

REPORT



- 과목명 : 객체지향프로그래밍
- 담당교수 : 엄진영 교수님
- 학과 : 컴퓨터공학과
- 학번 : 2019112130
- 이름 : 조 양 진

객체지향프로그래밍 6주차 실습과제 보고서

2019112130 조양진

문제 1.

다음 그림은 비트 연산자를 활용한 코드의 일부분이다.

```
c = -1; d = -1;
System.out.println("c >>> 2 = " + (c >>> 2));
System.out.println("d >>> 2 = " + (d >>> 2));
```

이 코드의 실행결과는 다음과 같다.

```
c >>> 2 = 1073741823
d >>> 2 = 4611686018427387903
```

주어진 코드는 단순하게 생각하면 동일한 -1이 어떻게 같은 연산을 했는데 값이 다른지 알 수 없습니다. 잘 확인해보면 변수 c와 d의 자료형에 대한 이야기가 빠져있습니다. 또한 비트 쉬프트 연산을 했는데 꽤 많은 자리수의 차이가 남을 보니 이는 자료형의 차이라고 생각되어, c의 자료형은 int, d의 자료형은 long으로 하면 되겠다는 생각이 들었습니다.

- 코드

```
// 2019112130 조양진
package week6;

public class Problem_1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int c = -1;
        long d = -1;

        System.out.println("c >>> 2 = " + (c >>> 2));
        System.out.println("d >>> 2 = " + (d >>> 2));
    }
}
```

```
Problem_1.java
1 package week6;
2
3 public class Problem_1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int c = -1;
8         long d = -1;
9
10        System.out.println("c >>> 2 = " + (c >>> 2));
11        System.out.println("d >>> 2 = " + (d >>> 2));
12    }
13
14 }
15
```

Console

<terminated> Problem_1 [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (2020. 10. 12. 오전 6:50:05)
c >>> 2 = 1073741823
d >>> 2 = 4611686018427387903

예상한대로 c의 자료형을 int로, d의 자료형을 long으로 주고 동일한 비트 연산 (>>>)을 하였더니 문제와 같은 결과값을 볼 수 있었습니다. 비트 연산자 '>>>'은 왼쪽에 있는 정수의 각 비트를 오른쪽에 있는 정수만큼 오른쪽으로 이동시키는 역할을 합니다.

우선 int형의 c의 경우

11111111_11111111_11111111_11111111

와 같은 형태로 이루어집니다. 이를 오른쪽으로 2번 쉬프트하고 0으로 채우면

00111111_11111111_11111111_11111111

이 됩니다. 이는 1073741823이며 문제에서 주어진 것과 동일합니다.

이후 long형의 d의 경우

11111111_11111111_11111111_11111111_11111111_11111111_11111111_11111111

에서 오른쪽으로 2번 쉬프트하고 0으로 채우면

00111111_11111111_11111111_11111111_11111111_11111111_11111111_11111111

그리고 이를 decimal로 바꾸면 4611686018427387903이 됩니다. 이는 문제에서 주어진 것과 동일합니다.

문제 2.

1. 아래 코드를 실행했을 경우 나오는 결과를 보고 빈칸에 들어갈 연산자를 정하세요.

```
package java_Week3;

public class sort {

    public static void main(String[] args) {
        int x = 0x0ffe;
        int y = 0xfff0;
        System.out.printf("%x\n", ~(x + 1) [A] y));
        System.out.printf("%x\n", ~(x - 14) [B] y));
    }
}
```

결과

```
ffffff00f
ffff000f
```

- 분석

int x에는 16진수로 0ffe, int y에는 16진수로 fff0이 대입되어 있습니다. 이 문제를 풀기 위해서는 수학문제를 풀듯이 차근차근 방정식을 풀어나가면 됩니다.

우리가 모르는 연산자를 ?라고 놓고 풀어보면

$$\sim((x + 1) ? y) = f00f$$

$$((x + 1) ? y) = f00f$$

$$x = 0ffe, y = fff0$$

$$x + 1 = 0fff$$

$$((0fff) ? fff0) = f00f$$

0fff와 fff0이 만나서 f00f가 되기 위해선? 자리에 &(AND) 비트 연산자가 올 수 밖에 없습니다.

$$\sim((x - 14) ? y) = ffff000f$$

$$x - 14 = 0ff0, y = fff0$$

$$\sim(0ff0) ? fff0 = 000f$$

$$0ff0 ? fff0 = fff0$$

0ff0과 fff0이 만나서 000f가 되기 위해선? 자리에 | (OR) 비트 연산자가 올 수 밖에 없습니다.

