

Java_연산자(Operator)

목차

1. 연산자와 우선순위
2. 연산자 - 산술, 증감, 관계, 삼항 연산자
3. 연산자 - 논리, 비트, 시프트, 대입 연산자

1. 연산자와 우선순위

[1] 연산자(operator)

- 연산자(operator)란 대상을 두고 계산을 할 때 사용되는 문자를 말하며 계산이 이루어지는 대상을 피연산자(operand)라고 한다. 연산자와 피연산자는 주어진 식을 계산하여 결과를 나타내는 연산을 하는 구성요소이다.

1. 연산자와 우선순위

- 자바에는 피연산자가 1개인 단항 연산자와 피연산자가 2개인 이항연산자, 피연산자가 3개인 삼항 연산자 등을 제공한다.

①**단항 연산자(unary operator)**는 항을 하나만 사용하는 연산자를 말하며 데이터를 가진 변수에게만 적용되는 연산자이다. 단항연산자의 종류로는 부호 연산자, 증감, 증가 연산자 등이 있다.

②**이항연산자(binary operator)**는 두 개의 피연산자와 한 개의 연산자를 가지는 형식을 말하며 종류로는 산술 연산자, 비트, 논리 연산자 등이 있다.

③**삼항연산자(Ternary operator)**는 한 개의 조건식과 두 개의 연산자를 가지는 구조로 제공되는 연산자로 세 개의 피 연산자로 구성되며 연산자로 조건에 따라 서로 다른 내용을 수행하는 경우에 사용 된다

1. 연산자와 우선순위

- 다음은 연산자와 피연산자와의 식(expression)을 표시한 것이다

$$x = a + -b * c$$

a, b, c, x 는 연산의 대상인 데이터를 가진 피연산자이며 **=, +, -, *** 는 피연산자를 계산하는 연산자이다.

1.연산자와 우선순위

- [2]연산자(operator) 우선 순위

연산자의 우선순위란 ?

자바의 연산식은 하나의 계산식에 연산자들이 섞여서 우선 순위를 가지며 계산된다. 우선순위라는 것은 어떤 연산을 먼저 처리하고, 어떤 연산을 나중에 처리할 것인가에 대한 기준을 의미한다.

1. 연산자와 우선순위

- 다음은 연산자의 종류와 연산할 때의 우선 순위를 표시한 것이다.

우선순위	명칭	연산자	결합성
1	최우선 연산자	. [] ()	→
2	증감 연산자	++ -- ! ~	←
3	승법 연산자	* / %	→
4	가법 연산자	+ -	→
5	Shift 연산자	<< >>	→
6	관계 연산자	< > <= >=	→
7	등가 연산자	== !=	→
8	bit곱 연산자	&	→
9	bit차 연산자	^	→
10	bit합 연산자		→
11	논리곱 연산자	&&	→
12	논리합 연산자		→
13	조건(삼항) 연산자	? :	←
14	대입연산자	= += -= *= /= %= &= ^= = >>= <<=	←
15	coma연산자	,	

1. 연산자와 우선순위

- 단항, 삼항, 대입연산자들을 제외한 연산자들은 우선순위가 동일한 것끼리 연산이 될 때는 먼저 기재된 연산자가 먼저 계산하면 기재된 순서대로 계산되어 진다.
- 다음 수식을 통해 연산되는 우선순위를 살펴 보자.

```
int a=2, b=4, c=3;  
x = a + -b * c;
```

- ① $-b \Rightarrow$ $-$ 연산자는 1 순위를 가지며 주어진 상수의 $-$ 값이 주어져 -4 값이 세팅 된다.
- ② $-b * \Rightarrow$ $*$ 연산자는 2 순위를 가지며 $-4 * 3$ 이 연산된다.
- ③ $a + -b * c \Rightarrow$ $+$ 연산자는 3 순위 가지며 $2 + -12$ 값이 연산된다.
- ④ $x = a + -b * c \Rightarrow$ $=$ 연산자는 14 순위를 가지며 결과 값이 x 에 대입 된다.

