

Java Virtual Machine e IDE

Conceptos sobre Java

Compilar

- Pasar de un lenguaje de programación a otro.
- En particular, cuando nosotros utilizamos la palabra compilar estamos queriendo decir: traducir el código escrito en “Lenguaje entendible por humanos” (por ejemplo Java, C, Pascal, Fortran), a un código en “Lenguaje Máquina” (que entienden las máquinas).
- Se hace esto porque a los humanos nos resultaría casi imposible trabajar directamente con el lenguaje de las computadoras. Es por eso que usamos un lenguaje más entendible para nosotros (en nuestro caso Java) y luego empleamos un traductor (compilador) que lo pasa a lenguaje de máquina.
- La creación de programas en muchos lenguajes se basa en el proceso: escribir código fuente --> compilar --> obtener programa ejecutable. Si el programa no compila (hay error de sintaxis), no se crea el ejecutable.

Es decir:

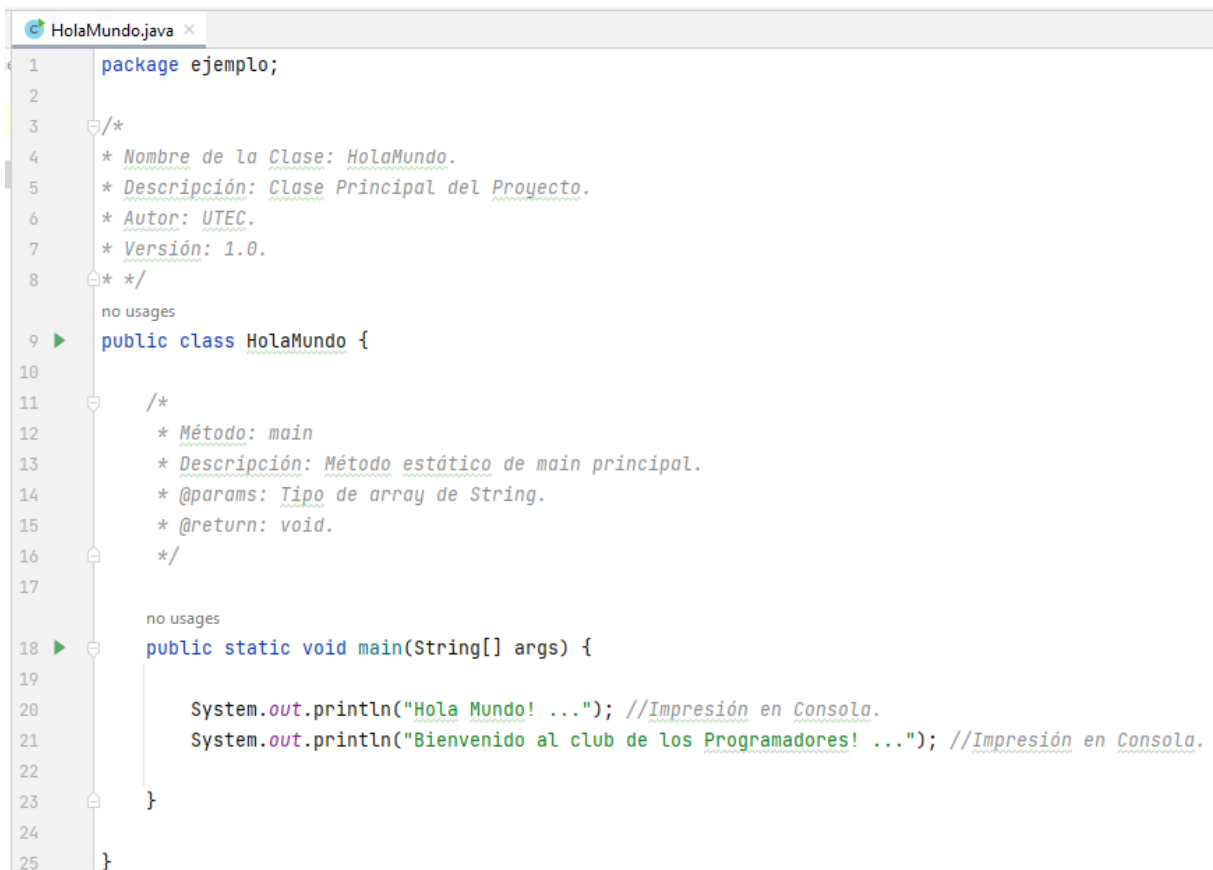
La compilación consiste en tomar los archivo de código fuente que conforman un programa y, línea por línea, traducir cada instrucción de alto nivel (en java, c#, Python, etc.) por varias instrucciones en código máquina que realicen lo que la instrucción de alto nivel expresa.

