

Introducción a Java

Pedro O. Pérez M., PhD.

Desarrollo de videojuegos con Java
Tecnológico de Monterrey

pperezm@tec.mx

10-2023

Introducción a Java

- Introducción

- Terminología de Java

- Historia de Java

- Puntos importantes de Java

- Ventajas de Java

- Desventajas de Java

C++ vs. Python vs. Java

- C++ vs. Java

- Java vs. Python

¿Qué es JDK (java Development Kit)?

- ¿Qué es JDK (java Development Kit)?

Empezando a trabajar con Java

Instalando Java (JVM e IDE)

Convenciones de código en Java

Comentarios

Nuestro primer programa en Java

Examen Rápido 1

Introducción

- ▶ Java fue desarrollado por James Gosling en Sun Microsystems Inc. en el año 1995 y posteriormente adquirido por Oracle Corporation.
- ▶ Es un lenguaje de programación orientado a objetos diseñado para tener la menor cantidad de dependencias posibles.
- ▶ Es un lenguaje de programación de propósito general creado para que los desarrolladores lo escriban una vez y se ejecute en cualquier lugar
- ▶ Las aplicaciones Java se compilan en código de bytes (byte code) que puede ejecutarse en cualquier Máquina Virtual Java (JVM).
- ▶ La sintaxis de Java es similar a la de C/C++.

Antes de continuar, aclaremos algunos términos que estaremos usando:

- ▶ **Kit de desarrollo de Java (Java Development Kit, JDK):** Es un completo kit de desarrollo de Java que incluye todo: Compilador, Java Runtime Environment (JRE), Depurador (Debugger), Generador de Documentación (javadoc), etc.
- ▶ **Entorno de ejecución de Java (Java Runtime Environment, JRE):** Su función es permitir que el programa de Java se ejecute, más no para compilar. El JRE incluye un navegador, una JVM, soporte para subprogramas y complementos.

- ▶ **Bytecode:** El compilador, `javac`, traduce el código fuente en un código intermedio conocido como **bytecode**. El compilador lo guarde como archivo `.class`. Ese código será luego interpretado por la Máquina Virtual de Java (Java Virtual Machine, JVM).
- ▶ **Máquina virtual Java (JVM):** Su función es ejecutar el código, en “bytecode”, producido por el compilador. Cada sistema operativo tiene su propia JVM diferente, pero el resultado que producen después de la ejecución del código es el mismo en todos los sistemas operativos. Por eso se conoce a Java como un lenguaje independiente de la plataforma.
- ▶ **Recolector de Basura (Garbage Collector):** En Java, los programadores no pueden eliminar los objetos. Para borrar o recuperar esa memoria JVM usa el Recolector de Basura. El Recolector de Basura puede recuperar los objetos a los que ya no se hace referencia. De esta forma, Java facilita la vida de un programador al manejar la administración de la memoria.

Historia de Java

- ▶ Fue creado en 1991 por James Gosling, Mike Sheridan y Patrick Naughton, un equipo de ingenieros de Sun conocido como el equipo Green.
- ▶ El compilador Java1.0 fue reescrito en Java por Arthur Van Hoff para cumplir estrictamente con sus especificaciones.
- ▶ En 1997, Sun Microsystems se acercó al organismo de estándares ISO y formalizó el estándar de Java.
- ▶ El 13 de noviembre de 2006, Sun lanzó gran parte de su Máquina Virtual Java como software gratuito de código abierto.
- ▶ El 8 de mayo de 2007, Sun finalizó el proceso y puso a disposición todo el código central de su JVM bajo términos de distribución de código abierto.
- ▶ Actualmente, Java se utiliza en dispositivos móviles, programación de Internet, juegos, comercio electrónico, etc.
- ▶ Java es el nombre de una isla de Indonesia donde se produjo el primer café (llamado café Java). Y este nombre lo eligió James Gosling mientras tomaba un café cerca de su oficina.

Puntos importantes de Java

- ▶ Java es independiente de la plataforma, lo que significa que el código escrito en Java puede ejecutarse en cualquier plataforma que tenga una máquina virtual Java (JVM) instalada.
- ▶ Java es conocido por su filosofía de “escribir una vez, ejecutar en cualquier lugar”, lo que lo convierte en una opción popular para el desarrollo multiplataforma.
- ▶ El código Java se compila en bytecode, que luego puede ser ejecutado por cualquier JVM.
- ▶ Java proporciona gestión automática de la memoria mediante la recolección de basura, lo que facilita la escritura y el mantenimiento del código.
- ▶ Java tiene una amplia biblioteca estándar que proporciona una amplia gama de herramientas para tareas de programación comunes.

- ▶ Java se utiliza ampliamente en aplicaciones empresariales, desarrollo Web y desarrollo de aplicaciones para Android.
- ▶ Java es un lenguaje fuertemente tipado, lo que significa que cada variable y expresión tiene un tipo específico que debe declararse antes de su uso.
- ▶ Java tiene un sólido mecanismo de manejo de excepciones que facilita el manejo de errores y comportamientos inesperados en el código.
- ▶ Java admite hilos, lo que permite escribir programas que pueden realizar múltiples tareas simultáneamente.