2. 연산자 - 산술,증감,관계,삼항 연산자

[1] 산술연산자 : 산술연산자는 연산자 중 가장 많이 사용되는 연산자로 덧셈 , 뺄셈 등이 해당되며 종류로는 +, -, *, /, %, = 가 있다.

다음은 산술 연산자의 종류와 의미를 표현한 것이다.

산술연산자	표현식	연산자의 의미	예제	결과
=	$A = B$	B의 값을 A에다가 대입 한다	$A=10$	10
+	$C = A + B$	A와 B를 더한 값을 C에 대입 한다	$A=10+2$	12
-	$C = A - B$	A에 있는 값에서 B에 있는 값을 뺀 다음 C에 대입 한다	$A=10-2$	8
*	$C = A * B$	A와 B를 곱한 값을 C에 대입 한다	$A=10*2$	20
/	$C = A / B$	A에 있는 값에서 B에 있는 값을 곱한 다음 C에 대입 한다	$A=10/2$	5
%	$C = A \% B$	A에 있는 값에서 B에 있는 값을 나눈 다음 나머지를 C에 대입 한다	$A=10\%2$	0

2. 연산자 - 산술,증감,관계,삼항 연산자

[2] 증가(++), 증감(--), 연산자

- 증가(++), 증감(--), 연산자는 피연산자의 값을 1씩 증가 또는 1씩 감소 시키는 연산자이다
- 변수나 상수 앞에 선언될 때와 뒤에 붙일 때의 결과 값이 달라져 연산자의 위치에 따라 전위 연산자, 후위 연산자라고 부른다.

전위 => ++a; 또는 --a

후위 => a++ ; 또는 a--

- 전위연산자는 증가, 증감 연산자를 피연산자의 앞에 선언이 되어 피연산자의 값을 변경한 다음 피연산자의 값을 연산에 리턴 하게 되며 피연산자의 뒤에 붙게 되는 후위 연산자는 연산자의 값을 먼저 리턴하고 값을 변경하게 된다.

2. 연산자 - 산술,증감,관계,삼항 연산자

종류로는 ++과 -- 두 종류가 있다.

- ++ : 1 증가 연산자 (Increment)로 변수의 값을 1 증가 시킴
- -- : 1 증감 연산자 (Decrement)로 변수의 값을 1 감소 시킴

2. 연산자 - 산술,증감,관계,삼항 연산자

다음은 증감과 증가 연산자의 종류와 의미를 표현한 것이다.

증감 연산자	표현식	연산자의 의미	x의 초기값	y 결과값	x 결과값
++	$y = ++x;$	x값에 1을 증가한 값을 y에 대입	5	6	6
	$y = x++;$	x값을 y에 대입한 후 x값을 1 증가	5	5	6
--	$y = --x;$	x값에 1을 증감한 값을 y에 대입	5	4	4
	$y = x--;$	x값을 y에 대입한 후 x 값을 1 감소	5	5	4

2. 연산자 - 산술,증감,관계,삼항 연산자

[3] 관계 연산자

- 관계 연산자는 비교 연산자라고도 하며 두 개의 값을 비교하여 그 결과를 리턴 하는 연산자이다

관계 연산자의 특징

- ① 관계 연산자는 주로 두수를 비교해서 수가 큰지 작은지, 같은지 다른지를 판단하는 데 사용된다.
- ② 연산의 결과값이 숫자가 아니라 'true', 'false'로 결과 값을 리턴 하기 때문에 논리 상수 또는 부울 상수라고도 한다.
- ③ 부울 상수는 논리 연산자에서도 사용된다.

2. 연산자 - 산술,증감,관계,삼항 연산자

다음은 관계 연산자의 종류와 의미를 표현한 것이다.

