

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



Copyright © 2020 Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

# Vocabulario

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

|  |  |
| --- | --- |
|  | Mecanismos de almacenamiento de datos primitivos u objetos con nombre definidos en un programa. El |
| variable y constante | 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 programación | Convenciones de formato y nomenclatura que siguen las mayorías 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; |
| Valor inicial predeterminado | 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; |
| presidencia | 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. |
| Casting de datos | Proceso de modificar explícitamente un tipo de dato para que se convierta en otro tipo de dato diferente. |
| Redondeo aritmético | Concepto por el que un número se redondea siempre al entero más cercano. |
| Asignación de variables | Signo igual “=” utilizado en una sentencia Java para asignar un valor a una variable. |
| Casting de tipos | 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 |

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; no es válido ya que los nombres de las variables en Java no pueden comenzar con un número.  float price index; no es válido porque el espacio entre "price" y "index" no está permitido en el nombre de la variable. Los espacios no son permitidos en los nombres de variables en Java.  double lastYear’sPrice; no es válido porque el apóstrofo no está permitido en los nombres de variables en Java. Solo se permiten letras, números y el signo de subrayado (\_) en los nombres de variables.  long class; no es válido porque "class" es una palabra reservada en Java y no puede ser utilizada como nombre de una 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 tienen nombres demasiado cortos e inespecíficos, lo que dificulta la comprensión del código.  double salesTax=.06; double La variable "salesTax" debería comenzar con letra minúscula según las convenciones de nomenclatura en Java.  gearratio=.05,Gear=4; La variable "gearratio" debería tener el nombre en minúscula y se recomienda utilizar un guion bajo para separar las palabras en nombres compuestos.  int current\_gear; La variable "Gear" debería comenzar con letra minúscula y su nombre no es descriptivo. No es recomendable utilizar guiones bajos al principio o al final del nombre de una variable, como ocurre con "current\_gear". | |