

Java

Java es un lenguaje de programación expresamente diseñado para su uso en ambientes distribuidos, en específico en Internet. Java fue diseñado para poseer la "aparición y presentación" del lenguaje C++, pero es más fácil de usar que éste último, y refuerza el paradigma de Programación Orientada a Objetos.

Java puede ser usado para crear aplicaciones completas que pueden ejecutarse en una sola computadora o sistemas que pueden ejecutarse a través de muchos servidores y clientes en una red. También puede ser usado para crear módulos relativamente más pequeños (Applets) que pueden ser parte de una página en la Internet.

Ultimamente Java ha desarrollado tecnologías para aplicaciones en Web específicamente en Intranet tales como Servlets y JSP, entre otros.

Entre sus características más importantes están:

Orientado a Objetos.

Java es un lenguaje orientado a objetos. Los objetos definen en estructuras encapsuladas tanto sus datos (atributos) como los procedimientos y/o funciones (métodos) que manipulan esos datos los cuales se comunican entre sí para el funcionamiento de un programa y la definición del sistema. Java fue diseñado como un lenguaje orientado a objetos desde el principio; la tendencia del futuro, a la que Java se suma, apunta hacia la programación distribuida, especialmente en entornos cada vez más complejos y basados en red.

Distribuido.

Es decir, para ejecución desde una máquina externa a la que reside el programa. Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

Portable

La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas; es decir, el código fuente es compilado en lo que se denomina como bytecode, que puede ejecutarse en cualquier lugar que posea un intérprete de Java (también denominado Máquina Virtual de Java - Java Virtual Machine). Así, versiones específicas para diferentes tipos de computadora (PC, Mac, plataformas Unix/Linux, Solaris) ya no son necesarias.

Interpretado y Compilado a la vez

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina: los bytecodes, semejantes a las instrucciones de ensamblador (Assembler). Por otra parte, es interpretado, ya que los bytecodes se

pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).

Robusto

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de apuntadores), ya que se ha prescindido por completo los punteros, y la recolección de basura (garbage collector) elimina la necesidad de liberación explícita de memoria. Lo que significa que a diferencia de los programas escritos en C++, los objetos en Java no contienen referencias a datos externos a ellos mismos u otros objetos conocidos. Esto asegura que una instrucción no puede contener las direcciones de los datos en otra aplicación o en el sistema operativo mismo, lo que podría causar que la ejecución del programa (o del sistema mismo) se interrumpa. La máquina virtual de Java realiza una serie de chequeos en cada objeto para asegurar la integridad de sus datos.

Seguro (¿?)

Dada la naturaleza distribuida de Java, donde las applets se bajan desde cualquier punto de la red, la seguridad se impuso como una necesidad de vital importancia. A nadie le gustaría ejecutar en su computador programas con acceso total a su sistema, procedentes de fuentes desconocidas. Así que se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

Independiente de Arquitectura

Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows, pasando por Mac, Linux y estaciones de trabajo Sun, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variados, el compilador de Java genera bytecode: un formato intermedio indiferente a la arquitectura diseñado para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete Java.

Dinámico

El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

Java fue introducido por Sun Microsystems en 1995 e instantáneamente creó un nuevo sentido con respecto a las posibilidades de la Internet.

Aplicación de Java

Una aplicación en Java es un programa que se ejecuta en una computadora o sistema.

Primero pasa por un editor de texto, generando el código fuente de éste el cuál en Java se denomina **Clase**, después de compilarlo se genera el byte code (o *código objeto*) el cual pasa a través de la Máquina Virtual de Java (*Java VM*) que lo ejecuta.

Compilar en Java (utilizando el JDK)

```
javac NombreClase.java
```

Ejecución de una Aplicación

```
java NombreClase
```

La aplicación se procesa en la sesión y utiliza la salida estándar (o en caso diferente sí se especificó otra) para su realización, y el procesamiento de excepciones en la salida de error estándar.

A continuación se presenta el ejemplo de una aplicación en java, la clase lleva por nombre HolaMundo y lo que hace es la salida del mensaje "Hola Super Mundo" en la salida estándar del sistema.

