CLASE 4. Operadores, Estructuras de Control y Clase Scanner

Variables y Constantes en Java

Una variable es un espacio en la memoria donde se almacenan valores que pueden cambiar durante la ejecución del programa. En Java, las variables deben ser declaradas especificando su tipo, lo que garantiza el tipo de dato que almacenarán. A lo largo de su ciclo de vida, el valor de una variable puede ser reasignado.

Por otro lado, las **constantes** son valores fijos que, una vez asignados, no pueden cambiar. En Java, las constantes se definen utilizando la palabra clave `final`. Es importante utilizarlas para garantizar la integridad de los valores que no deben modificarse en la ejecución del programa.

Ejemplo de Variables:

```
int numero = 10;
String nombre = "Juan";
```

En el ejemplo anterior, hemos declarado dos variables: `numero`, de tipo entero, y `nombre`, de tipo cadena. Ambas variables pueden cambiar su valor posteriormente en el programa.

Ejemplo de Constantes:

```
final double PI = 3.1416;
```

En este caso, la constante `PI` no podrá ser modificada una vez inicializada, asegurando que su valor sea inmutable durante la ejecución del programa.

Operadores en Java

Java ofrece una amplia gama de operadores que permiten realizar operaciones sobre los datos. Estos operadores se agrupan en diferentes categorías, según el tipo de operación que ejecutan.

Operadores Aritméticos

Los operadores aritméticos permiten realizar cálculos matemáticos entre variables y constantes. Los operadores principales son:

- `+` (suma)
- `-` (resta)
- `*` (multiplicación)
- `/` (división)
- `%` (módulo)

Ejemplo:

```
int resultado = 10 + 5;  // Resultado es 15
```

Operadores Relacionales

Los operadores relacionales permiten comparar valores, devolviendo un resultado booleano (`true` o `false`). Los operadores son:

- `==` (igual a)
- `!=` (diferente de)

- `>` (mayor que)
- `<` (menor que)
- `>=` (mayor o igual que)
- `<=` (menor o igual que)

Ejemplo:

```
boolean esMayor = 10 > 5;  // esMayor es true
```

Operadores Lógicos

Los operadores lógicos se utilizan para combinar expresiones booleanas, permitiendo evaluar múltiples condiciones simultáneamente. Los operadores son:

- `&&` (AND lógico)
- `||` (OR lógico)
- `!` (NOT lógico)

Ejemplo:

```
boolean resultado = (10 > 5) && (8 < 3);  // resultado
es false
```

Expresiones y Asignaciones

Una **expresión** en Java es una combinación de variables, operadores y valores que Java evalúa para producir un resultado. Las expresiones más comunes son aquellas que involucran operadores aritméticos, relacionales o lógicos.

Una **asignación** es el proceso mediante el cual un valor es almacenado en una variable. Esto se realiza utilizando el operador `=`.

Ejemplo:

```
int x = 10 + 5;  // Se evalúa la expresión 10 + 5 y se asigna el resultado a x
```

En este ejemplo, la expresión `10 + 5` se evalúa primero, y luego el resultado (15) se asigna a la variable `x`.

Estructuras de Control

Las estructuras de control en Java permiten alterar el flujo de ejecución de un programa en función de condiciones y repeticiones. Las más comunes son las estructuras condicionales y los bucles.

Estructura Condicional: If

La sentencia `if` permite ejecutar un bloque de código si una condición es verdadera. Si la condición es falsa, el programa continúa con el siguiente bloque de código.

Ejemplo:

```
int numero = 10;
if (numero > 5) {
    System.out.println("El número es mayor que 5.");
}
```

En este caso, si la variable `numero` es mayor que 5, se imprimirá el mensaje en la consola.

Bucles

Los bucles permiten ejecutar un bloque de código repetidamente hasta que se cumpla una determinada condición. Los bucles más utilizados en Java son `for`, `while` y `do-while`.

Clase Scanner en Java

La clase **Scanner** es parte del paquete `java.util` y se utiliza para obtener la entrada del usuario a través de la consola. Esto es particularmente útil cuando queremos que el usuario ingrese datos durante la ejecución del programa.

Importar la Clase Scanner

Antes de utilizar la clase Scanner, es necesario importarla en nuestro programa. Esto se realiza con la siguiente línea de código:

```
import java.util.Scanner;
```

Declarar y Crear un Objeto Scanner

Para utilizar la clase Scanner, debemos crear una instancia de esta. Un objeto Scanner permite leer diferentes tipos de datos desde la entrada estándar (generalmente, el teclado).

Ejemplo:

```
Scanner entrada = new Scanner(System.in);
```

En este ejemplo, `entrada` es un objeto de la clase Scanner que leerá los datos introducidos por el usuario a través de la consola.

Métodos de la Clase Scanner

La clase Scanner proporciona varios métodos para leer distintos tipos de datos:

- `nextInt()`: Lee un número entero.
- `nextDouble()`: Lee un número decimal.
- `nextLine()`: Lee una línea completa de texto.
- `next()`: Lee una palabra.

Ejemplo:

```
Scanner entrada = new Scanner(System.in);
System.out.print("Ingresa tu nombre: ");
String nombre = entrada.nextLine();
System.out.println("Hola, " + nombre);
```

En este ejemplo, el programa solicitará al usuario que ingrese su nombre, y luego lo imprimirá en la consola.

El dominio de los operadores, estructuras de control y la clase Scanner en Java es esencial para escribir programas eficientes y funcionales. Con estas herramientas, podemos realizar cálculos, tomar decisiones basadas en condiciones, repetir tareas automáticamente y, además, interactuar con el usuario de manera dinámica. El conocimiento profundo de estas funcionalidades te permitirá desarrollar aplicaciones sólidas y escalables en Java.