

비트 연산자

비트(bit)와 바이트(byte)

- 비트는 컴퓨터가 처리하는 정보의 최소 단위인데, 한 개만으로는 많은 양의 데이터를 나타내기에 턱없이 부족하기 때문에 정보를 표현하는 기본단위로는 8개의 비트를 묶은 바이트(Byte)를 사용한다.
- 1byte는 00000000 부터 11111111 까지 값을 표현할 수 있다.
- 1byte는 정수로 표현하면 0부터 254까지 표현가능하다.
- 1byte를 16진수로 표현하면 00 부터 FF 까지 표현가능하다. 4비트는 0부터 15까지 표현가능하기 때문이다.

비트 연산자는 논리 연산자와 비슷하지만, 비트(bit) 단위로 논리 연산을 할 때 사용하는 연산자이다.

- 비트 연산자의 종류로는 $\&$, $|$, \wedge , \sim , \ll , \gg , \ggg 이 있다.
- $\&$ 는 and, $|$ 는 or, \wedge 는 xor, \sim 는 not을 의미한다.
- \ll 는 좌측 시프트(shift), \gg 는 우측 시프트(shift), \ggg 는 우측 양수화 시프트라고 한다.
- $\&$, $|$, \wedge 는 논리연산자로도 사용된다.

and, or, xor, not 비트 연산 결과

AND

대응되는 비트가 1로 같으면 1

A	B	A & B
1	1	1
1	0	0
0	1	0
0	0	0

OR

대응되는 비트 중에서 하나라도 1이면 1

A	B	A B
1	1	1
1	0	1
0	1	1
0	0	0

XOR

대응되는 비트가 서로 다르면 1

A	B	A ^ B
1	1	0
1	0	1
0	1	1
0	0	0

NOT

비트를 반전함

A	~A
1	0
0	1

비트연산자는 바이트를 구성하고 있는 bit를 연산하는 연산자이다.

```
  00000000
& 11111111
-----
  00000000
```

- 두개의 바이트에 각각 00000000 과 11111111 이 저장되어 있을 때 2개의 바이트에 대한 and(&)연산의 결과는? 각각의 자리수에 해당하는 비트끼리 and연산을 한 결과가 구해진다.

>> 와 <<

- << 는 명시된 수만큼 비트들을 전부 왼쪽으로 이동시킨다.
- >> 는 부호를 유지하면서 지정한 수만큼 비트를 전부 오른쪽으로 이동시킨다.
 - 정수형 타입을 비트로 표현했을 때, 맨 좌측의 비트를 부호화 비트라고 한다. 맨 좌측의 비트가 1이면 음수, 0이면 양수를 나타낸다.

예제 shift연산자

```
public class BitOperatorExam1 {  
    public static void main(String[] args){  
        int a = 4;  
        int b = a >> 1;  
        System.out.println(b);  
  
        int c = 4;  
        int d = c << 1;  
        System.out.println(d);  
    }  
}
```

왜 결과가 저렇게 나올까?

```
00000000 00000000 00000000 00000100    (int 타입 4를 bit로 표현)
>> 1 을 하게 되면 우측으로 비트가 하나씩 밀리고, 맨 좌측엔 0이 채워진다.
00000000 00000000 00000000 00000010
```

```
00000000 00000000 00000000 00000100    (int 타입 4를 bit로 표현)
<< 1 을 하게 되면 좌측으로 비트가 하나씩 밀리고, 맨 좌측 비트는 사라지고 우측엔 0이 채워진다.
00000000 00000000 00000000 00001000
```

- 시프트하는 숫자의 값을 n 이라고 한다면, 좌측시프트는 2^n 으로 나눈 결과가, 우측시프트는 2^{-n} 으로 곱한 결과가 나온다.

>>> 는 지정한 수만큼 비트를 전부 오른쪽으로 이동시키며, 새로운 비트는 전부 0이 된다.

- >>> 는 그 결과 무조건 양수가 나온다.
- 10000000 >>> 2 를 하게 되면 00100000 가 된다.
- 10000000 는 음수, 00100000 는 양수이다.

생각해볼 문제

- 1byte에 10000000 이 저장되어 있다. 정수 값 얼마인가?
- 정수가 어떻게 저장되는지 알아야 한다. 너무 어려우면 건너뛰는 것도 공부 방법이다.

감사합니다.