



UNIVERSIDAD DE COSTA RICA

Sede de Occidente – Recinto de San Ramón

Curso: IF-1300 Introducción a la Computación e Informática

Profesor: Mci. Denis González Herrera

Introducción al lenguaje de programación Java (Codificación, Compilación, Ejecución, Depuración....Codificación)

PRIMERA PARTE [INSTALACIÓN DE PROGRAMAS]

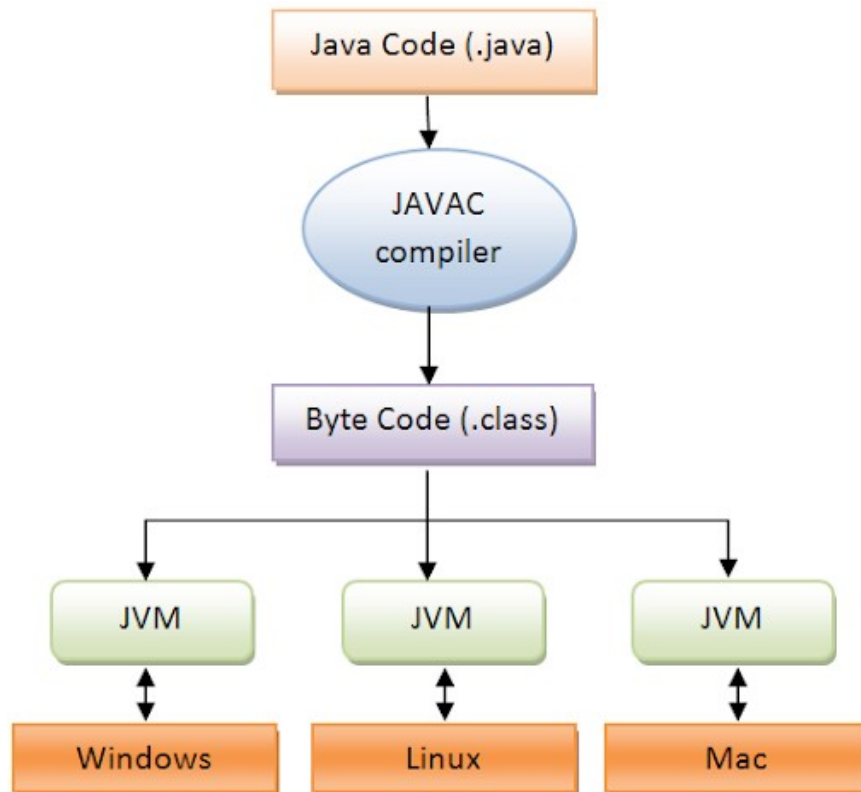
NOTA: La siguiente actividad requiere que las personas hayan realizado la lectura de al menos los apartados: “**antes de empezar**” y del **capítulo 1 y capítulo 2** introducción del libro de Deitel” **Libro1**.

TEMA A DESARROLLAR: Cómo compilar en java el código fuente (*.java) y cómo ejecutar mi programa compilado (.class) ?

1. **Paso1** - Necesitamos un archivo de texto escrito en lenguaje java con la sintaxis perfecta y con la extensión en el archivo: ***.java (código fuente)** [para una mejor comprensión ver los ejemplos de los capítulos 1 y 2 del **libro1**]
2. **Paso2** – Necesitamos tener **instalado** alguna versión del **JDK** (kit de desarrollo de java) que también instala el programa o el **compilador** de java (Este compilador tiene el nombre **javac**)
3. **Paso3** - Necesitamos tener **instalado** alguna versión del programa o **interprete** de java (**MVJ**) para **ejecutar** nuestros programas ya compilados (este programa tiene el nombre de java)
4. **Videos recomendado:** Los siguientes videos muestran la instalación, configuración y ejecución del programa escrito en java (pueden buscar otros videos de ejemplos si lo consideran necesario).
 1. Video instalación y configuración JDK recomendado: <https://www.youtube.com/watch?v=J9X4iz1ugUM>
 2. Video para compilar y ejecutar archivos en java: <https://www.youtube.com/watch?v=-F0bIeVhRH8>



IMPORTANTE: El siguiente diagrama describe nuevamente el proceso de escribir, compilar y ejecutar un programa hecho en el lenguaje java.





SEGUNDA PARTE [programación en java]

A continuación se recomienda realizar la siguiente actividad para poner en práctica todas las teorías y los conceptos de programación y del lenguaje de programación java.

Importante: Para realizar la siguiente práctica cada persona debe tener instalado el compilador y el interprete de java. Además, debe haber escrito y ejecutado al menos un ejemplo en java utilizando un editor de texto simple. También, se debe utilizar el CMD o el interprete de comando de Windows, o la terminal en MAC o Linux según correspondo a cada caso para compilar y ejecutar los programas. NO se debe por ninguna circunstancia utilizar programas profesionales o educativos para compilar y ejecutar los programas como por ejemplo: Netbeans, Eclipse entre otros.

