

Introducción a Java

Java: Motivaciones de su origen

- “Deja” atrás características “problemáticas”:
 - Punteros
 - Asignación de memoria (malloc)
 - Herencia múltiple (se verá en otra sesión)
- Ofrecer lenguaje independiente de:
 - Tipo de computador
 - Sistema operativo
 - Sistema de ventanas (win32, Motif, etc.)
 - Obs: Cuando Java aparece (1995) no existía Qt (herramienta para desarrollar software en C++ para múltiples plataformas)

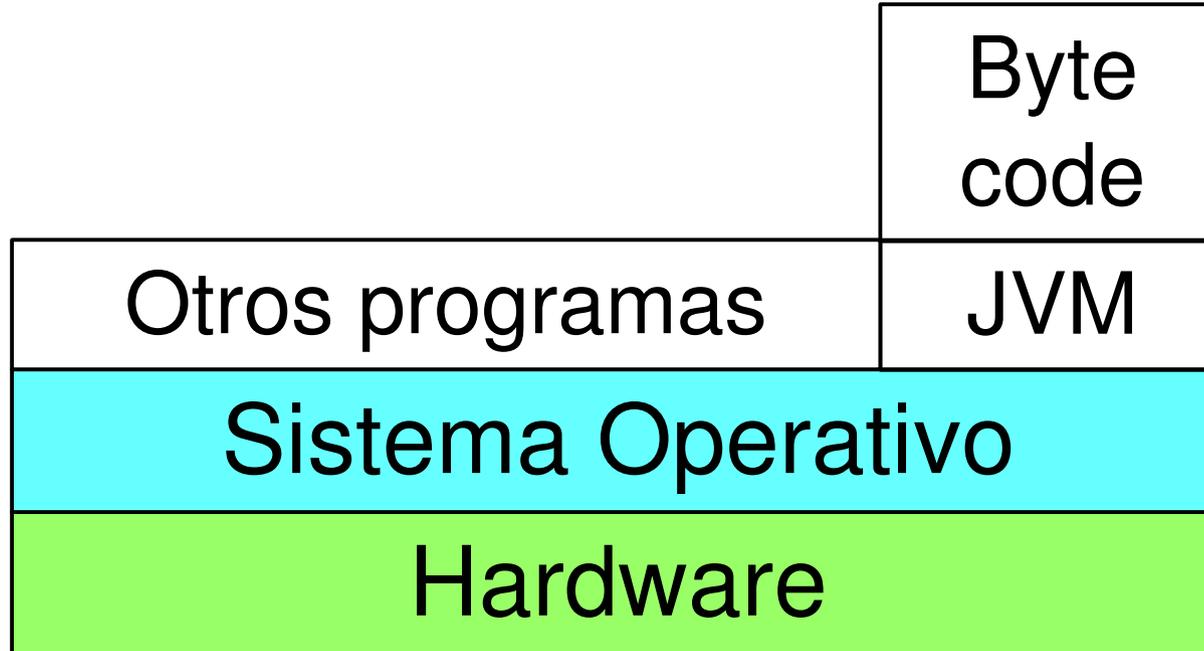
Elude Características “Problemáticas”

- Los punteros generan dificultades para muchos. No lo creo así para quienes estudian la estructura de un computador (caso ELO/TEL). Puede ser el caso de alumnos de segundo año.
- Java tiene 8 tipos de datos básicos (int, float, char, etc), todos los otros datos son objetos y son almacenados en memoria dinámica (heap: zona de memoria grande manejada por el Sistema Operativo para usos dinámicos por parte de las aplicaciones).
- Java no pide al programador liberar (free) la memoria solicitada al ubicar objetos en el heap (la creación de objetos es similar a usar malloc en C). El lenguaje se preocupa de reutilizar la memoria liberada por objetos fuera de uso (aquellos sin referencias para ser accedidos desde el programa).
- Java evita herencia múltiple, se verá en varias clases más.