관계 연산자	표현식	연산자의 의미	예제	결과
<	$A < B$	A는 B보다 작으면 true 아니면 false	$5 < 7$	true
>	$A > B$	A는 B보다 크면 true 아니면 false	$5 > 7$	false
<=	$A \leq B$	A는 B보다 작거나 같으면 true 아니면 false	$3 \leq 5$	true
>=	$A \geq B$	A는 B보다 크면 true 아니면 false	$3 \geq 5$	false
==	$A == B$	A는 B와 같으면 true 아니면 false	$3 == 5$	false
!=	$A != B$	A는 B와 같지 않으면 true 아니면 false	$3 != 5$	true

2. 연산자 - 산술,증감,관계,삼항 연산자

[4] 삼항연산자

- 삼항(ternary) 연산자는 세 개의 피연산자로 구성된 연산자로 조건에 따라 서로 다른 내용을 수행하는 경우에 사용되는 연산자이다
- 또한 자바의 구문 중 if~else문을 축약해서 사용 할 수 있는 연산자로 관계연산자의 결과인 true, false에 따라 명령을 간단하게 실행 시킬 수 있다.

2. 연산자 - 산술,증감,관계,삼항 연산자

형식은 다음과 같다.

(조건식) ? 식 1 : 식 2

- 조건식이 참(true)이면 식1, 거짓(false)이면 식2를 수행한다

2. 연산자 - 산술,증감,관계,삼항 연산자

다음은 삼항 연산자의 예문이다

```
int num, assign;  
num = 10;  
assign = (num > 5) ? 15 : 20;
```

정수형 변수 num에 10을 대입하고 나서 num이 5보다 크면 15를 assign에 대입하고 그렇지 않으면 20을 대입한다. 비교 연산자에 의해 true를 리턴하기 때문에 assign 변수에는 15가 대입 된다.

3. 연산자 - 논리,비트,시프트,대입 연산자

[1]논리연산자

논리연산자는 AND, OR, NOT 을 이용하여 각 논리값들 사이에 사용되는 연산자로 결과값을 true 나 false 로 받을 수 있는 연산자이다.

논리 연산자의 특징

- ① AND 연산은 '&' 기호가 두 개로 선언된 '&&' 로 표시되며 좌우로 두 개의 값을 비교했을때 좌측 연산의 결과가 'false' 이면 우측 연산은 연산 없이 결과를 'false'로 리턴 한다.
- ② '|' 기호가 두 개로 선언된 '||' 로 표시되는 OR 연산자는 좌측 연산의 결과가 ' true'이면 우측 연산은 연산 없이 결과가 'true'가 되어 리턴 한다.
- ③ NOT 연산은 '!' 기호를 사용하며 결과값이 'true'이면 'false'를 리턴하고 'false'이면 'true'를 리턴 한다.

3. 연산자 - 논리,비트,시프트,대입 연산자

다음은 논리 연산자의 기호와 의미이다

논리 연산자	연산자의 의미
&&	두값이 모두 참일때 만 결과값이 'true'
	두 값이 모두 거짓일 때만 결과값이 'false'
!	결과값이 참이면 'false' , 거짓이면 'true'로 반대로 리턴

3. 연산자 - 논리,비트,시프트,대입 연산자

논리 연산자를 이용한 AND 연산의 결과는 다음과 같다.

AND 표현식	예제	결과값
true && true	(5<7) && (2==2)	true
true && false	(5<7) && (2==3)	false
false && true	(5>7) && (2==2)	false
false && false	(5>7) && (2==3)	false

논리 연산자를 이용한 OR 연산의 결과는 다음과 같다.

OR 표현식	예제	결과값
true && true	(5<7) (2==2)	true
true && false	(5<7) (2==3)	true
false && true	(5>7) (2==2)	true
false && false	(5>7) (2==3)	false

3. 연산자 - 논리,비트,시프트,대입 연산자

논리 연산자를 이용한 NOT 연산의 결과는 다음과 같다.

NOT 표현식	예제	결과값
! true	! (5<7)	false
! false	! (5<7)	true

3. 연산자 - 논리,비트,시프트,대입 연산자

다음 논리 연산식의 예를 살펴 보자.

```
int a=10, b=4;  
System.out.println( (a > b++) || ( a< b++)) ; => true
```

위 코드를 실행 하게 되면 ' a > b++ ' 이 먼저 연산이 되며 '10 > 4' 가 수행 되어 결과는 'true'를 리턴 하고 b++ 후위 연산이 되어 a=10 , b= 5의 값이 저장된다.