//Aplicación Hola Mundo

```
public class HolaMundoAplicacion {  
  
    /**  
     * La Clase HolaMundoAplicacion implementa una Aplicación que  
     * despliega "Hola Super Duper Mundo" a la salida standard.  
     */  
  
    public static void main(String args[]) {  
  
        System.out.println("Hola Super Duper Mundo");  
    } // Fin del main  
} // Fin de la clase
```

La estructura de una aplicación en Java está determinada por 4 partes principales:

Comentario(s)

Los comentarios en cualquier lenguaje de programación sirven para especificar algo más en el programa que se quiera aclarar por lo general referente al programa o cualquier otra descripción que pueda ayudar a entenderlo mejor. Existen tres tipos de comentarios en Java, los de una línea, los de una a varias líneas y los comentarios javadoc (el cual veremos después más específicamente)

```
// Esto es un comentario de una línea
```

```
/* <-- inicio de comentario
```

Esto es el comentario de varias líneas, se inicia con un *asterisco* (*) y se termina el bloque de comentarios con un asterisco seguido de una diagonal (*), los pares de comentarios de abrir y cerrar deben corresponder uno a uno.

```
*/ <-- fin de comentario
```

Sentencia(s) import

Los lenguajes OO manejan Objetos, en Java las clases se agrupan en paquetes (packages) los cuáles pueden ser importados a través de una sentencia (import) y su dirección y/o ubicación de éste. De ésta manera se pueden utilizar los métodos de éstas clases.

import java.applet.Applet;

Nombre de la Clase

Un programa en java se compone por una o más clases las cuales pueden estar definidas (o predefinidas) en el programa. Siempre que se denomine una clase está debe tener un nombre, que define el programa, donde este debe ser igual al nombre del código fuente con la extensión ".java".

El nombre de la clase siempre va después de la especificación class; el nombre descriptivo de la clase tiene una Notación de la primera letra y de las palabras que conforman el nombre en mayúsculas y lo demás en minúsculas.

```
public class MiClase
```

Método main

Se menciona que las clases dan servicios, cuando se ejecuta una clase los servicios son administrados por métodos. Con java se procesa el método principal (main) en el cual se procesa en la sesión de la máquina.

Applet

Un applet es un programa que puede ser ejecutado en Internet. En términos generales es una aplicación programada en Java, que emplea la arquitectura cliente/servidor para realizar tareas a través de una GUI sin la necesidad (o en un caso específico) de una comunicación continua entre el servidor y los clientes. En adición a ser ejecutado en el cliente en vez de en el servidor, un applet de Java posee otras características diseñadas para hacerlo ejecutarse con rapidez.

A diferencia de una aplicación el applet puede ser interpretado por un navegador Web o por el appletviewer (visualizador de applets)

// Applet Hola Mundo

```
import java.applet.*;
import java.awt.*;
```

```
public class HolaMundoApplet extends Applet {
/**
 * La Clase HolaMundoApplet implementa un Applet que
 * despliega "Hola Super Duper Mundo" sobre este.
*/
    public void paint(Graphics g) {
        g.drawString("Hola Super Duper Mundo",50,50);
    } // Fin de paint
} // Fin de la clase
```

En los applet el contenido de una GUI (Graphic User Interface) se procesan en el método paint y su inicialización de componentes en el método init este sería su equivalente con el método main aquí se presenta un ejemplo de una clase llamada Applet Hola Mundo el cual muestra el mensaje "Hola Super Duper Mundo" el código fuente se interpreta a través de un archivo html que hace referencia al código objeto y el tamaño del applet.

El código fuente se llama **HolaMundoApplet.java**

```
<html>
<head>
<title>Applet Hola Mundo</title>
</head>
<body bgcolor="#FFFFFF">
<center>
<applet code="HolaMundoApplet.class"width="300"height="100">
</applet>
</center>
</body>
</html>
```

Swing

Swing es un programa que provee la habilidad de desarrollar componentes GUI (Graphic User Interface - Interfaces Gráficas de Usuario) tales como botones, listas de scroll, etc., que son independientes del sistema de interfaz gráfico del sistema (windowing system). Los componentes Swing son parte de grupo de desarrollo de interfaces gráficas de Java, llamada Java Foundation Classes (JFC).

Referencias:

<http://www.mhhe.com/wu2>

<http://java.sun.com/>

<http://www.whatis.com/>

<http://www.iec.csic.es/criptonomicon/java/quesjava.html>