Independiente del Computador y Sistema Operativo

- Esto se logra por el uso de una **Máquina Virtual Java (Java Virtual Machine)**.
- Una máquina virtual es una abstracción de una máquina real. La máquina virtual es generada por software.
- ¿Han usado programas emuladores de consolas de juegos?
- ¿Han usado programas emuladores de PC dentro de un PC? Así podemos tener varios Sistemas operativos corriendo concurrentemente en la misma máquina. Ej: Vmware, VirtualBox.
- Este concepto también es aplicable a sistemas operativos donde es posible crear la apariencia de tener varias máquinas independientes (jaulas o jails)

Java Virtual Machine



- Para cada combinación hardware+SO se desarrolla una máquina virtual Java (es un programa más)
- El programa compilado Java (byte code) corre “igual” en todas las máquinas virtuales
- Ver

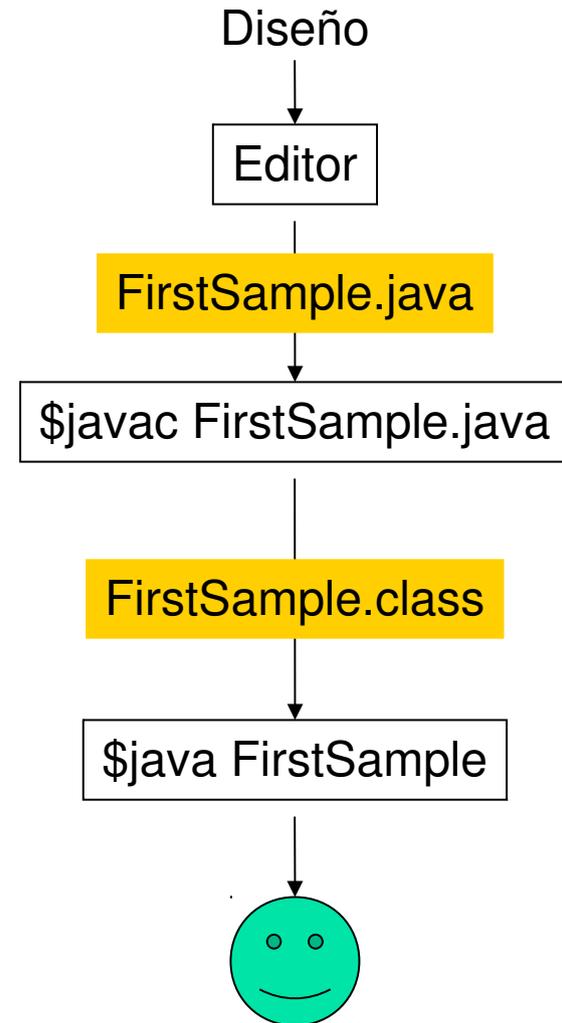
<http://www.oracle.com/technetwork/java/index.html>

Compilación



Trabajando con Java

- Creación programa: Con editor crear programa de extensión java (FirstSample.java)
- Hacer uso de documentación en manuales.elo.utfsm.cl
- **Compilación:** vía el comando en línea
\$ javac FirstSample.java
La salida serán archivos .class, es la versión del programa en código byte.
- **Ejecución:**
\$java FirstSample
Notar que java es el programa que corremos para crea la **máquina virtual** donde el “byte code” es ejecutado, equivale a una **interpretación** en la máquina real.
- Hay ambientes de desarrollo amigables para hacer estas tareas. IDE (Integrated Development Environment)



Editores de texto

- Recomiendo aprender a digitar bien.
- Usar ambientes integrados de Desarrollo (IDE) como:
 - **Jgraps**
 - **Eclipse**
 - Netbean (de la página de Oracle)
- **Emacs** (win o Linux) u otro. Para mi gusto un buen editor debería ayudar a indentar su programa.
- Hay otros, ver conveniencia.
- No usar notepad o similar. 
- Ver editores en página del ramo

Sistema de Desarrollo (JDK)

- Lo puede bajar de Oracle:
 - <http://www.oracle.com/technetwork/java/index.html>
- Se puede instalar del repositorio de Linux (apt-get)
- Tecnologías:
 - Java EE (Enterprise Edition),
 - **Java SE (Standard Edition, JDK)**, <= Esta asignatura
 - Java Embedded
 - Java ME (Micro-Edition)
 - Otras ...