따라서 A자리에는 &, B자리에는 |가 들어가야 합니다.

```
// 2019112130 조양진
```

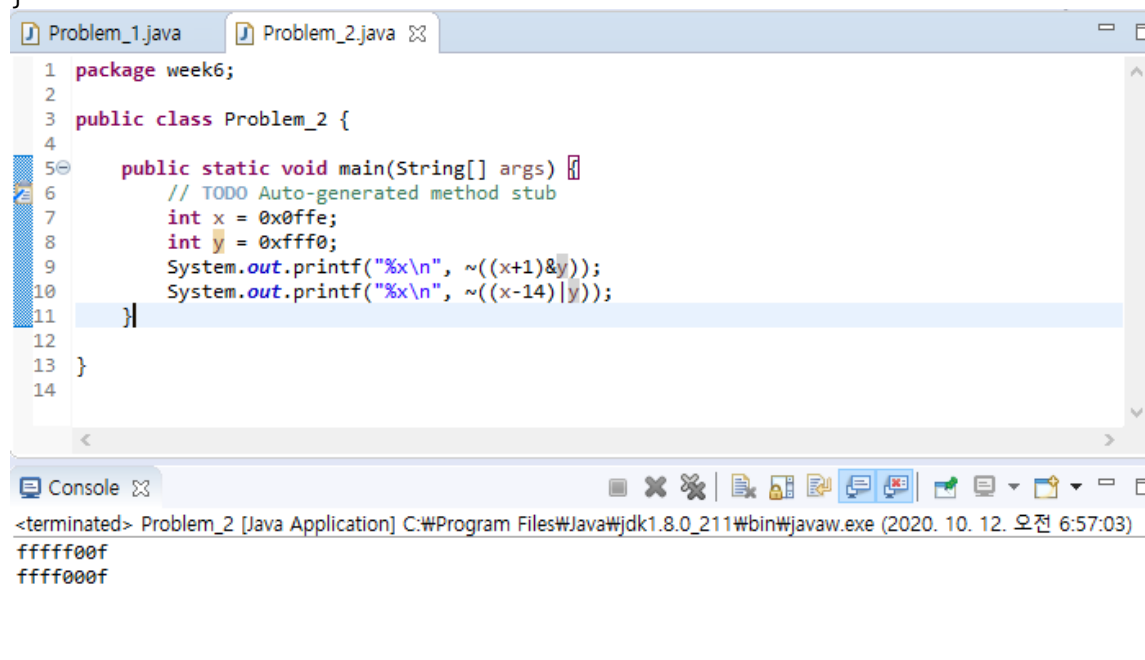
```
package week6;
```

```
public class Problem_2 {
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int x = 0x0ffe;  
        int y = 0xffff0;
```

```
        System.out.printf("%x\n", ~((x+1)&y));  
        System.out.printf("%x\n", ~((x-14)|y));
```

```
    }  
}
```



The screenshot shows an IDE with two tabs: Problem_1.java and Problem_2.java. The Problem_2.java tab is active, displaying the code from the previous blocks. The console window at the bottom shows the output of the program: fffff00f and ffff000f.

2. 위 코드의 x, y를 이용해 $(x+1)^{(y+15)}$ 의 결과를 구하고 계산 과정과 상세한 설명을 작성하세요.

$(x+1)$ 과 $(y+15)$ 의 XOR 연산을 해야 합니다.

우선 $x+1$ 과 $y+15$ 는 각각 fff, ffff입니다.

XOR 연산은 둘 다 f일 때 0으로 바꿉니다. 따라서 $(fff)^{(ffff)}$ 이므로 f000이 됩니다.

코드로 검산해 보면 다음과 같습니다.

```
// 2019112130 조양진
```

```
package week6;
```

```
public class Problem_2 {
```

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int x = 0x0ffe;
    int y = 0xffff0;

    System.out.printf("%x\n", x+1);
    System.out.printf("%x\n", y+15);
    System.out.printf("%x\n", (x+1)^(y+15));
}
}

1 // 2019112130 조양진
2 package week6;
3
4 public class Problem_2 {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         int x = 0x0ffe;
9         int y = 0xffff0;
10
11         System.out.printf("%x\n", x+1);
12         System.out.printf("%x\n", y+15);
13         System.out.printf("%x\n", (x+1)^(y+15));
14     }
15 }
16

<terminated> Problem_2 [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\java
fff
ffff
f000
```

결과도 동일하게 나왔습니다.