNotFoundException
{
    this.fh = system_open(name); // código nativo
    if (fh < 0) {
        throw new FileNotFoundException("No encuentro "
            + nombre);
    } else {
        // .. Hacer lo que sea necesario
    }
}
```

Hay que declararlas

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

29

## Excepciones

- ❑ Las excepciones hay que atraparlas para que sirva de algo.
  - Si no se atrapan se elevan (`raise`) al nivel superior.
- ❑ Tras atraparlas, podemos recuperar el error, o lanzarlas hacia arriba bajo otra forma más abstracta

```
try {
    FileInputStream in = new FileInputStream("archivo");
    in.read(buf, offset, size);
    // código a tutiplén
} catch (final FileNotFoundException ex) {
    ex.printStackTrace(); // y más cosas
} catch (final IOException ex) { ...
} finally {
    if (in != null) in.close();
}
```

Bloque a comprobar

Bloque a ejecutar

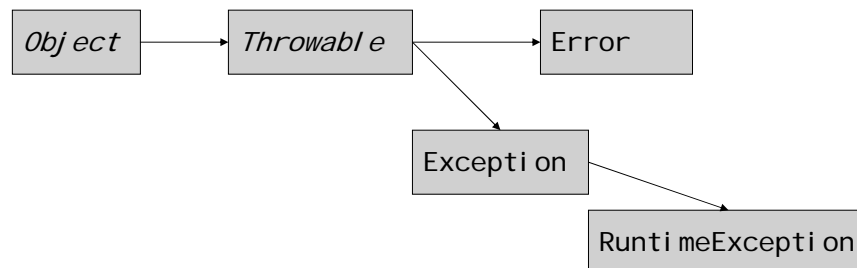
Siempre se ejecutará

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

30

## Excepciones



- ❑ Exception normalmente se lanza con `throw`.
  - Son **excepciones comprobadas** siempre hay que atraparlas o elevarlas explícitamente
- ❑ `RuntimeException` y `Error`
  - Son excepciones no comprobadas (unchecked)
  - No hay porqué atraparlas

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

31

## Seguridad

- ❑ Java implementa un esquema de seguridad basado en el control de acceso a los recursos (objetos, clases, archivos, conexiones)
- ❑ El control de seguridad corre a cargo de una clase gestor de seguridad (security manager)
- ❑ El control de acceso a los recursos se fija mediante archivos de política (policy)
  - Los applets tienen un gestor de seguridad por defecto y una política por defecto.

22/02/2005

Sistemas Distribuidos - ¿Qué es Java? (c) César Llamas 2003 (UVA)

32



## Enlaces útiles:

- ❑ El tutorial Java de Sun

<http://java.sun.com/docs/book.tutorial>

- ❑ Tutorial sobre fundamentos del lenguaje Java:

<http://developer.java.sun.com/developer/onlineTraining/Programming/BasicJava1/index.html>

<http://developer.java.sun.com/onlineTraining/Programming/BasicJava2/index.html>

- ❑ La especificación de Java

<http://java.sun.com/docs/books/jls/html/index.html>