 



Conceptos fundamentales de Java 4-3: Tipos de datos y operadores Actividades prácticas

# Objetivos de la lección:

* Usar tipos de datos primitivos en código Java
* Especificar literales para los tipos primitivos y para las cadenas
* Demostrar cómo inicializar variables
* Describir las reglas de alcance de un método
* Identificar la necesidad de conversión de tipo en una expresión
* Aplicar la conversión en código Java
* Usar operadores aritméticos
* Usar el operador de asignación
* Usar un método de la clase Math
* Acceder a un método de la clase Math de la API de Java

**Vocabulario**

Identifique el término correspondiente a cada una de las definiciones a continuación.

|  |  |
| --- | --- |
| **Variable y constante** | Mecanismos de almacenamiento de datos primitivos u objetos con nombre definidos en un programa. El valor asignado puede cambiar o no (constantes). |
| **Operador aritmético** | Se utilizan símbolos para sumar, restar, multiplicar y dividir, y aritmética modular en expresiones y fórmulas matemáticas. |
| **Datos primitivos** | Grupo de tipos de datos de Java que no utiliza la palabra clave new cuando se declara o inicializa. Los tipos de datos primitivos almacenan el valor en el mismo lugar de la memoria que el nombre de la variable. |
| **byte** | El tipo primitivo de Java más pequeño (1 byte) que puede contener un valor entero. |
| **long** | Este tipo de dato (8 bytes) es el tipo entero más grande. |
| **Convenciones de programación** | Convenciones de formato y nomenclatura que siguen las mayoría de los programadores. |
| **int** | Este tipo de dato primitivo de Java (4 bytes) puede contener valores enteros. |
| **long** | Este tipo de dato primitivo de Java (8 bytes) es el primitivo más grande que puede contener un valor decimal. |
| **Inicialización** | Cuando se asigna un valor a una variable por primera vez. |
| **float** | Este tipo de dato primitivo de Java (4 bytes) puede inicializarse con un número decimal precediendo a la letra f. Por ejemplo: float x = 3.5f; |
| **Valor inicial** | Puede ser cualquier número, texto o información que represente un valor; se utiliza para inicializar un tipo primitivo. |
| **Declaración de variable** | Sentencia Java cuando se define una variable pero no se le asigna necesariamente un valor. Ejemplo: int x; |
| **Precedencia** | Esta palabra describe la prioridad matemática de una variable en un programa Java. |
| **char** | Este tipo de dato primitivo de Java (2 bytes) puede contener valores de un solo carácter. Ejemplo: “a”, “#”, o “X” |
| **corchetes** | Se utilizan para describir el bloque de código cuando existe una variable en un programa. {} indica la existencia de un bloque de código. |
| **Conversión de tipo** | Proceso de modificar explícitamente un tipo de dato para que se convierta en otro tipo de dato diferente. |
| **Redondeo** | Concepto por el que un número se redondea siempre al entero más cercano. |
| **Asignación** | Signo igual “=” utilizado en una sentencia Java para asignar un valor a una variable. |
| **Conversión de tipo** | Proceso de modificar un tipo de dato para que se convierta en otro tipo de dato diferente, ya sea este implícito o explícito. |
| **short** | Tipo de dato primitivo de Java (2 bytes) que contiene números enteros dentro de un rango más corto que int. |
| **boolean** | Tipo primitivo de Java de un bit que puede contener un valor true |

# Inténtelo/resuélvalo:

1. Escriba un programa que tome la base y altura de un triángulo y calcule y muestre la superficie de este utilizando la fórmula a continuación.

𝐴 =

1 𝑏ℎ

2

1. Escriba las siguientes fórmulas matemáticas en Java. Tendrá que usar métodos de la clase Math así como una incrustación de métodos y paréntesis para forzar el orden de las operaciones y calcular correctamente la respuesta. Suponga que todas las variables de las fórmulas ya se han declarado e inicializado.

a. 𝑎 = √𝑥5−6

4

b. 𝑏 = 𝑥𝑦 − 6𝑥

𝑧𝑧

c.

