CONCEPTOS BASICOS DE JAVA

- **ORIGEN DE JAVA**

  En la década de 1990 los microprocesadores están teniendo impacto en los dispositivos electrónicos inteligentes para uso domestico, al conocer esto, SUN MICROSYSTEMS patrocino en el año de 1991 un proyecto interno de investigación denominado Green (verde), el cual desemboco en el desarrollo de un lenguaje basado en c++ al que su creador, james gosling, llamo OAK debido al roble que tenia de vista de la ventana en la oficina en sun.

  Posteriormente se dieron cuenta sun que había un lenguaje de programación con el mismo nombre. Cuando el grupo de gente de sun visito una cafetería local, sugirieron el nombre de java que es un café que proviene de la isla de java, que esta ubicado al sudeste de asia, y de ahí el famoso logo de la tasa de café.

  Pero el proyecto Green tuvo muchas dificultades. El mercado para los dispositivos electrónicos inteligentes de uso domestico no se desarrollaba tan rápido a principios de los noventas como Sun Habia anticipado. EL proyecto corria el riesgo en cancelarse. Pero para su buena fortuna, la popularidad de la world wide web exploto en 1993 y la gente de sun se dio cuenta inmediatamente del potencial de java para agregar contenido dinamico. Como interactividad y animaciones en la pagina web, Esto trajo nuevas ideas al proyecto.

  Sun anuncio formalmente a java en una importante conferencia que tuvo lugar en mayo de 1995. Java genero la atención de la comunidad de negocios debido al fenomenal interés en world wide web. En la actualidad, java se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar la funcionalidad de los servidores web, para proporcionar aplicaciones para los dispositivos domesticos (como teléfonos celulares, radiolocalizadores y asistentes digitales personales) y para muchos propósitos.

- **Conceptos básicos de java**

  Java, al igual que cualquier otro lenguaje de programación, dispone de sus propias particularidades, que representaran una ventaja o desventaja dependiendo aplicación que se vaya a realizar. Por ello se presenta las siguientes características propias de java y proporcionar la visión desde la perspectiva del desarrollar sobre las expectativas que se colocan en java.

CARACTERISTICAS PRINCIPALES DE JAVA

Las características principales que ofrece java respeto a cualquier otro lenguaje de programación.

Simple

Java Ofrece todas las funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos, C++ no es lenguaje convenientes por razones de seguridad, pero c y c++ eran los lenguajes mas difundidos en el nacimiento de java, por ello java se diseño para ser parecido a c++ y asi facilitar un rápido y fácil aprendizaje.

Java elimina muchas de las características de otros lenguajes de programación como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy utiles como el garbage collector (reciclador de memoria dinámica). No es necesario precuparse para liberar memoria, el reciclador se encarga de ellos y como es de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que limita mucho la fragmentación de la memoria.

Java reduce un 50% los errores más comunes de programación con lenguajes como c y c++ al eliminar muchas de las características de estos.

Orientado a objetos.

Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas clases y sus copias, instancias. Estas instancias necesitan ser construidas y destruidas en espacios de memoria.

Distribuido

Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Java en si no es distribuido, sino que proporciona las librerías y herramientas para que los programas pueden ser distribuidos, es decir, que se ejecuten en varias maquinas, interactuando.

Robusto

Java realiza verficaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos de java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo asi la posibilidad de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

Ademas, para asegurar el funcionamiento de la aplicación, realiza una verificación de los bytecodes, que son el resultado de la compilación de un programa en java

Arquitectura Neutral.

Para establecer java como parte integral de la red, el compilador java compila su código a un fichero objeto en formato independiente de la arquitectura de la maquina en que se ejecutar. Cualquier maquina que tenga sistema operativo en ejecución (run-time) puede ejecutarse ese código objeto , independientmente de la maquina en que ha sido generado. Actualmente existen run-time para solaris , Windows Linux MAcOS entre otros.

Seguro

La seguridad en java tiene dos facetas. En el lenguaje, características como los punteros o el casting implícito que hace el compilador de C y C++ se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa java para crear un navegador, se combinan las características del lenguaje con protección de sentidos común aplicadas al propio navegador.

El código de java pasa muchas comprobaciones antes de ejecutarse en una maquina. El código pasa a través de un verificador de bycodes que comprueba el formato de los fragmentos de código y aplica a un probador de teoremas para detectar fragmentos de código ilegal.

Portable

Mas alla de la portabilidad básica por ser de arquitectura independiente, java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre los enteros y, además, enteros de 32 bits en complementosa 2. Ademas java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que estas puedan ser implatadas en etornos unix, pc, mac etc.

