Fundamentos de programación en Java

PORTILLA IMBAQUINGO LUIS
FABIAN

ÍNDICE

Tabla de contenido

1. Fundamentos De Programación En Java

Los fundamentos de programación en Java abarcan varios conceptos clave que son esenciales para cualquier desarrollador que trabaje con este lenguaje de programación. se centra en el tratamiento de las operaciones como evaluación de funciones matemáticas y en la composición de funciones. Java adoptó características de programación funcional en su versión 8, lo que permitió a los desarrolladores utilizar expresiones lambda y aprovechar las interfaces funcionales. Este enfoque facilita el desarrollo de código más conciso y legible, así como la escritura de programas más robustos y modularizados.

2. ¿Qué es Java?

Java es un lenguaje de programación de propósito general que fue desarrollado por Sun Microsystems en la década de 1990. Es conocido por ser portátil, orientado a objetos, de alto rendimiento y tener amplias aplicaciones, desde aplicaciones empresariales hasta desarrollo de aplicaciones web y móviles. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo.Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías delservidor.

2.1. ¿Para qué se utiliza el lenguaje de programación Java?

Debido a que Java es un lenguaje versátil y de uso gratuito, crea software localizado y distribuido. Algunos usos comunes de Java incluyen:Desarrollo de videojuegos

Muchos videojuegos, así como juegos para móviles y computadoras, se crean con Java. Incluso los juegos modernos que integran tecnología avanzada, como el machine learning o la realidad virtual, se crean con la tecnología de Java.

1. Computación en la nube

Java a menudo se conoce como WORA: escribir una vez y ejecutar en cualquier lugar (por sus siglas en inglés “Write Once and Run Anywhere”), lo que lo hace perfecto para aplicaciones descentralizadas basadas en la nube. Los proveedores de la nube eligen el lenguaje Java para ejecutar programas en una amplia gama de plataformas subyacentes.

2. Macrodatos

Java se usa para motores de procesamiento de datos que pueden trabajar con conjuntos de datos complejos y cantidades masivas de datos en tiempo real.

3. Inteligencia artificial

Java es una fuente inagotable de bibliotecas de machine learning. Su estabilidad y velocidad lo hacen perfecto para el desarrollo de aplicaciones de inteligencia artificial como el procesamiento del lenguaje natural y el aprendizaje profundo.

4. Internet de las cosas

Java se ha utilizado para programar sensores y hardware en dispositivos de periferia que pueden conectarse de forma independiente a Internet.

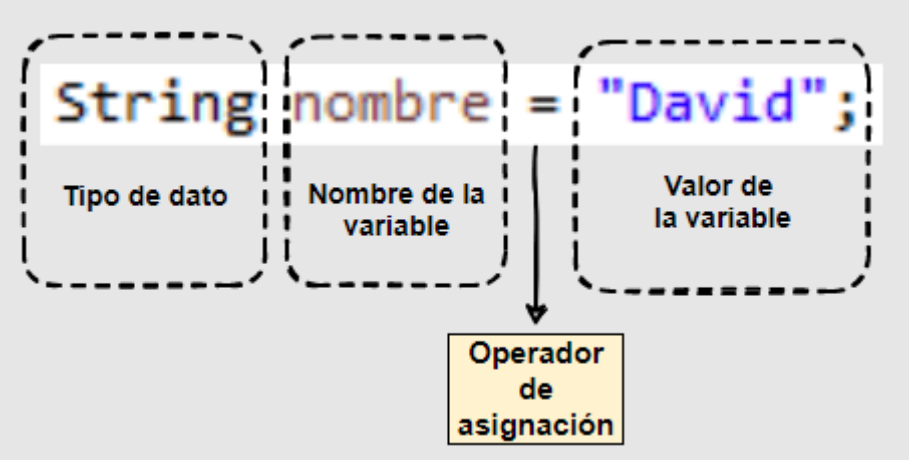
3. ¿Qué son los datos en Java?

En el mundo de la programación Java, los “datos” son la esencia de la información que nuestros programas manipulan y utilizan para llevar a cabo diversas operaciones. Los datos representan la unidad de información que nuestras funciones y métodos comprenden.

Estos datos pueden adoptar múltiples formas y representar una variada gama de información, desde números y texto, pasando por conjuntos de valores, hasta valores booleanos que señalan verdadero o falso.

Java brinda a los programadores una serie de tipos de datos predefinidos que facilitan la declaración de variables y el almacenamiento eficiente de información.

Además, es conocido por ser un lenguaje de programación “fuertemente tipado”, lo que significa que los tipos de datos de las variables y expresiones deben ser declarados y verificados en tiempo de compilación.



