Lección 6: Operadores utilizados en Java

Los operadores son símbolos que se emplean en un programa para operar con los datos del mismo. Por ejemplo:

```
int a=5; //asignación a una variable
int b=10; //asignación a una variable
int c=a+b;  //suma de dos variables y asignación a otra
```

En Java existen muchos operadores, que podemos clasificar en los siguientes grupos en función del tipo de operación que realizan

- Aritméticos
- Asignación
- Condicionales
- Lógicos
- Otros

Operadores aritméticos

Se emplean con tipos numéricos para realizar operaciones aritméticas en un programa. La siguiente tabla te muestra cuales son estos operadores y la operación que realizan.

Operador	Descripción
+	Suma dos valores numéricos.
-	Resta dos valores numéricos.
*	Multiplica dos números.
/	Divide dos números. El tipo de resultado depende de los operandos, pues en el caso de que ambos sean enteros, el resultado de la división siempre será entero.
%	Calcula el resto de la división entre dos números.
++	Incrementa una variable numérica en una unidad y deposita el resultado en la variable.
--	Decrementa una variable en una unidad y deposita el resultado en la variable.

A continuación te presento algunos ejemplos:

```
int a=3, b=8;
int c=b/a; //el resultado es 2, la división entre enteros es siempre entero
int n=a%2; //resto de dividir a entre 2
a++; //equivale a a=a+1;
n--; //equivale a n=n-1;
```

Respecto a los operadores ++ y --, hay que tener en cuenta que pueden ir delante o detrás de la variable. Aunque depende de donde se pongan, pueden afectar de forma diferente:

```
int a=3, b;
b=a++; //b toma el valor 3
```

```
int a=3, b;
b=++a; //b toma el valor 4
```

Fíjate en el primer caso, cuando el operador se coloca después de la variable, **la operación de asignación se realiza antes** del incremento.

Operadores de asignación

Asignan el resultado de la expresión que aparece a la derecha del operador a la variable que se indica a la izquierda del mismo:

variable = expresión

La siguiente tabla resume los operadores de asignación existentes en Java:

Operador	***Descripción***
=	Asigna la expresión de la derecha, a la variable situada a la izquierda del operador.
+=	Suma la expresión de la derecha, a la variable situada a la izquierda del operador y deposita el resultado en la variable
-=	Resta la expresión de la derecha a la variable situada a la izquierda del operador y deposita el resultado en la variable
***=**	Multiplica la expresión de la derecha con la variable y deposita el resultado en la variable.
/=	Divide la variable situada a la izquierda entre la expresión de la derecha, depositando el resultado en la variable.
%=	Calcula el resto de la división entre la variable situada a la izquierda y la expresión de la derecha, depositando el resultado en la variable.

Ejemplos:

```
int a=3, b=2;
a+=10;  //equivale a a=a+10;
b*=2; //equivale a b=b*2;
b%3; //equivale a b=b%3;
```

Operadores condicionales

Evalúan dos operandos y dan como resultado un valor boolean (true o false). Habitualmente se emplean en instrucciones de control de flujo.

En la siguiente tabla te muestro los operadores condicionales empleados en Java. Fíjate en el primero de ellos (==), utilizado para comprobar la igualdad de dos datos, se utiliza un doble igual en lugar del simple:

Operador	***Descripción***
==	Compara dos valores, en caso que sean iguales el resultado de la operación será *true*
<	Si el operando de la izquierda es menor que el de la derecha el resultado es *true*
>	Si el operando de la izquierda es mayor que el de la derecha el resultado es *true*
<=	Si el operando de la izquierda es menor o igual que el de la derecha el resultado es *true*
>=	Si el operando de la izquierda es mayor o igual que el de la derecha el resultado es *true*
!=	Si el valor de los operandos es diferente el resultado es *true*.

Ejemplos:

```
int a=3;
int c=5;
if(a>c) //comprueba si a es mayor que c
if(a==3) //comprueba si a es igual a 3
if(c!=0) //comprueba si c es diferente a 0
```

Operadores lógicos

Evalúan expresiones de tipo boolean, dando como resultado también un boolean. Son tres los operadores lógicos de Java:

Operador	Descripción
&&	Operador AND. El resultado es verdadero únicamente si ambos operando son verdaderos
||	Operador OR. El resultado es verdadero si alguno de los operandos es verdadero
!	Operador NOT. Se aplica sobre un único operando, su resultado es el contrario al valor del operando

A continuación te mostramos algunos ejemplos:

```
int a=3;
int c=9;
int n=0;
if(a>n && a<c) //verdadero
if(a==2 || c>10) //falso
if(!(n==0)) //falso
```

Otros operadores

Además de los que te he presentado, Java dispone de otros operadores que no entran en una categoría específica. Entre los más importantes están:

new. Se utiliza para crear objetos a partir de una clase. Lo veremos en la sección de arrays

instanceof. Se emplea para comprobar si un objeto es de un determinado tipo. De momento, no veremos su uso

Operador ternario (?:). Se utiliza para realizar una operación u otra, en función de una condición:

variable = (condicion)?expresion1:expresion2;

Si la condición es verdadera, se ejecutará la expresión1 y el resultado se almacenará en la variable, pero si la condición es falsa se ejecutará expresión2 y será este resultado el que se guarde en la variable

En el siguiente ejemplo se guarda en la variable div el resultado de dividir el mayor entre el menor de dos números:

```
int a=3;
int b=9;
int div;
div=(a>b)?a/b:b/a;
```

Precedencia de operadores

En Java, salvo las que implican asignación, las operaciones se ejecutan de izquierda a derecha. Sin embargo, hay operaciones que tienen preferencia sobre otras. Por ejemplo, dada la siguiente instrucción:

int a=3+4*5;

¿Cuál sería el valor de la variable a?. La respuesta es 23, pues aunque primero aparece la suma y después la multiplicación, como esta tiene preferencia sobre la primera, se ejecuta antes la operación 4*5 y después, el resultado se suma a 3.

Si quisiéramos realizar primero la suma, podemos recurrir al uso de paréntesis o corchetes:

int a=(3+4)*5; //el resultado es 35

La siguiente tabla nos muestra la preferencia de los operadores Java en orden decreciente, de modo que los operadores de la parte superior tienen más preferencia que los de la parte inferior:

Operador
() []
++ --
* / %
+ -
> >= < <=
== !=
&&
||
?:
= += -= *= /= %=