



Contenido

Introducción al lenguaje de programación Java	2
---	---



Introducción al lenguaje de programación Java

Para poder comenzar a trabajar con Java, es necesario tener una noción de qué se trata el mundo que rodea a éste. Como lenguaje de programación, Java es una herramienta que nos permite escribir y desarrollar aplicaciones de propósitos múltiples. En sí, combina una serie de características presentes en otros lenguajes, pero que logra ofrecer un entorno de desarrollo donde la construcción del software resulta más simple, segura y portable que con el resto.

Sin embargo, Java no es simplemente un lenguaje de programación, podríamos decir más bien que es un *“entorno para el desarrollo y la ejecución de aplicaciones”*.

¿Qué es la tecnología Java?

La tecnología Java es un conjunto de componentes y características que nos brindan las herramientas necesarias para construir software moderno en diferentes ambientes. Podemos decir que la tecnología Java contiene:

- Un Entorno de programación
- Un Entorno de desarrollo
- Un entorno de aplicación
- Un entorno de despliegue

Es un entorno de programación porque nos ofrece una sintaxis, similar a la del lenguaje C++, a través de la cual se puede escribir cualquier tipo de aplicación, como con cualquier otro lenguaje de programación.

Es un entorno de desarrollo porque nos provee una amplia gama de herramientas para el desarrollo de aplicaciones, como ser: un compilador (javac), un intérprete (JVM – Java Virtual Machine), un generador de documentación (javadoc), una herramienta de paquetes de archivos (JAR) entre otras.

Como entorno de aplicación podemos mencionar distintos tipos de aplicaciones que pueden ser construidas mediante Java; aplicaciones de escritorio, applets, aplicaciones cliente servidor, etc.

Como entorno de despliegue, Java nos ofrece una herramienta (JRE – Java Runtime Environment), el cual se encarga de brindar todo lo necesario para que las aplicaciones escritas con Java funcionen de manera correcta.

Objetivos

El lenguaje de programación Java fue diseñado para ser:

- Orientado a objetos



- Distribuido
- Simple
- Multihilo
- Seguro
- Independiente de la plataforma

Orientado a objetos

Para escribir cualquier programa en java es necesario definir los objetos que en él deben intervenir. Esta característica nos da la posibilidad de representar de manera clara la realidad y los problemas que debemos resolver.

Distribuido

Java, en una segunda etapa desde su creación, encontró como destino fundamental para ser aplicado, las tecnologías de redes distribuídas. Otra característica importante de Java es brindar la posibilidad de desarrollar applets, las cuales se descargan a través de Internet y son ejecutadas en los clientes de manera completa.

Simple

Es simple por varios motivos. El hecho de ser orientado a objetos, nos brinda la posibilidad de visualizar el programa en términos de la vida real. Sin embargo, la característica fundamental que hace que Java sea simple, es la no utilización de aritmética de punteros y el manejo de la memoria que afectan la robustez del código.

Multihilo

Esta característica permite que un programa realice tareas no relacionadas al mismo tiempo, o por lo menos sin la necesidad que finalice una para poder comenzar con la siguiente.

Seguro

Los programas basados en la tecnología Java son seguros debido a que el lenguaje de programación Java, junto al entorno en el cual estos programas ejecutan, utilizan medidas de seguridad para protegerse de ataques externos. Estas medidas incluyen:

- ✓ Prohibir el manejo de la memoria a usando punteros.
- ✓ Prohibir a los programas distribuidos, tales como los applets, leer y escribir en el disco rígido del computador.
- ✓ Verificar que todos los programas basados en tecnología java contienen código válido.

Independiente de la plataforma



Esta sin duda es una de las características distintivas y más importantes de la tecnología Java, dado que nos permite escribir un código el cual podrá ser ejecutado en cualquier ambiente que esté preparado para ejecutar aplicaciones escritas en Java.

Características

Ahora bien, ¿A través de qué características, la tecnología Java consigue alcanzar los objetivos para los cuales fue diseñada?

A través de:

- Java Virtual Machine (JVM)
- Java Runtime Environment (JRE)
- Garbage Collector
- Java Development Kit (JDK)

Java Virtual Machine (JVM)

Los programas en java generan el código fuente en archivos de extensión .java. Al igual que los programas escritos en C++, para poder ejecutar este código, es necesario atravesar un proceso de compilación. Sin embargo, a diferencia de C++, el compilador de Java (javac) no genera un archivo binario (.bin), el cual puede ser ejecutado directamente por el SO en una plataforma determinada, sino que genera un archivo intermedio llamado bytecode (.class), el cual debe ser interpretado por un programa especial, en este caso la Java Virtual Machine (JVM).