Interpretado

El interprete de java (sistema Run-Time) puede ejecutar directamente el código objeto. Enlazar (linkar) un programa normalmente consume menos recursos que compilarlo, por lo que los desarrolladores de java pasaran mas tiempo desarrollando y menos esperando por el ordenador. No obstante, el compilador actual de jdk es bastante lento. Por ahora, no hay compiladores específicos de java para las diversas plataformas, java es mas lento que otros lenguajes de programación ya que debe ser interpretado y no ejecutado como sucede en cualquier programa tradicional. La situación se acerca mucho a la de los programas compilados, sobre todo en lo que a la rapidez en la ejecución del código se refiere.

Multitarea

Java es permite realizar muchas actividades simultaneas en un programa.

Dinamico

Java se beneficia todo lo posible de la tecnología orientada a objetos y no intenta conectar todos los modulos que comprenden una aplicación hasta el mismo timepo de ejecución.

Difundido

Java, en la actualidad, ya no es un lenguaje recién nacido que solamentese conoce en unos cuantos centros de investigación. Java se ha convertido en el lenguaje más difundido en este momento.

Disponibilidad de un amplio conjunto de librerías

Java tiene una amplia posibilidad de utilizar conjunto de clases que ponse a disposición del programador y con las cuales es posible realizar, prácticamente cualquier tipo de aplicaciones.

Encontramos clases para la creación interfaz grafica, gestión de red, multitarea, acceso a datos y etc.

Amplio soporte de fabricante de software.

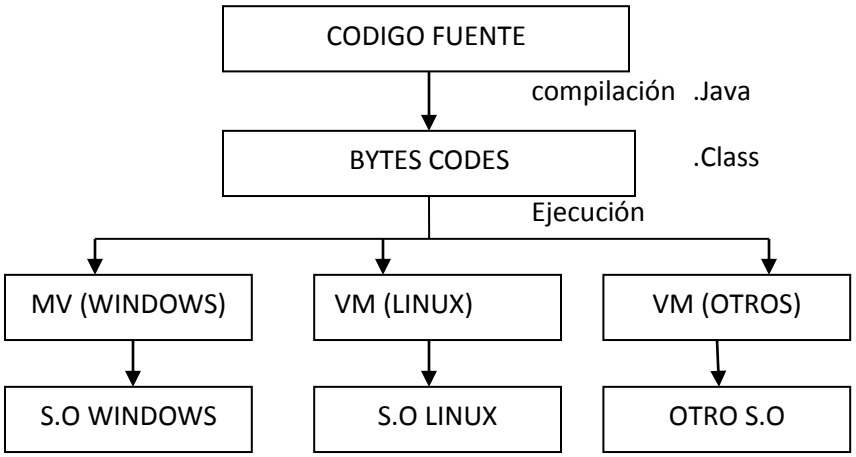
Esta característica se deriva de la parte de las anteriores, sobre todo, del hecho de que los programas java no estén vinculados a un determinado sistema operativo.

Hoy en dia encontramos una amplia variedad de productos de software diferentes fabricantes que dan soporte a java, como puede ser el caso de los entornos de desarrollo o servidores de aplicación.

MAQUINA VIRTUAL JAVA (JVM)

La maquina virtual de java o JVM es un entorno de ejecución para aplicaciones java, cuya principal finalidad es la de adaptar los programas java compilados a las características del sistema operativo donde se va a ejecutar.

En la figura tenemos un esquima en qu ese ilustra todo el proceso de compilacion y ejecucion de las aplicaciones.



Todo programa java esta organizado en clases, estas se codifican en archivos de texto con extensión java. Cada archivo de código fuente .java puede contener una o varias clases. Aunque lo normal es que haya un archivo por clase.

Cuando se compila un .java se genera uno o varios archivos .class de código binario (uno por cada clase), denominados bytecodes, que son independientes de la arquitectura.

Esta independencia supone que los bytecodes no pueden ser ejecutados directamente por ningún sistema operativo; es durante la fase de ejecución cuando los archivos .class se someten a un proceso de interpretación, consistente en traducir los bytecodes a código ejecutablepor el sistema operativo. Esta operación es realizada por un software conocido como maquina virtual java.

Cada sistema operativo proporciona su propia implementación de jvm, todas ellas ofrecen el mismo aspecto de cara de los bytecodes, sin embargo, cada uno realiza la interpretación de acuerdo a las características sistema operativo para el que ha sido diseñada.

EDICIONES DE JAVA.

Java cuenta con 3 ediciones de la tecnología java son:

1. **JAVA 2 STANDAR EDITION (J2SE)** forma parte de este grupo los paquetes de clase de uso general (tratamientos de cadenas, colecciones, acceso a datos, etc), es decir aquellos que se utilizan en cualquier tipo de aplicación. J2SE incluye también los paquetes de clases para la creación de entornos graficos y aplicaciones para navegadores de internet (applets).
2. **JAVA 2 ENTERPRISE EDITION (J2EE**) Proporciona los paquetes y tecnologías necesarias para la creación de aplicaciones empresariales multicapa, entre ellas, las aplicaciones que se van a ejecutar en entorno web.
3. **JAVA 2 MICRO EDITION (J2ME),** Tambien los dispositivos electrónicos, tales como agendas electrónicas, pda o teléfonos móviles, pueden beneficiarse de la tecnología java. Esta edición incluye una serie de pquetes y especificaciones que posibilitan la creación de aplicaciones java ejecutables en dispositivos electrónicos de capacidades limitadas.