먼저 연산된 결과 값이 'true' 이기 때문에 다음 연산인 '(a < b++)' 이 수행되지 않고 'true'를 리턴 하게 된다.

3. 연산자 - 논리,비트,시프트,대입 연산자

[2] 비트 연산자

- 피연산자의 각 비트들을 이용해서 연산이 이루어 지는 비트 연산자는 ' \sim , $\&$, $|$, \wedge ' 등의 4가지 연산자가 제공 된다.

3. 연산자 - 논리,비트,시프트,대입 연산자

비트연산자 의 종류와 특징

- ① ' ~ ' 연산자는 complement(보수) 비트 반전 연산자로 피연산자를 2진수값으로 바꾸어 2진수의 각 비트를 0인 것은 1로, 1인 것은 0으로 모두 반전시킨다.
- ② ' & ' 연산자는 값A와 값B의 각 비트들을 AND로 연산을 하며 두 비트가 모두 진리값 1일 때만 1이 되며 나머지는 0 으로 연산 한다.
- ③ ' | ' 연산자는 값A와 값B의 각 비트들을 OR로 연산을 하며 두 비트가 모두 진리값 0일 때만 0이 되며 나머지는 1로 연산 한다.
- ④ ' ^ ' 연산자는 값A와 값B의 각 비트들을 XOR 로 연산하며 두 비트가 서로 다르면 진리값 1, 같으면 진리값 0으로 연산 한다.

3. 연산자 - 논리,비트,시프트,대입 연산자

다음은 비트 연산자의 종류이다

비트 연산자	표현식	연산자의 의미
&	$A \& B$	A와 B값을 비트 단위로 AND 연산
	$A B$	A와 B값을 비트 단위로 OR 연산
^	$A \wedge B$	A와 B값을 비트 단위로 배타적 OR연산
~	$\sim A$	A의 값을 보수 연산

3. 연산자 - 논리,비트,시프트,대입 연산자

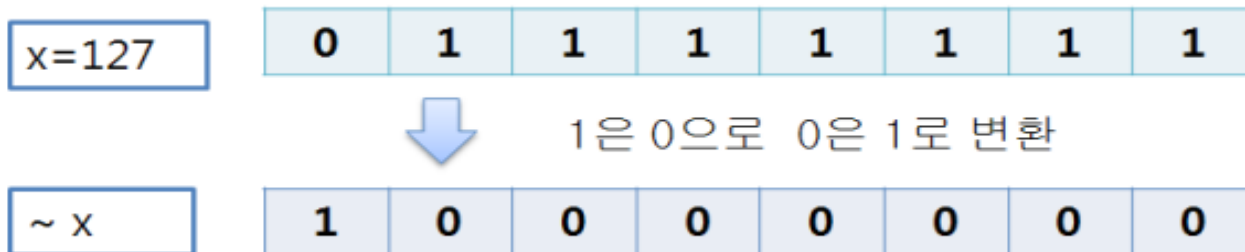
- 다음은 비트 연산자의 기능이다.

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

3. 연산자 - 논리,비트,시프트,대입 연산자

- 비트 연산자 중 ~(보수)를 구하는 연산을 byte 데이터 형과 short 데이터 형을 이용해서 풀이한 내용을 살펴보자
- byte 데이터 형을 이용한 ~(보수)연산의 예이다

```
byte x = 127 ;  
byte y = ~x ;
```



3. 연산자 - 논리,비트,시프트,대입 연산자

- short 데이터 형을 이용한 ~ (보수)연산의 예이다

```
short x = 0x12 ;
```

```
short y = ~x ;
```

x = 0x12

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



1은 0으로 0은 1로 변환

~ x

1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

3. 연산자 - 논리,비트,시프트,대입 연산자

다음은 각 비트들을 이용해서 AND, OR, XOR로 연산한 결과 이다.

