Introducción a Java

Java es un poderoso lenguaje de programación que ha revolucionado el mundo de la tecnología. Desarrollado por Sun Microsystems a principios de los años 90, Java se ha convertido en una herramienta indispensable para crear una amplia variedad de aplicaciones, desde sitios web y juegos hasta sistemas empresariales y dispositivos móviles. En esta presentación, exploraremos los conceptos fundamentales de Java y cómo su estructura y ejecución hacen de él un lenguaje tan versátil y eficiente.

Anatomía de un programa Java

Clases y Métodos

Todo programa Java se basa en la creación de clases, que actúan como moldes para objetos. Dentro de estas clases, se definen los métodos, que son las acciones que los objetos pueden realizar. La estructura y organización de clases y métodos es fundamental para el diseño y la funcionalidad de un programa Java.

Sintaxis y Estructura

Java tiene una sintaxis clara y estricta, con reglas específicas para la declaración de variables, la escritura de sentencias de control y la formatación del código. Comprender esta sintaxis es crucial para escribir programas Java correctos y legibles.

Bibliotecas y APIs

Java cuenta con una extensa biblioteca de clases y métodos preexistentes, conocida como Java API (Application Programming Interface). Estas APIs facilitan el desarrollo de aplicaciones al proporcionar funcionalidades comunes, como manejo de archivos, interfaz gráfica de usuario y conexiones de red.

Ejecución de código

1

Código Fuente

El proceso comienza con el código fuente escrito por el desarrollador en un editor de texto o un entorno de desarrollo integrado (IDE).

2

Compilación

El código fuente se compila utilizando un compilador Java, que traduce el código a bytecode, un lenguaje de bajo nivel que puede ser interpretado por la Máquina Virtual de Java (JVM).

3

Ejecución

La JVM ejecuta el bytecode, lo que permite que el programa Java se ejecute en diferentes plataformas y sistemas operativos sin tener que recompilar el código.

Proceso de Compilación Java

1

Código Fuente

El desarrollador escribe el código Java en un archivo de texto con la extensión ".java".

2

Compilador Java

El compilador Java, como el javac, procesa el código fuente y lo traduce a bytecode, un formato de bajo nivel que puede ser interpretado por la Máquina Virtual de Java (JVM).

3

Bytecode

El bytecode generado por el compilador se almacena en archivos con la extensión ".class", que pueden ser ejecutados por la JVM en diferentes plataformas.

Variables en Java

1

Declaración

Las variables en Java se declaran utilizando un tipo de datos específico, seguido del nombre de la variable. Por ejemplo: "int edad = 25;".

2

Inicialización

Las variables pueden inicializarse al momento de la declaración o en un momento posterior en el código. Es importante inicializar las variables antes de usarlas.