JAVA DEVELOPMENT KIT (JDK)

El java development kit proporciona le conjunto de herramientas básico para el desarrollo de aplicaciones con java estándar. Se puede obtener de manera gratuita

.Java

http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u2-download-1377129.html

JAVA RUNTIME ENVIROMENT (JRE)

Proporciona únicamente el entorno de ejecución de las aplicaciones, incluyendo librerías J2SE. Esta es la opción que utilizaríamos si solo quisiéramos ejecutar aplicaciones java creadas por terceros.

ENTORNOS DE DESARROLLO PARA JAVA (IDE)

Cuando se va a desarrollar una aplicación que puede contar con un elevado número de líneas de código y va estar constituido de varias clases, la utilización de las herramientas del SDK para la compilación y ejecución de los programas puede resultar engorrosas, además de dificultar la detección y solución de errores, tanto de compilación como de ejecución.

En este casos resulta mucho mas practica la utilización de un entorno de desarrollo integrado (IDE).
Un IDE proporciona todos los elementos indispensables para la codificación, compilación,depuración y ejecución de programas dentro de un entorno grafico amigable y fácil de utilizar.

Los IDE para java utilizan internamente las herramientas básicas del JDK en la realización de estas operaciones, sin embargo, el programador no tendrá que hacer uso de la consola para ejecutar esos comando, dado que el entorno le ofrecerá una forma alternativa de utilización, basado en menus y barras de herramientas.

IDENTIFICADORES Y PALABRA CLAVE

IDENTIFICADORES

Los Identificadores nombran variables, funciones, clases y objetos cosa que el programador necesite identificar o usar.

En java, un identificador comienza con una letra, un subrayado(_) o un símbolo de dólar ($), los siguientes caracteres pueden ser letras o digitos. Se distinguen las mayúsculas de las minúsculas y no hay una longitud máxima establecida para el identificador. La forma básica de una declaración de variable, por ejemplo, seria:

Tipo identificador [=valor][,identificador[= valor]….];

Serian nombre identificadores validos:
Identificador
Nombre_usuario
Nombre_Usuario
_variable _del_sistema
$transaccion

Y su uso será, por ejemplo:

Int contador_principal;
Char _lista_de_ficheros;
Float $cantidad_en_Pesos;

PALABRAS CLAVE

Las palabras clave son las palabras que están definidas en java y que no se pueden utilizar como identificadores:

Abstract	Continue	For	New	swtich
Assert	Default	Goto	Null	synchronized
Boolean	Do	If	Package	This
Break	Double	Implements	Prívate	threadsafe
Byte	Else	Import	Protected	throw
Byvalue	Enum	Instanceof	Public	transient
Case	Extends	Int	Return	true
Catch	False	Interface	Short	try
Char	Final	Long	Static	void
Class	Finally	Native	Super	while
Const	Float			

Palabras reservadas

El lenguaje se reserva unas cuantas palabras más, pero que hasta ahora no tienen un cometido especifico:

Cast	Future	Generic	Inner
Operator	Outer	Rest	var

- **Tipos de datos**

Java utiliza cinco tipos de elementos: enteros, reales en como flotantes, booleanos, caracteres y cadenas, que se pueden poner en cualquier lugar del código fuente de java:

Cada uno de estos literales tienen un tipo correspondiente asociado con el:

ENTEROS:

BYTE	8 BITS	COMPLEMENTO A DOS
SHORT	16 BITS	COMPLEMENTO A DOS
INT	32 BITS	COMPLEMENTOS A DOS
LONG	64 BITS	COMPLEMENTOS A DOS

EJEMPLO: INT X= 2 COMPLEMENTO 2 0010

REALES EN COMA FLOTANTES

Float	32 bits	IEEE 754
DOUBLE	64 BITS	IEEE 754

EJEMPLO Y = 3.4
EJEMPLO $Z=1.3X10^{-45}$ JAVA ES 1.3e-45f

BOOLEANOS

TRUE
FALSE

CARACTERES

CARÁCTER EJEMPO X=’A’
CADENAS DE CARACTERES B=”HOLA HOY”

String nombre=”hola como estas”;

Constantes y variables

Una constante es una variable del sistema que mantiene un **valor inmutable a lo largo de toda la vida del programa**. Las constantes en Java se definen mediante **el modificador final**.

Variable: Variable se utilizan en la programación java para almacenar datos que varían durante la ejecución del programa. Para usar una variable, hay que indicarle al compilador el tipo y nombre de esa variable. Declaración de esa variable. EL tipo de la variable determina el conjunto de valores que se podrán almacenar en la variable y el tipo de operaciones que podrán realizar con ella.