Ventajas de Java

- ▶ **Independiente de la plataforma:** El código Java se puede ejecutar en cualquier plataforma que tenga instalada una máquina virtual Java (JVM), lo que significa que las aplicaciones se pueden escribir una vez y ejecutar en cualquier dispositivo.
- ▶ **Orientado a objetos:** Java es un lenguaje de programación orientado a objetos, lo que significa que sigue los principios de encapsulación, herencia y polimorfismo.
- ▶ **Seguridad:** Java tiene características de seguridad integradas que lo convierten en una plataforma segura para desarrollar aplicaciones, como la administración automática de memoria y la verificación de tipos.
- ▶ **Gran comunidad:** Java tiene una comunidad grande y activa de desarrolladores, lo que significa que hay mucho soporte disponible para aprender y usar el lenguaje.
- ▶ **Aplicaciones de nivel empresarial:** Java se utiliza ampliamente para desarrollar aplicaciones de nivel empresarial, como aplicaciones web, sistemas de comercio electrónico y sistemas de bases de datos.

Desventajas de Java

- ▶ **Rendimiento:** Java puede ser más lento en comparación con otros lenguajes de programación, como C++, debido al uso de una máquina virtual y a la gestión automática de la memoria.
- ▶ **Gestión de la memoria:** La gestión automática de la memoria de Java puede provocar un rendimiento más lento y un mayor uso de la memoria, lo que puede ser un inconveniente para algunas aplicaciones.

C++ vs. Java

Concepto	C++	Java
Gestión de la memoria	Uso de punteros, estructuras y/o unión.	Sin uso de punteros. Admite referencias, hilos e interfaces.
Bibliotecas	Comparativamente disponible con funcionalidades de bajo nivel.	Amplia gama de clases para diversos servicios de alto nivel.
Herencia múltiple	Proporciona herencia única y múltiple.	Las herencias múltiples se realizan parcialmente a través de interfaces.
Sobrecarga de operadores	Soporta la sobrecarga de operadores	No es compatible con esta función.

Concepto	C++	Java
Manejo del programa	Las funciones y variables pueden residir fuera de las clases.	Las funciones y variables residen sólo en clases, se utilizan paquetes.
Portabilidad	Dependiente de la plataforma, se debe volver a compilar para una plataforma diferente.	El código de bytes generado independiente de la plataforma funciona en todos los sistemas operativos.
Soporte de hilos	No hay soporte integrado para hilos, depende de las bibliotecas.	Tiene soporte para hilos incorporado.

Java vs. Python

Concepto	Java	Python
Proceso de compilación	Java es un lenguaje tanto compilado como interpretado, que primero se compila y luego se interpreta en un código de bytes.	Python es un lenguaje de programación interpretado.
Longitud del código	Líneas de código más largas en comparación con Python.	3-5 veces más corto que los programas equivalentes en Java.
Complejidad de sintaxis	Define un bloque particular con llaves, finalice las declaraciones con ;	No se necesitan puntos y comas ni llaves, utiliza sangría.

Concepto	Java	Python
Facilidad de escribir	Fuertemente tipado, es necesario definir el tipo de datos exacto de las variables.	Dinámico, no es necesario definir el tipo de datos exacto de las variables.
Velocidad de ejecución	Java es mucho más rápido que Python en términos de velocidad.	Se espera que se ejecute más lento que los programas en Java.
Herencia múltiple	La herencia múltiple se realiza parcialmente a través de interfaces.	Proporciona herencia única y múltiple.

¿Qué es JDK (java Development Kit)?

- ▶ Java Development Kit (JDK) es un entorno de desarrollo de software multiplataforma que ofrece una colección de herramientas y bibliotecas necesarias para desarrollar aplicaciones y subprogramas de software basados en Java. Es un paquete central utilizado en Java, junto con JVM (Java Virtual Machine) y JRE (Java Runtime Environment).
- ▶ JDK contiene:
 - ▶ Un entorno de ejecución en Java (Java Runtime Environment, JRE),
 - ▶ Un intérprete/cargador (Java),
 - ▶ Un compilador (javac),
 - ▶ Un archivador (jar)
 - ▶ y mucho más.

- ▶ Los JDK's más populares:
 - ▶ Oracle JDK: el JDK más popular y principal distribuidor de Java,
 - ▶ OpenJDK: Listo para usar: JDK 19. 18. 17 y varios más,
 - ▶ Azul Systems Zing: JDK eficiente y de baja latencia para sistema operativo Linux,
 - ▶ IBM J9 JDK: Para AIX, Linux, Windows y muchos otros sistemas operativos,
 - ▶ Amazon Corretto: La opción más nueva con la versión gratuita de OpenJDK y soporte a largo plazo.

- ▶ **Windows:**

`https://www.eclipse.org/downloads/`

- ▶ **Mac Os:** `https://mkyong.com/java/how-to-install-java-on-mac-osx/`

Listing 1: Single line comment

```
// System.out.println("This is an comment.");
```

Listing 2: Multi-line comment

```
/*  
System.out.println("This is the first line comment.");  
System.out.println("This is the second line comment.");  
*/
```

Nuestro primer programa en Java

- ▶ **Nombre del archivo:** El nombre de un archivo fuente debe coincidir exactamente con el nombre de la clase pública con la extensión `.java`. El nombre del archivo puede ser diferente si no tiene ninguna clase pública. Por ejemplo:
- ▶ **Nombre de la clase:**
 - ▶ La primera letra de la clase debe estar en mayúscula.
 - ▶ Si se utilizan varias palabras para formar el nombre de la clase, la primera letra de cada palabra interna debe estar en mayúscula.
- ▶ El método `main()` es el principal punto de entrada a un programa Java.

```
public class Example1 {  
    public static void main(String args[]) {  
        System.out.println("Hello world!!!");  
    }  
}
```

Examen Rápido 1 (Introducción a Java)