3

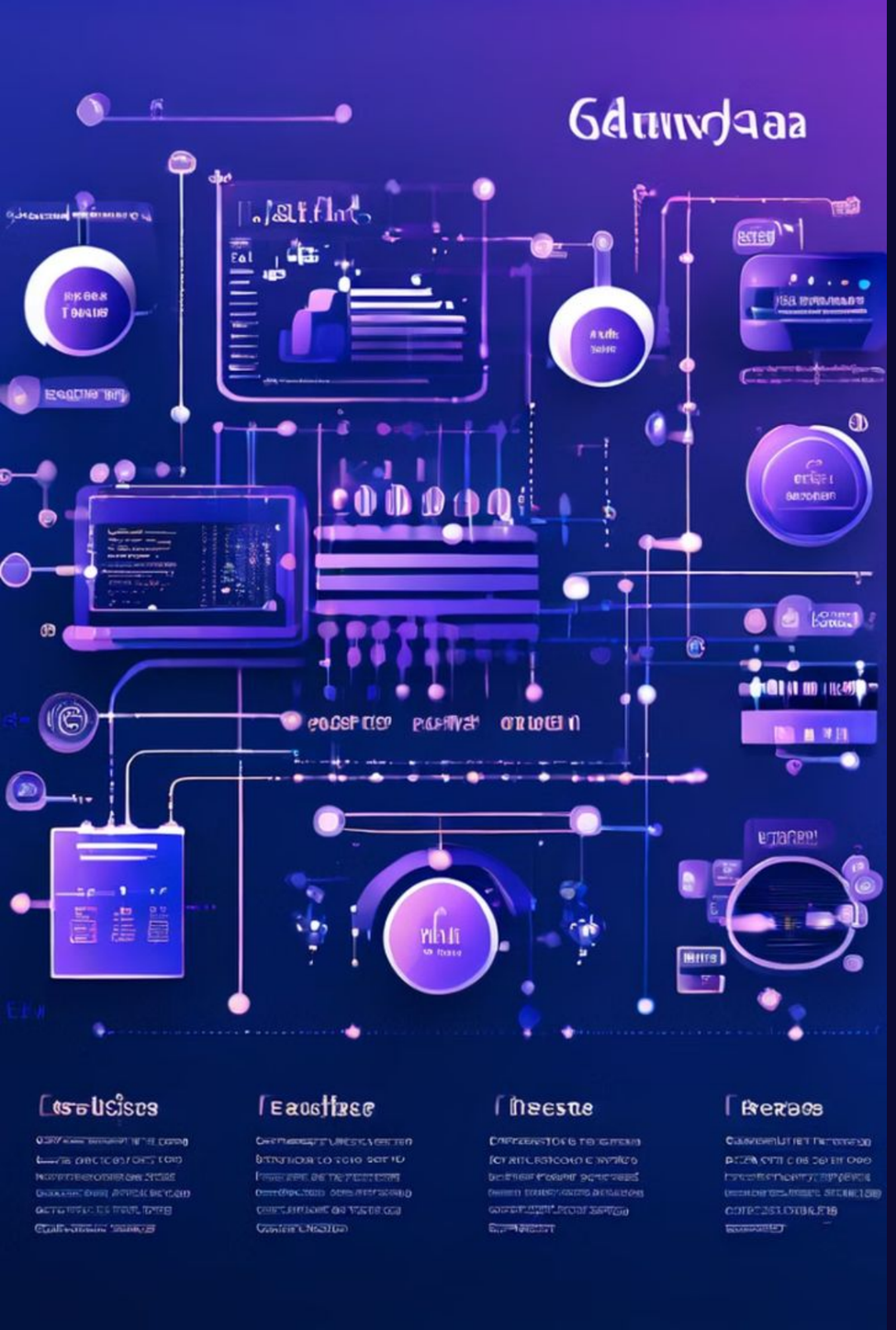
Ámbito

El ámbito de una variable determina dónde puede ser accedida y modificada. Las variables pueden tener ámbitos de clase, método o bloque.

4

Convenciones

Java tiene convenciones de nomenclatura para las variables, como utilizar nombres descriptivos y seguir un estilo de escritura, como camelCase.



Tipos de Datos en

Primitivos

Java tiene ocho tipos de datos primitivos, como int, double y boolean, que representan valores básicos como números enteros, decimales y valores lógicos.

No Primitivos

Además de los tipos primitivos, Java también cuenta con tipos de datos no primitivos, como String, Arrays y Clases personalizadas, que permiten representar estructuras de datos más complejas.

Conversión de Tipos

Java permite la conversión explícita e implícita entre tipos de datos, lo que facilita el manejo de diferentes formatos de información en un programa.

Nominación y Tamaños

Cada tipo de dato en Java tiene un nombre y un tamaño de almacenamiento específico, lo que afecta la precisión y el rango de valores que pueden representar.

Operadores en Java

Aritméticos

Los operadores aritméticos, como +, -, *, / y %, permiten realizar operaciones matemáticas básicas en variables y valores.

Lógicos

Los operadores lógicos, como &&, || y !, se utilizan para evaluar expresiones booleanas y tomar decisiones en el código.

Asignación

Los operadores de asignación, como =, +=, -=, etc., permiten modificar el valor de una variable de forma concisa.

Comparación

Los operadores de comparación, como ==, !=, <, > y <=, se utilizan para evaluar y comparar valores en el código.

Estructuras de Control en Java

If-Else	Permite tomar decisiones basadas en condiciones, como if (edad > 18) { ... } else { ... }
Loops	Permiten repetir una serie de instrucciones, como for, while y do-while
Switch	Evalúa una expresión y ejecuta un bloque de código específico según su valor, como switch (mes) { case 1: ... }
Break y Continue	Permiten controlar el flujo de ejecución dentro de los bucles, interrumpiéndolos o saltando a la siguiente iteración