	AND	OR	XOR
A = 51	00110011	00110011	00110011
B = 240	& 11110000	 11110000	^ 11110000
	<hr/> 00110000	<hr/> 11110011	<hr/> 11000011

각 비트 값을 변수 A는 51로 이진수는 '**0011 0011**', 변수 B의 값은 240으로 이진수는 '**1111 0000**' 이다.

이 두 변수의 값을 AND로 연산 하면 '**0011 0000**'으로 48의 결과 값을 리턴하고 , OR로 연산하게 되면 '**1111 0011**'로 243의 결과 값을 XOR로 연산하면 '**1100 0011**'로 195 로 결과 값을 리턴 하게 된다.

3. 연산자 - 논리,비트,시프트,대입 연산자

[3] 시프트 연산자 (Shift Operator)

시프트 연산자는 피연산자의 비트들을 주어진 비트수와 연산자에 특성에 따라 비트를 이동시키는 연산자이다.

3. 연산자 - 논리,비트,시프트,대입 연산자

시프트 연산자의 특징은 다음과 같다.

- ① 시프트 연산자는 int, byte, short, long, char 형에 대해서만 정의가 되어 사용 할 수 있으며 float 또는 double, boolean 에 대한 시프트 연산은 사용할 수 없다.
- ② << 연산자는 왼쪽으로 비트를 이동시키고 오른쪽으로는 항상 0을 채운다.
- ③ >> 연산자는 오른쪽으로 비트를 이동시키고 왼쪽 비트는 양수 일때 0, 음수 일 때는 1로 채운다.
- ④ >>>은 부호를 고려하지 않은 shift 연산을 수행 한다. 비트를 오른쪽으로 이동 시키면서 왼쪽에는 부호와 상관없이 0으로 채운다.
- ⑤ short나 byte형에 대해서 >>> 연산을 하면 int형으로 type conversion(promotion)된 후 연산 된다.

3. 연산자 - 논리,비트,시프트,대입 연산자

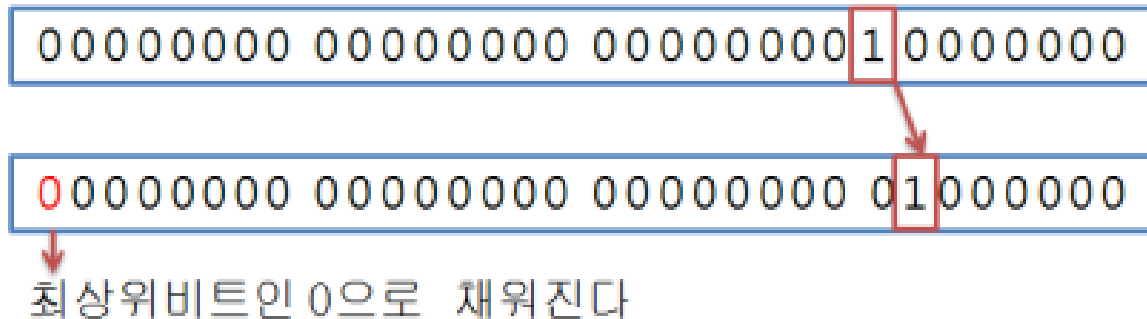
다음은 시프트 연산자의 종류와 연산자의 의미이다

연산자	표현식	연산자의 의미	예제	결과
<<	$x \ll n$	정수 x 를 n 비트 왼쪽으로 이동시키고, 오른쪽 비트는 항상 0으로 채운다	$1 \ll 2$	4
			$-1 \ll 2$	-4
>>	$x \gg n$	정수 x 를 n 비트 오른쪽으로 이동시키고, 왼쪽 비트는 양수일 경우 0으로, 음수일 경우 1로 채운다	$8 \gg 2$	2
			$-8 \gg 2$	-2
>>>	$x \ggg n$	정수 x 를 n 비트 오른쪽으로 이동시키고, 왼쪽 비트는 항상 0으로 채운다	$8 \ggg 2$	2

3. 연산자 - 논리,비트,시프트,대입 연산자

- 다음과 같이 연산자 $>>$, $<<$ 은 부호를 고려한 shift 연산을 수행한 경우이다.
정수인 양수 128을 1비트 오른쪽으로 이동시켜 왼쪽 비트를 0으로 채운 후에
결과를 보면 비트연산이 이루어져 64의 결과 값을 리턴 하는 것을 볼 수 있다.