- Si se repite esa traducción para todas las líneas del código fuente, obtenemos un conjunto de instrucciones máquina.
- Si nuestro programa no tiene errores de sintaxis, se graban esas instrucciones máquina en un archivo y obtenemos el archivo ejecutable (capaz de ser ejecutado en un sistema operativo).

Compilador

Habitualmente, cuando programamos y codificamos en cierto lenguaje, escribimos este en uno (o varios) archivos con texto. Los mismos tienen una extensión específica según el lenguaje (por ejemplo, en java, los archivos son de la forma “.java”). A estos llamaremos código fuente.

Por ejemplo, en java, podemos tener algo como lo siguiente:



```
1 package ejemplo;
2
3 /*
4  * Nombre de la Clase: HolaMundo.
5  * Descripción: Clase Principal del Proyecto.
6  * Autor: UTEC.
7  * Versión: 1.0.
8  */
9 public class HolaMundo {
10
11     /*
12      * Método: main
13      * Descripción: Método estático de main principal.
14      * @params: Tipo de array de String.
15      * @return: void.
16      */
17
18     public static void main(String[] args) {
19
20         System.out.println("Hola Mundo! ..."); //Impresión en Consola.
21         System.out.println("Bienvenido al club de los Programadores! ..."); //Impresión en Consola.
22     }
23 }
24
25 }
```

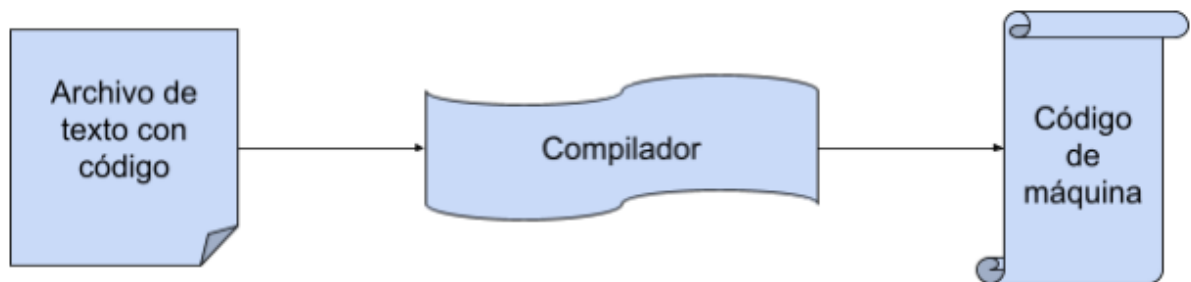
Tenemos el archivo HolaMundo.java El mismo contiene el texto escrito por el programador. El procesador de la computadora, no entiende un archivo con texto. Típicamente, se maneja con una serie de operaciones que tiene el procesador del computador, bastante limitado, utilizando código binario.

El código binario, es un número representado con ceros y unos: 00110101101

Este código, si lo entiende el procesador de la computadora.

Entonces... ¿Cómo entiende la computadora, el código que escribimos en un archivo que contiene texto?

Existe un componente de software, llamado Compilador. El mismo se encarga de recibir como entrada, nuestros archivos de texto y transformarlos a este lenguaje que si entiende el computador.



Llamamos proceso de Compilación, al proceso de traducción de un programa en un lenguaje dado, al lenguaje de la máquina.

En este proceso se valida también la sintaxis de nuestro programa para informar al programador ante errores en sus archivos. Además, en algunos casos el compilador puede validar la semántica del programa.

<https://es.wikipedia.org/wiki/Compilador>

Java Virtual Machine (JVM)

Los ejecutables que se generan en código de máquina, no siempre son iguales para todos los sistemas operativos (Windows, Macintosh, Linux, etc).

