

Unidad 2

Características del lenguaje Java

Programación
1º D.A.M.

Contenido

1. Origen
2. Ventajas
3. Tipos de aplicaciones
4. Etapas del desarrollo
5. Software Java
6. Editores y entornos



1. Origen

■ Años 80 : lenguajes estructurados

■ Problema

- Complejidad aumenta mucho con tamaño de problema

■ Solución

- Adaptación de los lenguajes existentes a la POO
 - C → C++
 - Pascal → Turbo Pascal → Delphi
 - Basic → QuickBasic → Visual Basic

1. Origen



■ Líder inicial : C++

■ Ventajas

- Soporte de POO (incluida herencia múltiple)
- Potentes bibliotecas para desarrolladores
 - MFC
- Permite escribir en C
- Muy veloz

■ Inconvenientes

- Lenguaje compilado
 - Produce ejecutable válido sólo para una plataforma
- Híbrido
- Inseguro
- No apropiado para la web

1. Origen

■ Sun Microsystems

■ 1991

- Oak

■ 1995

- Java
 - Versión mejorada de Oak
 - Sintaxis basada en C++
 - Funcionamiento muy distinto a C++



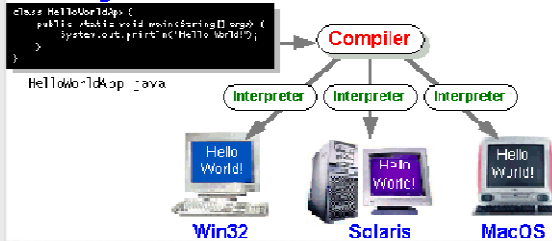
2. Ventajas



- Sintaxis similar a C y C++
- Más seguro
 - No hay punteros
 - La máquina virtual decide qué ejecutar
- Totalmente orientado a objetos
- Muy preparado para aplicaciones TCP/IP
- Implementa excepciones de forma nativa
- Interpretado
- Permite multihilos (multithread)
- Admite firmas digitales
- Tipos de datos más estrictos
- Independiente de la plataforma

2. Ventajas

Java Program



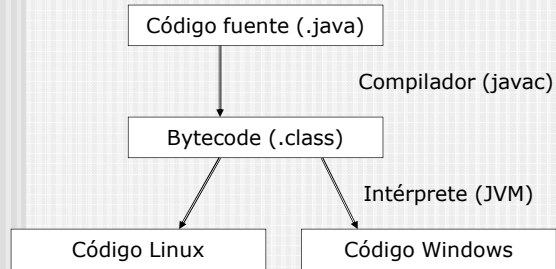
3. Tipos de aplicaciones

- Aplicaciones de consola
 - Mostradas en la consola del sistema
- Aplicaciones gráficas
 - Usan objetos del sistema de gráficos
- Applets
 - Embebidas en una página web
 - Ejecutadas en el cliente
- Servlets
 - Embebidas en una página web
 - Ejecutadas en el servidor

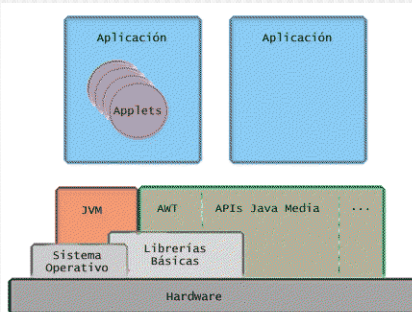
4. Etapas del desarrollo

- Creación del código fuente
 - Herramienta → Editor de textos
 - Origen → -
 - Resultado → Fichero fuente
 - Fichero.java
- Compilación del código fuente
 - Herramienta → Compilador java
 - Javac
 - Origen → Fichero fuente
 - Fichero.java
 - Resultado → Bytecode
 - Fichero.class
 - Código semicompilado
- Interpretación del código fuente
 - Herramienta → Intérprete java
 - Máquina virtual
 - Origen → JVM
 - Fichero.class
 - Resultado
 - Ejecución independiente de plataforma

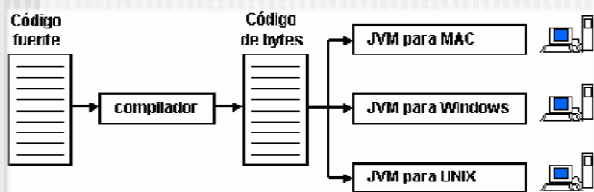
4. Etapas del desarrollo



4. Etapas del desarrollo



4. Etapas del desarrollo



5. Software Java

- Software de ejecución
 - JRE
- Software de desarrollo
 - JDK
 - JSE
 - JEE
 - JME

5. Software Java

- JRE
 - *Java Runtime Environment*
 - Entorno de ejecución Java
 - Contiene
 - Máquina Virtual Java (JVM)
 - Clases necesarias para ejecutar programas

5. Software Java

- JDK
 - *Java Development Kit*
 - Kit de desarrollo Java
 - SDK : *Software Development Kit*
 - Kit de desarrollo de software
 - Contiene
 - JRE
 - Clases necesarias para desarrollar Java
 - Visores de código, depuradores y otras utilidades
 - Características
 - Gratuito (www.oracle.com/technetwork/java)
 - Permite compilar y ejecutar programas Java

5. Software Java

■ JDK

■ Versiones

- JSE (J2SE) *Java2 Standard Edition*
 - Plataforma Java estándar
- JEE (J2EE) *Java2 Enterprise Edition*
 - Versión empresarial
- JME (J2ME) *Java2 Micro Edition*
 - Versión para dispositivos portátiles

6. Editores y entornos

- Net Beans (Free Code)
- Eclipse (Free Code)
- Geany
- Sun ONE Studio
- Borland Jbuilder
- Oracle Jdeveloper
- Visual Age
- Visual Café
- Visual J#.Net y Visual J++
- IntelliJ Idea
- Kawa Pro
- JCreator

Unidad 2

Características del lenguaje Java

Programación
1º D.A.M.