



5월13일 복습 & 퀴즈 & 질문

이태양



복습 : BitAndTest

```
public class BitAndTest {  
    public static void main(String[] args) {  
        int num1 = 10, num2 = 8;  
  
        // 10 ==> 1010  
        // 8 ==> 1000  
        // -----  
        //          1000  
        System.out.printf("%d AND %d = %d\n", num1, num2, num1 & num2);  
        //&연산자는 겹치는 자리수가 둘다 1 이여야만 1 이다!  
  
        num2 = 138;  
  
        // 138 ==> 10001010  
        // 10 ==>      1010  
        // -----  
        //          00001010  
        System.out.printf("%d AND %d = %d\n", num1, num2, num1 & num2);  
    }  
}
```

10 AND 8 = 8

10 AND 138 = 10



복습 : BitOrTest

```
public class BitOrTest {  
    public static void main(String[] args) {  
        int num1 = 10, num2 = 5;  
  
        // or 연산자는 둘 중 하나만 1이면 1이된다.  
        // 10 ==> 1010  
        // 5 ==> 0101 OR  
        // -----  
        // 15 ==> 1111  
        System.out.printf("%d OR %d = %d\n", num1, num2, num1 | num2);  
  
        num2 = 136;  
  
        // 10 ==> 00001010  
        // 136 ==> 10001000 OR  
        // -----  
        // 138 ==> 10001010  
        System.out.printf("%d OR %d = %d\n", num1, num2, num1 | num2);  
    }  
}
```

10 OR 5 = 15

10 OR 136 = 138



복습 :BitShiftTest

```
public class BitShiftTest {  
    public static void main(String[] args) {  
        int num1 = 2, num2 = 5, num3 = 10;  
  
        // 2^1 x 2^5 = 2^6(64)  
        System.out.printf("%d << %d = %d\n", num1, num2, num1 << num2);  
        // 5 x 2^5 = 160  
        System.out.printf("%d << %d = %d\n", num2, num2, num2 << num2);  
        // 10 x 2^5 = 320  
        System.out.printf("%d << %d = %d\n", num3, num2, num3 << num2);  
  
        // 2^1 x 2^2 = 2^3(8)  
        System.out.printf("%d << %d = %d\n", num1, num1, num1 << num1);  
        // 5 x 2^2 = 20  
        System.out.printf("%d << %d = %d\n", num2, num1, num2 << num1);  
        // 10 x 2^2 = 40  
        System.out.printf("%d << %d = %d\n", num3, num1, num3 << num1);  
  
        // 2^1 x 2^10 = 2^11(2048)  
        System.out.printf("%d << %d = %d\n", num1, num3, num1 << num3);  
        // 5 x 2^10 = 5120  
        System.out.printf("%d << %d = %d\n", num2, num3, num2 << num3);  
        // 10 x 2^10 = 10240  
        System.out.printf("%d << %d = %d\n", num3, num3, num3 << num3);  
  
        // 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. *****기억하기  
        System.out.printf("%d >> %d = %f\n", num2, num1, (float)(num2 >> num1));  
    }  
}
```

2 << 5 = 64
5 << 5 = 160
10 << 5 = 320
2 << 2 = 8
5 << 2 = 20
10 << 2 = 40
2 << 10 = 2048
5 << 10 = 5120
10 << 10 = 10240
5 >> 2 = 1.000000



복습 :InterruptComment

```
public class InterruptComment {  
    public static void main(String[] args) throws InterruptedException {  
        for (int i = 0; ; i++) {  
            if (i % 2 == 0) {  
                System.out.println("하이 난 짝수");  
            } else {  
                System.out.println("하이 난 홀수");  
            }  
            Thread.sleep(500); /// 인터럽션 설명을 위해 만든 간단한 예제 스레드 실행되는거 구경,,  
        }  
    }  
}
```



퀴즈 : 18~21

//18번: 십진수 21을 이진수로 10101

//19번: 십진수 73을 이진수로 1001001

//20번: $2^0 + 2^3 + 2^5 + 2^6 = 1101001(2) = 1+8+32+64 = 105(10)$

//21번: $2^0 + 2^3 + 2^6 + 2^7 + 2^8 + 2^9 = 1111001001 = 1+8+64+128+256+512 = 969(10)$



퀴즈 23

```
public class QuizNum23 {
    public static void main(String[] args) {
        final int BIN = 1;
        int testBit = 0;
        int testBit2 = 0;

        int randNum;
        int randNum2;

        for(int i=0; i<6; i++){ //5,6,7,8,9,10 6개의 숫자를 뽑기위해 6번반복
            randNum = (int)(Math.random()*6+5); //6을 곱해줘서 0부터 6미만의 범위에 5를 더해줘서 5 이상 11미만의 정수로 범위를 바꾸어준다

            while((testBit & (BIN <<randNum)) != 0){//testBit의 초기값은 0, 발생된난수의 중복확인
                randNum = (int)(Math.random()*6+5);
            }
            System.out.printf("randNum = %d \n",randNum);
            testBit |= (BIN << randNum);//비교후 중복되지않으면 testBit에 저장
        }

        for(int i=0; i<4; i++){
            randNum2 = (int)(Math.random()*4+7); //4을 곱해줘서 0부터 4미만의 범위에 7를 더해줘서 7 이상 11미만의 정수로 범위를 바꾸어준다
            while((testBit2 & (BIN <<randNum2)) != 0){//testBit2의 초기값은 0, 발생된난수의 중복확인

                randNum2 = (int)(Math.random()*4+7);
            }
            System.out.printf("randNum2 = %d \n",randNum2);

            testBit2 |= (BIN << randNum2); //비교후 중복되지않으면 testBit에 저장
        }
    }
}
```



퀴즈 23 출력창

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