3.1. Las ventajas del tipado de variables en Java son:

- Seguridad y prevención de errores: El tipado fuerte reduce la posibilidad de errores al forzar la coincidencia precisa de tipos de datos, lo que evita operaciones inapropiadas.
- Claridad y legibilidad del código: Los tipos de datos declarados hacen que el código sea más comprensible y facilitan la colaboración entre programadores.
- Rendimiento y optimización: Al conocer los tipos de datos de antemano, Java puede realizar optimizaciones que mejoran la velocidad de ejecución.
- Detección temprana de errores: Los errores se detectan en tiempo de compilación, lo que permite corregirlos antes de que el programa se ejecute, reduciendo así los errores en tiempo de ejecución.
- Facilita el mantenimiento y la refactorización: Los tipos de datos explícitos hacen que sea más fácil realizar cambios en el código sin introducir errores inadvertidos.
- Conviértete en un Backend Developer
- Domina los lenguajes de programación más demandados. Accede a cursos, talleres y laboratorios para crear proyectos con Java, Python, PHP, Microsoft .NET y más

4. Tipos de datos en Java

Los tipos de datos en Java se dividen en dos categorías principales: tipos de datos primitivos y tipos de datos de referencia.

5. Datos primitivos

Los datos primitivos son tipos de datos básicos que representan valores simples y se almacenan directamente en la memoria.

Algunos ejemplos de datos primitivos en Java incluyen int para números enteros, double para números decimales y boolean para valores lógicos verdaderos o falsos, entre otros.

5.1.La lista de datos primitivos puede ser:

5.2.Numéricos:

byte: Representa un tipo de dato de 8 bits con signo. Puede almacenar valores numéricos en el rango de -128 a 127, incluyendo ambos extremos. Es útil para ahorrar memoria cuando se necesita almacenar valores pequeños.

short: Este tipo de dato utiliza 16 bits con signo y puede almacenar valores numéricos en el rango de -32,768 a 32,767. Se utiliza cuando se necesita un rango más amplio que el proporcionado por los bytes, pero aún se desea ahorrar memoria en comparación con los tipos de dato más grandes.

int: Es un tipo de dato de 32 bits con signo utilizado para almacenar valores numéricos. Su rango va desde -2,147,483,648 (-2^31) hasta 2,147,483,647 (2^31 - 1). Es el tipo de dato más comúnmente utilizado para representar números enteros.

long: Este tipo de dato utiliza 64 bits con signo y puede almacenar valores numéricos en el rango de -9,223,372,036,854,775,808 (-2^63) a 9,223,372,036,854,775,807 (2^63 - 1). Se utiliza cuando se necesitan números enteros muy grandes.

float: Es un tipo de dato diseñado para almacenar números en coma flotante con precisión simple de 32 bits. Se utiliza cuando se requieren números decimales con un grado de precisión adecuado para muchas aplicaciones.

double: Este tipo de dato almacena números en coma flotante con doble precisión de 64 bits, lo que proporciona una mayor precisión que float. Se usa en aplicaciones que requieren una alta precisión en cálculos numéricos.

5.3.Carácter:

char: Es un tipo de datos que representa un carácter Unicode sencillo de 16 bits. Se utiliza para almacenar caracteres individuales, como letras o símbolos en diferentes lenguajes y conjuntos de caracteres.

5.4.Boléanos:

boolean: Sirve para definir tipos de datos booleanos que pueden tener solo dos valores: true o false. Aunque ocupa solo 1 bit de información, generalmente se almacena en un byte completo por razones de eficiencia.

5.5.Table de resumen de los tipos primitivos en Java:

6. Datos de referencia

Los datos de referencia son tipos de datos más complejos que hacen referencia a objetos almacenados en memoria. Estos objetos pueden ser instancias de clases personalizadas o clases predefinidas en Java, como String.

Los datos de referencia no almacenan directamente el valor, sino una referencia a la ubicación en memoria donde se encuentra el objeto.

6.1. Datos de referencia especiales

Java también tiene tipos de datos de referencia especiales, como null, que representa la ausencia de un objeto, y void, que se utiliza en métodos que no devuelven ningún valor.

7. Qué son las variables en Java

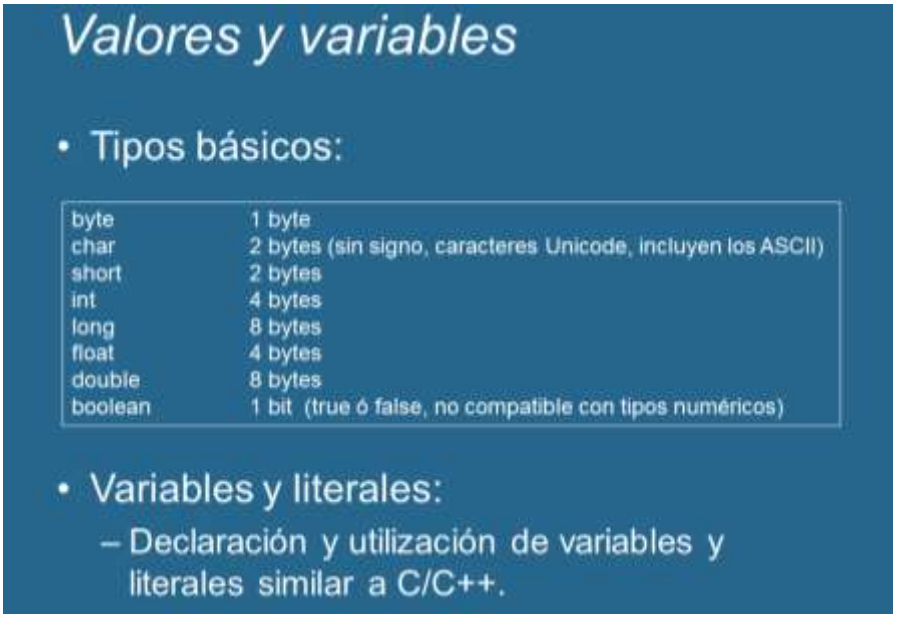
Las variables en Java son contenedores que se utilizan para almacenar y manipular datos en un programa.

