

Tema 3: Operadores

Antonio J. Sierra

Índice

- Aritméticos
- A nivel de bit
- Relacionales
- Lógicos
- Asignación

Aritméticos

- Se utilizan para operaciones matemáticas, exactamente de la misma manera en la que están definidos en álgebra.
- Los operandos deben ser de tipo numérico.
- No se pueden utilizar estos operandos con tipos **boolean**.
- Se pueden utilizar con operandos de tipo **char**.

Operadores Aritméticos

Operador	Resultado
+	Suma (también el más unario).
-	Resta (también el menos unario).
*	Multipliación
/	División
%	Módulo
++	Incremento
+=	Suma y asignación
-=	Resta y asignación
*=	Multipliación y asignación
/=	División y asignación
%=	Módulo y asignación
--	Decremento

Operadores al nivel de bit

- Se pueden aplicar a los tipos enteros, **long**, **int**, **short**, **char** y **byte**.
- Los enteros son con signo, el bit de mayor peso indica el signo

Operadores al nivel de bit

Operador	Resultado
~	NOT unario a nivel de bit.
&	AND a nivel de bit.
	OR a nivel de bit.
^	OR exclusivo a nivel de bit.
>>	Desplazamiento a la derecha.
>>>	Desplazamiento a la derecha rellenando con ceros.
<<	Desplazamiento a la izquierda.
&=	AND a nivel de bit y asignación.
=	OR a nivel de bit y asignación.
^=	OR exclusivo a nivel de bit y asignación.
>>=	Desplazamiento a la derecha y asignación.
>>>=	Desplazamiento a la derecha rellenando con ceros y asignación.
<<=	Desplazamiento a la izquierda y asignación.

Operadores lógicos al nivel de bit

- Son los siguientes: **&**, **|**, **^** y **~**.

A	B	A B	A&B	A^B	~A
0	0	0	0	0	1
1	0	1	0	1	0
0	1	1	0	1	1
1	1	1	1	0	0

Desplazamiento a la derecha sin signo

```
int a = -1;
```

```
a = a >>>24;
```

Para explicar qué es lo que ocurre, veamos la misma operación en formato binario:

```
11111111 11111111 11111111 11111111
```

-1 en binario como entero

```
>>>24
```

```
00000000 00000000 00000000 11111111
```

255 en binario

como entero

Operadores Relacionales

- Determinan la relación que un operando tiene con otro.
- El resultado de estas operaciones es un valor **boolean**.
- Expresiones que controlan la sentencia **if** y las distintas sentencias de bucles.

Operador	Resultado
==	Igual a
!=	Distinto a
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

Operadores Lógicos booleanos

- Operan únicamente con operandos **boolean**
- Resultado un valor **boolean**

Operadores Lógicos booleanos

Operador	Resultado
&	AND lógico
	OR lógico
^	XOR lógico
	OR en cortocircuito
&&	AND en cortocircuito
!	NOT unario lógico
&=	Asignación AND
=	Asignación OR
^=	Asignación XOR
==	Igual a
!=	Distintio a
?:	if-then-else ternario

Tablas de los operadores Lógicos booleanos

A	B	A B	A&B	A^B	!A
false	false	false	false	false	true
true	false	true	false	true	false
false	true	true	false	true	true
true	true	true	true	false	false

Operadores lógicos en cortocircuito

- Si se utilizan las formas `||` y `&&`, en lugar de las formas `|` y `&`, Java no se molestará en evaluar el operando de la derecha cuando el resultado de la expresión esté ya decidido a partir del operando de la izquierda.

```
if (denom !=0 && num /denom>10)
```

```
...
```

```
if (c==1 & e++ <100) d= 100;
```

El operador de asignación

- El **operador de asignación** es un único signo igual (=).
- En Java, funciona igual que en el resto de lenguajes de programación. Su formato general es:

```
var = expresion;
```

- El tipo de **var** tiene que ser compatible con el tipo de **expresión**.
- El operador de asignación tiene una interesante característica con la que puede no estar familiarizado: **permite crear una cadena de asignaciones**. Por ejemplo, consideremos el siguiente fragmento de código:

```
int x, y, z;  
x = y = z = 100; //asigna a x, y y z el valor de 100
```

El operador ?

- Su formato general es

Expresión1 ? expresión2: expresión3

Precedencia de los operadores

Precedencia más alta	
Operadores postfijo	() [] . expr++ expr--
Operadores unarios	++expr --expr +expr -expr ~
Creación o cast	new (tipo)expr
multiplicativas	* / %
aditivas	+ -
shift	<< >> >>>
relacional	< > <= >= instanceof
igualdad	== !=
Bitwise AND	&
Bitwise OR exclus.	^
Bitwise OR inclus.	
AND lógico	&&
OR lógico	
condicional	?:
asignación	= += -= *= /= %= &= ^= = <<= >>= >>>=
Precedencia más baja	

Asociatividad

- Todos los operadores binarios excepto los operadores de asignación se evalúan de **izquierda a derecha**.
- Los operadores de asignación se evalúan de **derecha a izquierda**.