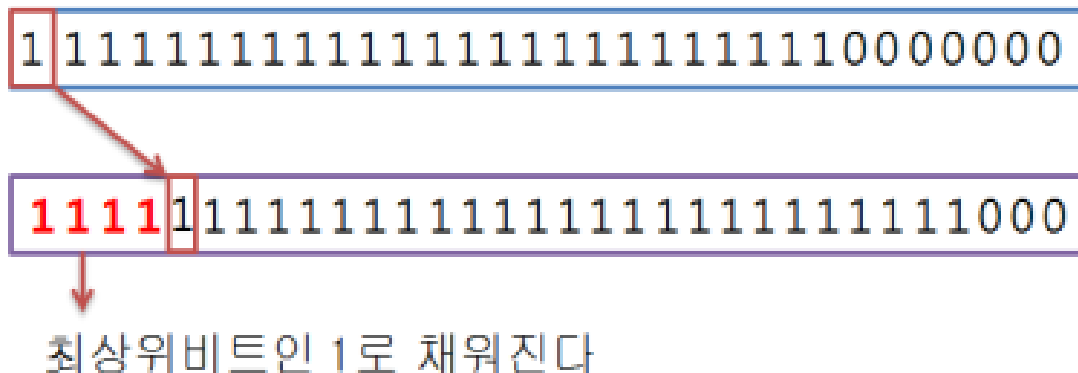
128 $>>$ 1 의 결과 $128/2^1 = 64$



3. 연산자 - 논리,비트,시프트,대입 연산자

- 정수인 음수 -128을 4 비트 오른쪽으로 이동시키고 왼쪽 비트를 1로 채운 후에 결과를 보면 비트연산이 이루어져 -8을 리턴하는 것을 볼 수 있다.

-128 >> 4 의 결과 $-128/2^4 = -8$



3. 연산자 - 논리,비트,시프트,대입 연산자

[5] 대입연산자

대입 연산자는 변수에 값을 저장하기 위한 용도로 오른쪽 수식을 계산한 값을 왼쪽에 있는 변수에 대입하는 연산자이다.

대입 연산자의 특징은 다음과 같다.

- ① 대입연산자에 의해 변수 값이 바뀌면 이전에 해당 변수에 저장 되어 있던 값은 없어진다.
- ② 대입 연산자는 수식을 계산한 후 대입하는 오른쪽 값과 왼쪽 변수의 자료형은 같은 자료형 이어야 한다
- ③ 대입연산자의 수식을 계산한 후 대입하는 오른쪽 값과 왼쪽 변수의 자료형이 서로 다른 자료형인 경우는 왼쪽 변수의 값을 기준으로 type conversion(promotion)이 일어난다.

3. 연산자 - 논리,비트,시프트,대입 연산자

- 다음은 대입 연산자의 종류와 연산자의 의미이다

대입연산자	표현식	연산자의 의미
=	$A = B$	B 의 값을 A에 대입
+=	$A += B$	$A = A + B$ 와 동일
-=	$A -= B$	$A = A - B$ 와 동일
*=	$A *= B$	$A = A * B$ 와 동일
/=	$A /= B$	$A = A / B$ 와 동일
%=	$A \% = B$	$A = A \% B$ 와 동일
&=	$A \& = B$	$A = A \& B$ 와 동일
=	$A = B$	$A = A B$ 와 동일
^=	$A \wedge = B$	$A = A \wedge B$ 와 동일
<<=	$A \ll = B$	$A = A \ll B$ 와 동일
>>=	$A \gg = B$	$A = A \gg B$ 와 동일
>>>=	$A \ggg = B$	$A = A \ggg B$ 와 동일