3. Operadores y expresiones

Todo lenguaje de programación va a requerir del conjunto de operaciones de la vida real que podemos utilizar en las matemáticas o en otras disciplinas para poder transformar esas soluciones analógicas a digitales. Una expresión es una sucesión (sintácticamente correcta) de varios valores, variables u operaciones relacionadas que habrá que ir calculando hasta obtener el resultado definitivo, siendo el tipo de la expresión el tipo de este valor. Así pues, en este apartado vamos a aprender cada uno de estos valores.

3.1. Operadores aritméticos

En Java disponemos de las siguientes operaciones para trabajar con números enteros (sin decimales). Supongamos que op1 vale 5 y que op2 vale 2:

Operación	Símbolo	Expresión	Resultado con números enteros
Suma	+	op1 + op2	7
Resta	-	op1 - op2	3
Multiplicación	*	op1 * op2	10
División entera	/	op1 / op2	2
Resto de la división entera	%	op1 % op2	1

Tabla 2.1. Listado de operaciones aritméticas en Java

3.2. Operadores incrementales

Son operadores unarios (con un solo operando) que permiten incrementar o decrementar las variables en una unidad. Y tienen correspondencia con operaciones en lenguaje máquina que son muy rápidas de ejecutar. Es importante diferenciar entre la notación prefija y sufija. En la prefija, se realiza primero la operación y luego se la compara con el resultado obtenido, mientras que en la sufija el proceso es al revés: primero comparamos con el valor actual y después realizamos la operación que hayamos utilizado de suma o resta incremental.

Operación	Símbolo	Expresión prefija	Expresión sufija	Equivalente
Suma incremental	++	++op1	op1++	op1 = op1 +1
Resta incremental	--	--op1	op1--	op1 = op1 - 1

Tabla 2.2. Listado de operaciones incrementales en Java

En este caso ++op1 incrementa el valor y luego lo signa, mientras que op1++ asigna su valor y luego se autoincrementa.

3.3. Operadores de asignación

El principal operador es el símbolo igual '='. Este almacena el resultado reflejado a su derecha en la variable que se encuentre a su izquierda (como `perimetro=22`).

También hay más operadores de asignación con distintas funciones. Por ejemplo, realizar primero una operación aritmética utilizando el valor actual de la variable que se encuentra a su izquierda —que a la vez será la que guarde el resultado de la operación— y, por tanto, modificando su valor final después de su ejecución. Supongamos que op1 vale 5 y que op2 vale 2.

Operación	Símb.	Expresión	Equivalente	Serie de resultados
Suma y asignación	+=	op1 += op2	op1 = op1 + op2	7, 9, 11, 13, 15
Resta y asignación	-=	op1 -= op2	op1 = op1 - op2	3, 1, -1, -3, -5
Multiplicación y asignación	*=	op1 *= op2	op1 = op1 * op2	10, 20, 40, 80, 160
División y asignación	/=	op1 /= op2	op1 = op1 / op2	2, 1
Resto de la división y la asignación	%=	op1 %= op2	op1 = op1 % op2	1, 0

Tabla 2.3. Listado de operaciones aritméticas con asignación en Java

3.4. Operadores relacionales

Permiten comparar variables según relaciones de igualdad/desigualdad o del tipo mayor/menor. Devuelven siempre un valor booleano. Al utilizar el símbolo de igual '=' para la operación de asignación, vamos a necesitar poner dos símbolos igual juntos para efectuar realmente este tipo de comparaciones de igualdad en este tipo de operaciones relacionales. Supongamos que op1 vale 5 y que op2 vale 2.

Operación	Símbolo	Expresión	Resultado
Mayor que	>	op1 > op2	true
Menor que	<	op1 < op2	false
Iguales	==	op1 == op2	false
Distintos	!=	op1 != op2	true
Mayor o igual que	>=	op1 >= op2	true
Menor o igual que	<=	op1 <= op2	false

Tabla 2.4. Listado de operaciones relacionales en Java

3.5. Operadores lógicos

Nos permiten construir expresiones lógicas con valores booleanos. Nos vamos a centrar en las tres que más se utilizan. Dos son binarias: la 'Y' y la 'O' lógicas. Y otra, unaria: la negación.

Operación	Símbolo	Expresión	Resultado	Observaciones
Y lógica	&&	op1 && op2	Devuelve true si ambos son true.	Operador binario, requiere de dos operandos.
O lógica	||	op1 || op2	Devuelve true si alguno es true.	Operador binario, requiere de dos operandos.
Negación	!	! op1	Niega el operando que se le pasa.	Operador unario, solo requiere de un operando.

Tabla 2.5. Listado de operaciones lógicas en Java

Podemos observar en la tabla de verdad que solo cuando ambos operandos son verdaderos, el operador lógico Y dará como resultado verdadero. Mientras que el operador lógico O, al contrario, solo será falso cuando ambos operandos sean falsos.

op1	op2	op1 && op2	op1 || op2	! op1
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Tabla 2.6. Tabla de verdad de las operaciones lógicas en Java

3.6. Operador de concatenación

Este operador de concatenación, representado por el símbolo más '+', sirve para unir cadenas de caracteres, pudiendo intercalar cualquier otro valor del tipo que sea, que se convertirá en una cadena previamente. Por ejemplo, utilizando el método de mostrar por pantalla una cadena de texto, **suponiendo que el valor de la variable** `resultadoTotal` **sea 15, este código mostrará por pantalla** el texto «**El resultado es 15 unidades**»:

```
System.out.println("El total es "+ resultadoTotal + " unidades.");
```

3.7. Otros operadores

Existen otros tipos de operadores como los de **desplazamiento de bits**, heredados de la sintaxis de C++, que no se suelen utilizar. Los operadores de bits raramente los vas a utilizar en tus aplicaciones de gestión, pero te los comentamos para que sepas que existen.

También existe un operador ternario (requiere de tres operandos). Se trata del **condicional,** que se explica con más detalle en la unidad de uso de estructuras de control.

Su sintaxis es: **expresion ? sentencia1 : sentencia2**

Esto significa que se evalúa el valor de la expresión condicional situado a la izquierda del símbolo '?'. Si es cierta la operación, se ejecutará la sentencia1, y si no lo es, se ejecutará la sentencia2, como veremos más adelante de manera detallada.

3.8. Operadores por orden de precedencia

Por último, debes conocer el orden de precedencia entre los distintos tipos de operadores que acabas de aprender, ya que este alterará el resultado final. En la siguiente tabla se muestran ordenados de mayor a menor prioridad. En las próximas unidades conoceremos más operaciones que puede que cambien un poco el orden a la hora de ejecutarse.

Operadores	Asociatividad	Tipo
( )	De izquierda a derecha ->	Paréntesis
++ -- (tipoCasting)	<- De derecha a izquierda	Unarios y casting de tipos
* / %	De izquierda a derecha ->	Aritméticos multiplicativos
+ -	De izquierda a derecha ->	Aritméticos aditivos
< <= > >=	De izquierda a derecha ->	Relacionales
== !=	De izquierda a derecha ->	De igualdad
&& ||	De izquierda a derecha ->	Lógicos
? :	<- De derecha a izquierda	Condicional ternario
= += -= *= /= %=	<- De derecha a izquierda	Asignación

Tabla 2.7. Tabla de orden de precedencia entre las operaciones en Java