La JVM es un programa para una plataforma específica que entiende bytecode independiente de la plataforma y que puede ejecutarlo sobre una plataforma particular. Por esta razón, el lenguaje de programación Java es frecuentemente referido como un lenguaje interpretado y los programas con tecnología Java son denominados portables o ejecutables sobre cualquier plataforma.

Para que los programas con tecnología Java sean independientes de la plataforma, se requiere una máquina virtual, sobre cada plataforma donde los programas serán ejecutados.

Java Runtime Environment (JRE)

Sin embargo, un programa basado en tecnología Java necesita más que una JVM para poder ejecutar. Un programa basado en tecnología Java necesita también un conjunto de bibliotecas de clases Java estándar para la plataforma. Las bibliotecas de clases Java (que también podemos llamar API), son bibliotecas de código previamente escrito que pueden ser combinadas con el código que nosotros escribamos para crear aplicaciones robustas. Conjuntamente, la JVM y las bibliotecas de clases conforman el JRE.



En consecuencia, para poder ejecutar un programa escrito en Java simplemente necesitamos disponer del JRE adecuado para la plataforma en la que queramos ejecutarlo. Existen JRE para los entornos más comunes y se puede construir un JRE para cualquier entorno de cualquier característica que necesitemos.

En tiempo de ejecución, los bytecode que constituyen un programa de software son cargados, verificados y ejecutados en un intérprete.

Garbage Collector

Anteriormente mencionamos como uno de los objetivos principales de la tecnología Java, la simpleza. Alcanzándose ésta, entre otras cosas, por la no utilización de la aritmética de punteros y la delegación de la administración de memoria al entorno de ejecución.

Ahora bien, ¿cómo resuelve esta problemática Java? Seguramente ya conocen la respuesta y efectivamente es a través del Garbage Collector. Programa que se encarga de recorrer constantemente (o en los tiempos muertos de la JVM), la tabla de referencias a objetos. Cuando un objeto no es referenciado desde ningún otro objeto (o ninguna función, etc), éste es eliminado y en consecuencia liberada la porción de memoria en la que estaba almacenado.

Esto permite que el programador no tenga que preocuparse por liberar la memoria en ningún momento de la construcción del software. Lógicamente, existe un precio a pagar para lograr esta facilidad que nos ofrece Java. En este caso, el costo serían los recursos que utiliza el Garbage Collector recorriendo constantemente las referencias y limpiando la memoria.

Java Developer's Kit (JDK)

Las tres características anteriormente descritas, están presentes en cualquier entorno capaz de ejecutar aplicaciones de la tecnología Java. Sin embargo, como desarrolladores, necesitamos contar con herramientas que nos brinden el soporte necesario para que estas aplicaciones sean escritas. El Java Developer's Kit (JDK) contiene todo lo necesario para construir aplicaciones Java.

Sin embargo, al ser Java un lenguaje tan versátil, existen distintos grupos de productos dependiendo del tipo de aplicaciones que queramos desarrollar.

- Java 2 Platform, Standard Edition (J2SE)
- Java 2 Platform, Enterprise Edition (J2EE)
- Java 2 Platform, Micro Edition (J2ME)

Java 2 Platform, Standard Edition (J2SE)

El J2SE permite el desarrollo de applets y aplicaciones que ejecutan en el navegador Web y sobre el escritorio del computador, respectivamente.

Java 2 Platform, Enterprise Edition (J2EE)



El J2EE permite crear aplicaciones distribuidas cliente-servidor.

Java 2 Platform, Micro Edition (J2ME)

Permite crear aplicaciones para dispositivos electrónicos con recursos restringidos.

Conclusión

Los kits de desarrollo vienen provistos de todo lo necesario para poder construir y ejecutar las aplicaciones Java, es decir que incluyen todas las características antes descriptas. Al mismo tiempo, existen herramientas que nos brindan un entorno de desarrollo aún más completo y facilitan mucho el proceso de construcción de software como ser los IDE Netbeans o Eclipse, entre otros.

En el presente curso, aprenderemos a construir aplicaciones de escritorio, que representan la base de la programación en Java, y sirve como puerta de entrada a la construcción de aplicaciones web, entre otras. Como IDE recomendamos utilizar Eclipse el cual viene provisto de todo lo necesario para realizar la totalidad de actividades aquí propuestas.

Referencias

Fundamentos del Lenguaje de Programación Java™ SL-110 – Sun Microsystems

El lenguaje de Programación Java™ SL-275 – Sun Microsystems