

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 aretmetico	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 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;
	Puede ser cualquier número, texto o información que represente un valor; se utiliza para inicializar un tipo primitivo.
	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.
	Proceso de modificar explícitamente un tipo de dato para que se convierta en otro tipo de dato diferente.
	Concepto por el que un número se redondea siempre al entero más cercano.
	Signo igual "=" utilizado en una sentencia Java para asignar un valor a una variable.
	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.

$$A = \frac{1}{2}bh$$

2. 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. $a = \frac{\sqrt{x^5-6}}{4}$

b. $b = x^y - 6^x$

c. $c = 4\cos\left(\frac{z}{5}\right) - \sin x^2$

d. $d = x^4 - \sqrt{6x - y^3}$

e. $e = \frac{1}{y - \frac{1}{x-2y}}$

f. $f = 7(\cos(\sqrt{5 - \sin\sqrt{3x - 4}}))$

3. 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.
4. 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);
```

```
_____
```

```
true_false=(i>j || true_false && j>=4);
```

```
_____
```

```
true_false=(!(i<2 && j==5));
```

```
_____
```

```
true_false=!true_false;
```

```
_____
```

5. 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;
```

6. 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;
```