Entonces, el proceso de compilación en el lenguaje java, no es estrictamente el del diagrama anterior, ya que se introduce un nuevo concepto: Máquina Virtual

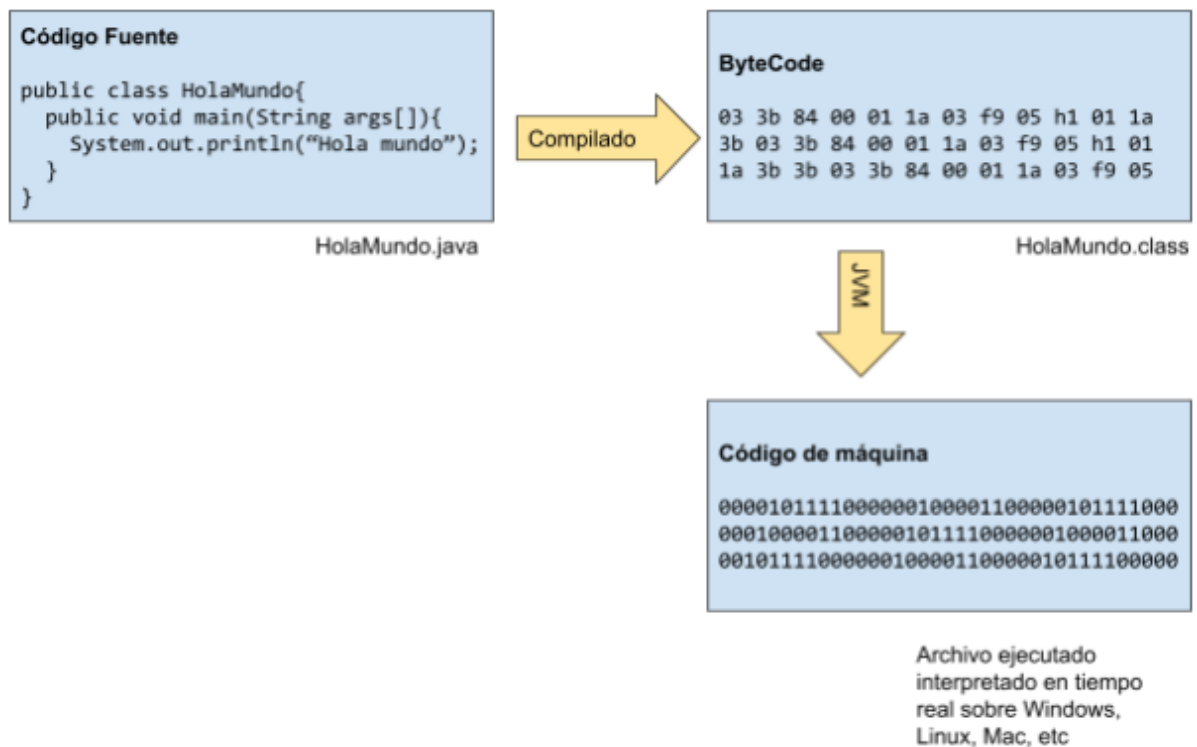
Máquina Virtual es un software que simula a una computadora y puede ejecutar programas como si fuese una computadora real.

Java simula una “máquina virtual” y es allí donde se termina realizando la compilación del programa.

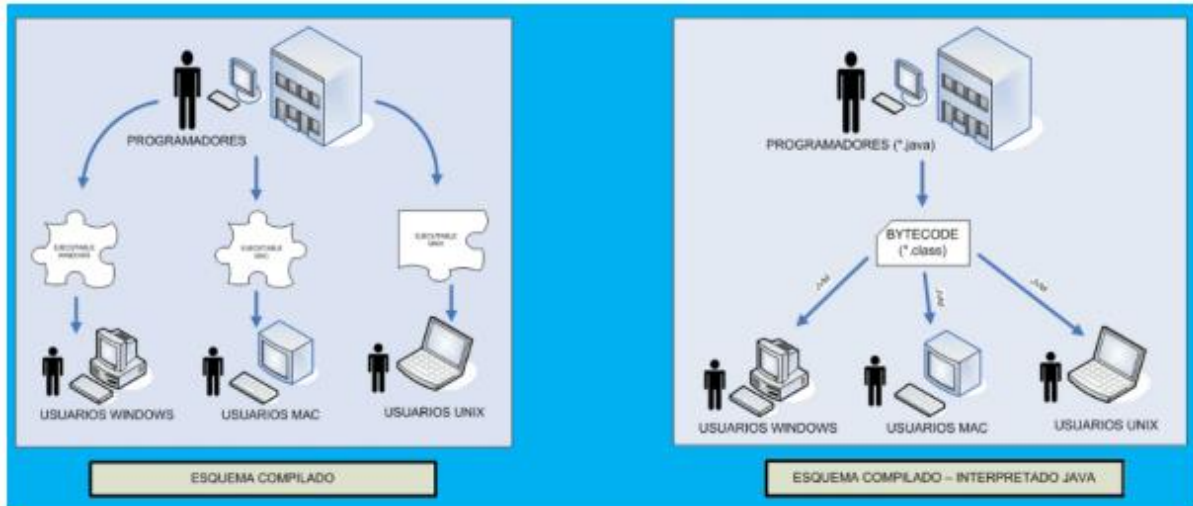
¿Cuál es el proceso de compilación y ejecución en Java? El compilador Java traduce nuestro código fuente (en Java) a un lenguaje intermedio llamado “byte code”