Ejercicios de práctica:

- Se les invita a resolver en el lenguaje java todos los ejemplo de la práctica: (Practica 1 Ejercicios propuestos en PSEINT.pdf de la unidad 4). Pueden utilizar los diagramas de flujos o el pseudocódigo de las soluciones como guía. Se recomienda como **mínimo** resolver los primeros ejercicios **secuenciales** y luego pueden trabajar en los siguientes ejercicios: selectivos y repetitivos según su tiempo y disposición para que trabajen a su propio ritmo. NOTA: antes de continuar por favor leer la **tercera parte** de este documento.

Nota: a continuación se muestra un resumen de los conceptos de la **presentación fundamentos de java** que deben ser dominados de forma **teórica** (concepto y definición) y de forma **práctica** (aplicación, uso y comprensión).

- Variable?
 - Qué es una variable?
 - Declaración o creación?
 - Inicialización?
 - Asignación?
 - Nombres adecuados para variables?
 - Ámbito de las variables?
 - Ámbito Global de una variable?
 - Ámbito Local de una variable?
 - CONSTANTE?
 - Literal o literales?
 - Tipos de variables?
 - Tipo primitivos de variables?
 - Tipo de datos?
 - Convertir valores (casting)
- Operadores aritméticos?
 - Prioridad entre operadores aritméticos?
 - Operadores Unarios?
 - Operadores de asignación directa?



- Operadores relacionales?
 - Prioridad entre operadores relacionales
- Operadores lógicos
- Concatenación
 - Secuencias de escape o literales de carácter
 - Uso de caracteres con el sistema Unicode
 - Uso de caracteres con el sistema Ascii
- Estructuras de control en java y bloques de control
 - Estructuras secuenciales
 - Estructuras selectivas
 - Estructuras repetitivas
- Librerías y paquetes para las Entradas y salidas de datos (JOptionPane o scanner, entre otros)



TERCERA PARTE [consejos y buenas prácticas]

A continuación se le brinda algunas sugerencias y recursos de mucha utilidad para facilitarles y ayudarle en el proceso de aprendizaje de la programación de computadoras.

CONSEJOS Y BUENAS PRÁCTICAS EN PROGRAMAR

1. **Resuelva** primero las partes del enunciado que considera más simple para usted, o resuelva primero las que comprenda perfectamente. Una vez resueltas las partes simples continúe escalando a las más complejas o difíciles.
2. **Trate** de programar pocas líneas de código una a la vez, de manera que las partes sean pequeñas y de un tamaño inteligible (**inteligible**: significa que toda la información necesaria para terminar una tarea o instrucción, puede estar en su mente sin requerir de estar viendo o consultando otros datos o ayudas adicionales).
3. **Recuerde**, Escribir el código fuente pensando en que otras personas van a necesitar leerlo y comprenderlo. Y nunca debe programar únicamente para mí mismo.
4. **Prueba y error, corrección de errores o Depurar**: siempre que tenga un fragmento de código listo (compilable o ejecutable) realice las pruebas en busca de errores en orden y estructurados desde el inicio al final del código escrito.
5. **Escribir el código**, no copiar código (ctrl+c, ctrl+v). Copiar es aumentar exponencialmente los errores y trabajar en corregir errores es completamente evitable. Copiar es perder tiempo, esfuerzo y dinero. Al copiar no se está aprendiendo el nuevo lenguaje, se necesita de escribir palabra por palabra, instrucción por instrucción para poder memorizar, practicar y mejorar en sus habilidades de programación. Una de las mejores formas de aprender a programar es programando.
6. **Abrir y cerrar** siempre un bloque de código debe ser lo primero, todo tiene un inicio y un fin. Se recomienda Escribir siempre primero el inicio y fin de un bloque, de una sentencia o de una instrucción, para no generar errores muy difíciles y costosos de corregir que son completamente evitables.
7. **Utilizar nombres significativos** para todas las palabras, variables, CONSTANTES, Clases, métodos o funciones. De esta manera cualquier lector de su código puede comprender fácilmente su trabajo o usted mismo puede encontrar errores o mejorar su algoritmo.
8. **Cumplir** lo que pida el **enunciado** o el requerimiento de forma **explícita** primero y luego programar lo que diga de forma implícita. O consultar dudas que no se comprendan sobre el enunciado del problema.
9. **Realice documentación** interna de al menos los procedimientos y cálculos más importantes o complicados. Esta información será valiosa para usted y otras personas.



RECURSOS OFICIALES DE CONSULTA SOBRE EL LENGUAJE JAVA

El lenguaje java pertenece a la empresa ORACLE quienes se encargan de actualizar el lenguaje y crear recursos para enseñar las formas correctas de hacer uso de este lenguaje de programación. A continuación se le brinda algunos sitios web con información fundamental e introductoria para el aprendizaje de la sintaxis del lenguaje de programación, **estos recursos oficiales son los primeros que deben ser consultados ante cualquier error o duda sobre el lenguaje.**

1. Oracle:

- Tutoriales de java (fundamentos de programación)
<https://docs.oracle.com/javase/tutorial/java/index.html>
- Documentación completa sobre java:
<https://docs.oracle.com/javase/8/docs/index.html>
- Ayuda técnica sobre java
<https://docs.oracle.com/javase/10/docs/api/index.html?overview-summary.html>

2. Libros de java

1. Como programar a Java, 10ª Edición - Paul Deitel