

## 비트 연산자 AND 개념

```
// & 이 비트연산자 AND 관계 연산자에서는 && 형태
// & 이 비트연산자 AND
// 관계 연산자에서는 && 형태로 존재하였음
// 10 ==> 1010
// 8 ==> 1000 AND
// 계산-----
// 8 ==> 1000
// 다음과 같이 AND 연산자는 같은 1이 존재하지 않는다면 출력되지 않는다.
```

```
// int num1 = 10, num2 = 8;
```

```
int num1 = 10, num2 = 8;
```

```
System.out.printf("%d AND %d = %d\n", num1, num2, num1 & num2);
```

```
// 위의 수식과 같이 결과는 같은 8이 출력된다.
```

출력값

10 AND 8 = 8

## 비트 연산자 OR 개념

```
// | 이 비트연산자 OR
// 관계 연산자에서는 || 형태로 존재하였음
// 10 ==> 1010
// 5 ==> 0101 OR
// -----
// 15 ==> 1111
// 다음과 같이 OR 연산자는 한쪽비트에 1만 있으면 그대로 출력된다.
```

```
System.out.printf("%d OR %d = %d\n", num1, num2, num1 | num2);
```

```
// 위의 수식과 같이 결과는 같은 10이 출력된다.
```

출력값

10 OR 8 = 10

결론 : OR 연산은 합집합 개념, AND 연산은 교집합 개념으로 생각하면 편하다.

## 쉬프트 연산자 예제

```
//=====
int num1 = 2, num2 = 5, num3 = 10;

//예제 1
System.out.printf("%d << %d = %d\n", num1, num3, num1 << num3);
//예제 2
System.out.printf("%d << %d = %d\n", num2, num3, num2 << num3);
//예제 3
System.out.printf("%d << %d = %d\n", num3, num3, num3 << num3);
```

출력값

2 << 10 = 2048

5 << 10 = 5120

10 << 10 = 10240

## 쉬프트 연산자 개념

```
// 왼쪽 쉬프트의 경우 단순히 2^n을 곱하면 되지만
// 오른쪽 쉬프트의 경우 단순히 2^n을 나누면 안된다.
// 5 / 2^2 = 1.25
// 결론: 오른쪽 쉬프트는 2^n으로 나누되 소수점을 버려야 한다.
System.out.printf("%d >> %d = %f\n", num2, num1, (float)(num2 >> num1));

// 이유:
// 0101 ----> 5
// 0001 ----> 1
```

출력값

5 >> 2 = 1.000000

**결론** : 쉬프트 연산자는 X << N 쉬프트의 위치와 N값에 따라 2^N을 곱하거나 나눈다. 하지만 정수형만 가능 소수점 버림 주의

## 비트 연산 복합 문제 : 중복값 제거

```
3 public static void main(String[] args) throws InterruptedException {  
4     고정값  
5     final int BIN = 1;  
6     변수지정  
7     int testBit = 0;  
8     int randNum;  
9     10회 반복 for (int i = 0; i < 10; i++) {  
10    랜덤값 출력 randNum = (int)(Math.random()*10);  
11  
12    변수 제거 while ((testBit & (BIN << randNum)) != 0) {  
13        randNum = (int) (Math.random() * 10);  
14    }  
15  
16    System.out.printf("randNum = %d\n", randNum);  
17  
18    변수 제거 testBit |= (BIN << randNum);  
19  
20 }  
21 Thread.sleep( millis: 500);  
22 }  
23  
24 }
```

### 출력값

```
randNum = 1  
randNum = 8  
randNum = 3  
randNum = 4  
randNum = 6  
randNum = 9  
randNum = 0  
randNum = 7  
randNum = 2  
randNum = 5
```

랜덤 출력값이라 출력이 계속 다르다.

1. 10회 랜덤값을 출력한다.
2. 변수 제거 while문 내부
  - 1.1 ) BIN=1 값이 랜덤 출력 값을 통해 쉬프트 연산을 한다.
  - 1.2) testBit와 는 &연산이기때문에 교집합이 없어 0이 발생
  - 1.3) 중복값이 있다면 내부 연산 실행하여 0이 나올때까지 반복
3. while 문을 빠져나와 출력하고 testBit는 OR 연산을 통해 출력값의 비트가 입력된다.
4. 최종적으로 bit는  $2^9$ 까지 차게된다. 정확하게는  $1024 - 1$

//  $2^9$   $2^8$   $2^7$   $2^6$   $2^5$   $2^4$   $2^3$   $2^2$   $2^1$   $2^0$

## Quiz 문제

```
1 ▶ public class Practice {
2 ▶     public static void main(String[] args) throws InterruptedException {
3         final int BIN = 1;
4
5         int testBit = 0;
6         int randNum;
7         for (int i = 0 ; i < 6 ; i++) {
8             randNum = (int)(Math.random()*6 + 5);
9
10            while ((testBit & (BIN << randNum)) != 0) {
11                randNum = (int) (Math.random()*6 + 5);
12            }
13
14            System.out.printf("randNum = %d\n", randNum);
15
16            if (randNum == 7 || randNum == 8 || randNum == 9 || randNum == 10) {
17                System.out.println("randNum2 = " + randNum);
18            }
19
20            testBit |= (BIN << randNum);
21
22        } Thread.sleep( millis: 500);
23        //제가 이해한게 맞는지 모르겠습니다. 출력 값은 이렇게나옵니다.
24    }
25
26 }
```

### 출력값

```
randNum = 7
randNum2 = 7
randNum = 8
randNum2 = 8
randNum = 6
randNum = 5
randNum = 9
randNum2 = 9
randNum = 10
randNum2 = 10
```