Los archivos en byte code son pasados a la máquina virtual Java. Este código, es interpretado y entendido por la Máquina Virtual, y se encarga de convertirlo en código de máquina y ejecutarlo.

A esta máquina virtual es lo que llamaremos JAVA VIRTUAL MACHINE o como dice su sigla, JVM. De esta forma los programas Java son independientes del hardware y del sistema operativo en se ejecuta.



Esto nos permite, independizarnos de la plataforma en la que estemos programando, ya que es la JVM la que se encarga de compilar e interpretar nuestro código fuente. Lo único que debemos hacer es instalar la JVM en nuestra computadora, lo que nos brindará las herramientas necesarias para correr nuestros programas. Como desventaja, podemos encontrar que, si deseamos correr un programa Java en nuestra máquina debemos tener la Máquina Virtual Instalada.



Java Runtime Environment

Como la propia palabra lo indica (en inglés), la JRE proporciona el entorno de ejecución de los programas escritos en Java. Entre las cosas que nos brinda la JRE, es la JVM para poder correr nuestros programas.

https://es.wikipedia.org/wiki/Java_Runtime_Environment

Java Development Kit

Es un conjunto de herramientas básicas brindadas por Java para facilitarnos el desarrollo. Contiene librerías con elementos ya implementados que podemos utilizar.

https://en.wikipedia.org/wiki/Java_Development_Kit

IDE (Integrated Development Environment)

El IDE es una aplicación que nos brinda un entorno visual atractivo para desarrollar y escribir nuestro código. Es una herramienta que nos ayuda a organizar los archivos escritos y sus dependencias (cuando queremos utilizar funcionalidades implementadas en un archivo desde otro). Particularmente, utilizaremos como IDE: IntelliJ IDEA. Nos brinda editores de

texto, donde se colorean las palabras y le da un cierto formato a los códigos que escribamos, para que quede mejor organizado.

Por otro lado, también nos brinda herramientas para depurar nuestros programas, y conocer los valores de las variables en cierto momento para poder encontrar potenciales errores que contengamos en nuestro código. Además, nos valida la sintaxis del lenguaje que estemos utilizando (Java). Si bien un IDE nos ayuda a desarrollar en diferentes lenguajes, podríamos hacerlo perfectamente en un editor de texto (bloc de notas → HolaMundo.java) y realizar por línea de comandos la compilación (comando `javac HolaMundo.java`) y ejecución (`java HolaMundo.class`)

https://es.wikipedia.org/wiki/Ambiente_de_desarrollo_integrado

IDE (Integrated Development Environment)

El IDE es una aplicación que nos brinda un entorno visual atractivo para desarrollar y escribir nuestro código. Es una herramienta que nos ayuda a organizar los archivos escritos y sus dependencias (cuando queremos utilizar funcionalidades implementadas en un archivo desde otro). Particularmente, utilizaremos como IDE: IntelliJ. Nos brinda editores de texto, donde se colorean las palabras y le da un cierto formato a los códigos que escribamos, para que quede mejor organizado.

Por otro lado, también nos brinda herramientas para depurar nuestros programas, y conocer los valores de las variables en cierto momento para poder encontrar potenciales errores que contengamos en nuestro código. Además, nos valida la sintaxis del lenguaje que estemos utilizando (Java).

Si bien un IDE nos ayuda a desarrollar en diferentes lenguajes, podríamos hacerlo perfectamente en un editor de texto (bloc de notas → HolaMundo.java) y realizar por línea de comandos la compilación (comando `javac HolaMundo.java`) y ejecución (`java HolaMundo.class`)

https://es.wikipedia.org/wiki/Ambiente_de_desarrollo_integrado