 



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 nomenclatura | 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. |
| inicializa | 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; |
| literal | Puede ser cualquier número, texto o información que represente un valor; se utiliza para inicializar un tipo primitivo. |
| Declaración | Sentencia Java cuando se define una variable pero no se le asigna necesariamente un valor. Ejemplo: int x; |
| presedencia | 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 datos | Proceso de modificar explícitamente un tipo de dato para que se convierta en otro tipo de dato diferente. |
| Truncamiento | Concepto por el que un número se redondea siempre al entero más cercano. |
| Operador de asignación | Signo igual “=” utilizado en una sentencia Java para asignar un valor a una variable. |
| Conversión cast | 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;

False

True

False

False

False

False

False

True

False

False

True

False

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);

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

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;

R/ 1. Int 2beOrNot2be; : El nombre de la variable empieza con un número, lo cual es inválido en Java. Los nombres de las variables

deben empezar con una letra o un guion bajo.

2. float price index; : El nombre de la variable tiene un espacio en medio, lo cual es inválido en Java. Los nombres de las variables no

pueden contener espacios.

3. double lastYear&#39;sPrice; : El nombre de la variable contiene un apóstrofo, lo cual es inválido en Java. Los nombres de las variables no pueden contener caracteres especiales, excepto el guion bajo.

4. long class; : &quot;class&quot; es una palabra reservada en Java, lo cual significa que no se puede utilizar como nombre de variable.

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;

R// 1. Las variables tienen nombres cortos y poco descriptivos: En la segunda lista, se utilizan nombres cortos como &quot;c&quot;, &quot;s&quot;,&quot;g&quot;, &quot;SALES\_TAX&quot;, &quot;gearratio&quot;, &quot;Gear&quot; y &quot;current\_gear&quot;. Esto puede hacer que el código sea menos legible y comprensible para otros programadores.

2. No se sigue la convención de CamelCase: En la segunda lista, se utilizan nombres de variables en minúsculas y mayúsculas sin seguir la convención de CamelCase. La convención de CamelCase sugiere que los nombres de las variables deben empezar en minúscula y las palabras siguientes empezar en mayúscula, sin usar guiones ni guiones bajos.

3. Uso inconsistente de mayúsculas: En la segunda lista, algunas variables empiezan en mayúscula y otras en minúscula, lo

cual es inconsistente.

4. Nombres de variables que son palabras reservadas en Java: En la segunda lista, se utiliza la palabra reservada &quot;int&quot; como

nombre de variable, lo cual no es válido en Java.