

Operaciones Aritméticas

Operaciones Aritméticas	1
¿Qué aprenderás?	2
Introducción	2
Motivación	2
Operadores aritméticos	3
Operaciones con variables	3
Creando una calculadora	4
Precedencia de operadores	4
Orden de las operaciones	5
Operaciones y paréntesis	5
Operaciones con números enteros y decimales	6
Float	6
Enteros y float	6
Ejercicio guiado: Calculando la velocidad de un automóvil	7
Ejercicio propuesto (3)	8



¡Comencemos!

¿Qué aprenderás?

- Hacer uso correcto de los operadores aritméticos de tal manera que se puedan implementar en un código Java para realizar cálculos matemáticos simples y complejos.

Introducción

Las operaciones aritméticas no son nada más que la implementación de cálculos matemáticos en lenguaje Java. Esto es como traducir fórmulas matemáticas al lenguaje en sí.

A continuación aprenderemos a construir aplicaciones del tipo de calculadora, donde el usuario ingresa valores y le entregamos resultados.

Motivación

¿Por qué debemos aprender operaciones matemáticas si estamos interesados en crear aplicaciones web o videojuegos?

Los operadores aritméticos se ocupan todo el tiempo ya sea para calcular el total de un carro de compras o cambiar la posición de un personaje en un videojuego.

En Java existen herramientas que nos permiten, entre otras cosas, sumar, restar, asignar valores, comparar, entre otros, gracias a los operadores.

¡Vamos con todo!



Operadores aritméticos

Las operaciones aritméticas nos permiten realizar operaciones matemáticas sobre los números:

Operador	Nombre	Ejemplo	Resultado
+	suma	2+3	5
-	resta	2-3	-1
*	multiplicación	2*4	8
/	división	12/3	4
%	módulo o resto	5/2	1

Tabla 1. Operadores aritméticos.

Fuente: Desafío Latam.

Operaciones con variables

El proceso es exactamente igual si guardamos los valores en variables

```
int a = 2;  
int b = 3;  
System.out.println(a+b);
```

Creando una calculadora

Esto nos permite que el usuario ingrese los valores, y nosotros operemos sobre ellos.

```
Scanner sc = new Scanner(System.in);
int a = sc.nextInt(); //2
int b = sc.nextInt(); //3
System.out.printf("a + b es igual a %d \n", a+b);
System.out.printf("a * b es igual a %d \n", a*b);
//a + b es igual a 5
//a * b es igual a 6
```

Precedencia de operadores

Un concepto muy importante que debemos conocer es el de precedencia, es decir, saber en qué orden se realiza un grupo de operaciones.

En el lenguaje Java y todos los lenguajes, la precedencia sigue las mismas reglas aritméticas de las matemáticas convencionales, es decir, primero los paréntesis con multiplicaciones o divisiones, etc. Por ejemplo:

```
10 - 5*2
10 - 10
0
```

Orden de las operaciones

Veamos una tabla simplificada de precedencia. Esta tabla está ordenada de mayor a menor prioridad, esto quiere decir que la operación de exponenciación tiene mayor precedencia que la suma.

Operador	Nombre
Math.pow(a,b)	Potencia o elevado a..
*, /, %	Multiplicación, división, resto o módulo
+, -	Suma, resta

Tabla 2. Precedencia de operadores.
Fuente: Desafío Latam.

Cuando dos operaciones tienen el mismo nivel de prioridad entonces se resuelven de izquierda a derecha.

Operaciones y paréntesis

Al igual que en matemáticas, los paréntesis cambian el orden en que preceden las operaciones. Dando prioridad a las operaciones que estén dentro de los paréntesis.

```
System.out.println((10 - 5)*2); // 10  
System.out.println(10-5*2); //0
```

¡Los paréntesis sí importan!

Operaciones con números enteros y decimales

Si dividimos números enteros nos encontraremos con una sorpresa.

```
System.out.println(5/3); // = 1
```

Esto es muy común en todos los lenguajes de programación, para obtener la respuesta que esperamos necesitamos ocupar otro tipo de dato, el float.

Float

El tipo de dato primitivo asociado a los números decimales se llamaba float.

Enteros y float

La división entre entero y float, o float y entero da como resultado un float.

```
System.out.println(5.0f/3.0f);
```

O bien podríamos tener dos variables de tipo float.

```
float a = 5.0f;  
float b = 2.0f;  
System.out.println(a/b);
```

La división entre floats también es un float. En Java podemos transformar un entero a float usando cast, esto será especialmente útil cuando estemos trabajando con variables que contengan enteros.

```
int a = 1;  
int b = 2;  
/*a esto se le llama cast, es una transformación de un tipo de datos a otro  
tipo.*/  
System.out.println((float)a/b);
```

Ejercicio guiado: Calculando la velocidad de un automóvil

Calcular la velocidad de un auto en km/h teniendo en cuenta la distancia y el tiempo, cabe destacar que puedes usar decimales para la entrada de datos.

Lo primero que debemos hacer es declarar las variables:

```
float distancia;  
float tiempo;
```

Después solicitamos la entrada de los datos:

```
Scanner sc = new Scanner(System.in);  
  
    System.out.println("Ingresa la distancia en km: ");  
    distancia = sc.nextFloat();  
  
    System.out.println("Ingresa el tiempo en horas: ");  
    tiempo = sc.nextFloat();
```

Creamos la variable para la realización del cálculo y su resultado:

```
// Calcular  
float velocidad = distancia / tiempo
```

Finalmente, mostramos el resultado por pantalla:

```
System.out.printf("La velocidad es: " +velocidad+ " km/h");
```