Una variable tiene un tipo de dato que define qué tipo de valor puede contener y un nombre que se utiliza para hacer referencia a ella en el código.

7.1. Tipos de variables en Java

En Java, las variables se dividen en tres tipos principales: variables locales, variables de instancia y variables de clase.

- Las variables locales se declaran en un método y solo son accesibles dentro de ese método.
- Las variables de instancia pertenecen a una instancia específica de una clase y se declaran dentro de la clase, pero fuera de cualquier método.
- Las variables de clase son compartidas por todas las instancias de una clase y se declaran utilizando la palabra clave static.



8. Estructura de un programa java:

Un programa describe cómo un ordenador debe interpretar las órdenes del programador para que ejecute y realice las instrucciones dadas tal como están escritas. Un programador utiliza los elementos que ofrece un lenguajede programación para diseñar programas que resuelvan problemas concretos o realicen acciones bien definidas.

El siguiente programa Java muestra un mensaje en la consola con el texto “Hola Mundo”.

```
/*
* Este programa escribe el texto "Hola Mundo" en la consola
* utilizando el método System.out.println()
*/

public class HolaMundo {

   public static void main (String[] args) {
      System.out.println("Hola Mundo");
   }

}
```

9. La estructura de un programa Java

En este programa se pueden identificar los siguientes elementos del lenguaje Java: comentarios, definiciones de clase, definiciones de método y sentencias.

Comentario. El programa comienza con un comentario. El delimitador de inicio de un comentario es /* y el delimitador de fin de comentario es */. El texto del primer comentario de este ejemplo sería: ‘Este programa escribeel texto “Hola Mundo” en la consola utilizando el método System.out.println()'. Los comentarios son ignorados por el compilador y solo son útiles para el programador. Los comentarios ayudan a explicar aspectos relevantes de un programa y lo hacen más legible. En un comentario se puede escribir todo lo que se desee, el texto puede ser deuna o más líneas.

Definición de clase. La primera línea del programa, después del primer comentario. Define una clase que se llama `HolaMundo`. La definición de la clase comienza por el carácter { y termina con el carácter }. El nombre dela clase lo define el programador.

Definición de método. Después de la definición de clase se escribe la definición del método `main()`. Todos los programas Java deben incluir un método `main()`. Este método indica las sentencias a realizar cuando se ejecuta un programa. Un método es una secuencia de sentencias ejecutables. Las sentencias de un método quedan delimitadas por los caracteres { y } que indican el inicio y el fin del método, respectivamente.

Sentencia. Dentro del método `main()` se incluye una sentencia para mostrar un texto por la consola. Los textos siempre se escriben entre comillas dobles para diferenciarlos de otros elementos del lenguaje. Todas las sentencias de un programa Java deben terminar con el símbolo punto y coma. Este símbolo indica al compilador que ha finalizado una sentencia.

Una vez que el programa se ha editado, es necesario compilarlo y ejecutarlopara comprobar si es correcto. Al finalizar el proceso de compilación, el compilador indica si hay errores en el código Java, dónde se encuentran y eltipo de error que ha detectado: léxico, sintáctico o semántico.

```
/* Este programa calcula el perímetro de una circunferencia */

public class PerimetroCircunferencia {
   public static void main (String[] args) {

      // declaración de PI y la variables radio y perimetro

      final double PI = 3.1415926536;
      double radio = 25.0, perimetro;

      perimetro = 2.0*PI*radio;
      System.out.print("El perimetro de la circunferencia de radio ");
      System.out.print(radio);
      System.out.print(" es ");
      System.out.print(perimetro);

   }
}
```

Las palabras reservadas de Java

En todos los lenguajes de programación existen palabras con un significado especial. Estas palabras son reservadas y no se pueden utilizar como nombres de variables.

abstract	final	public
assert	finally	return
boolean	float	short
break	for	static
byte	if	strictfp
case	implements	super
catch	import	switch
char	instanceof	synchronized
class	int	this
continue	interface	throw
default	long	throws
do	native	transient
double	new	true
else	null	try
enum	package	void
extends	private	volatile
false	protected	while

10. Bibliografía:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DZUmYAsAPOwQ&psig=AOvVaw20tqrOHP49QrCueZ4V03rz&ust=1708987317097000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCPiilYHIx4QDFQAAAAAdAAAAABAZ

https://www.deleonnet.com/blog/recomendaciones-desarrollo-web/fundamentos-de-programacion

https://www.java.com/es/download/help/whatis_java.html

https://aws.amazon.com/es/what-is/java/

https://czstore.online2023sale.ru/?c=tipos%20de%20variables%20en%20java