2

𝑐 = 4𝑐𝑜𝑠 � � − 𝑠𝑠𝑠𝑛𝑥

5

d. 𝑑 = 𝑥4 − �6𝑥 − 𝑦3

e. 𝑒 = 1

𝑦− 1

𝑥−2𝑦

f. 𝑓𝑓 = 7(𝑐𝑜𝑠(�5 − 𝑠𝑠𝑠𝑛√3𝑥 − 4))

1. Un autobús lleva 45 personas. El colegio solo utilizará el autobús si puede llenarlo del todo. El resto de personas irá en furgonetas. Escriba un programa que registre el número de personas que se han apuntado para ir de viaje. Haga que el programa muestre el número necesario de autobuses y, después, el número total de personas que tendrá que viajar en furgonetas.
2. Escriba true o false en los espacios en blanco del programa a continuación para mostrar el valor de la variable booleana true\_false mientras se ejecuta el programa.

int i=5; int j=6;

boolean true\_false; true\_false=(j<5); true\_false=(j>3); true\_false=(j<i); true\_false=(i<5); true\_false=(j<=5); true\_false=(6<6); true\_false=(i!=j); true\_false=(i==j || i<50); true\_false=(i==j && i<50);

false

false

true

false

true

false

false

false

true

false

true\_false=(i>j || true\_false && j>=4); true\_false=(!(i<2 && j==5)); true\_false=!true\_false;

false

true

1. Explique por qué las declaraciones de la segunda lista son erróneas.

boolean gameOver = false; int students=50,classes=3; double sales\_tax;

short number1;

int 2beOrNot2be; float price index;

double lastYear'sPrice;

long class;

* + - Las declaraciones de la segunda lista son erróneas por las siguientes razones:
    - int 2OrNot2; : los nombres de variables no pueden comenzar con un número, por lo que este nombre no es válido.
    - float price index;: el espacio no es un carácter válido en un nombre de variable, por lo que este nombre no es válido. Se podría utilizar un guión bajo en su lugar: float price\_index;.
    - double lastYear'sPrice;: las comillas simples no son válidas en un nombre de variable, por lo que este nombre no es válido. Se podría utilizar un guión bajo en su lugar: double lastYear\_sPrice;.
    - long class;: "class" es una palabra clave en Java que se utiliza para definir clases. No se pueden utilizar palabras clave como nombres de variables, por lo que este nombre no es válido. Se podría utilizar un nombre diferente: long classSize;.

1. Explique por qué las declaraciones de la segunda lista no siguen las convenciones para los nombres de variables.

int cadence=3, speed=55, gear=4;

final double SALES\_TAX=.06; double gearRatio=.5;

int currentGear=5;

int c=3,s=55,g=4;

final double salesTax=.06; double gearratio=.05,Gear=4; int current\_gear;

Las declaraciones de la segunda lista no siguen las convenciones para los nombres de variables por las siguientes razones:

* int c=3,s=55,g=4;: los nombres de variables deben ser descriptivos y legibles. Utilizar letras simples como nombres de variables no proporciona información útil sobre su propósito o uso en el programa.
* final double salesTax=.06;: las convenciones de nombres de variables en Java establecen que los nombres deben comenzar con letra minúscula y utilizar la convención CamelCase para separar las palabras. En este caso, el nombre de variable debería ser salesTax en lugar de SALES\_TAX.
* double gearratio=.05,Gear=4;: los nombres de variables deben utilizar la convención CamelCase para separar las palabras. Además, Gear debería ser gear para seguir la convención de nombres de variables en Java.
* int current\_gear;: las convenciones de nombres de variables en Java establecen que los nombres de variables deben comenzar con letra minúscula y utilizar la convención CamelCase para separar las palabras. En este caso, el nombre de variable debería ser currentGear en lugar de current\_gear.