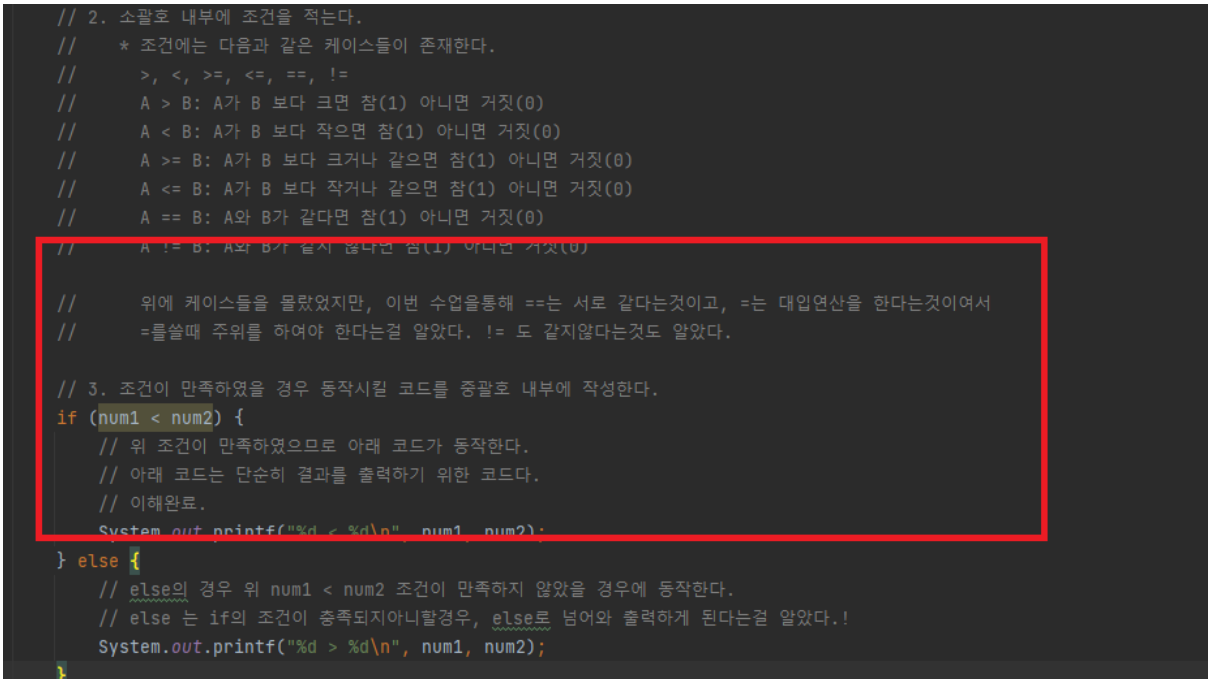
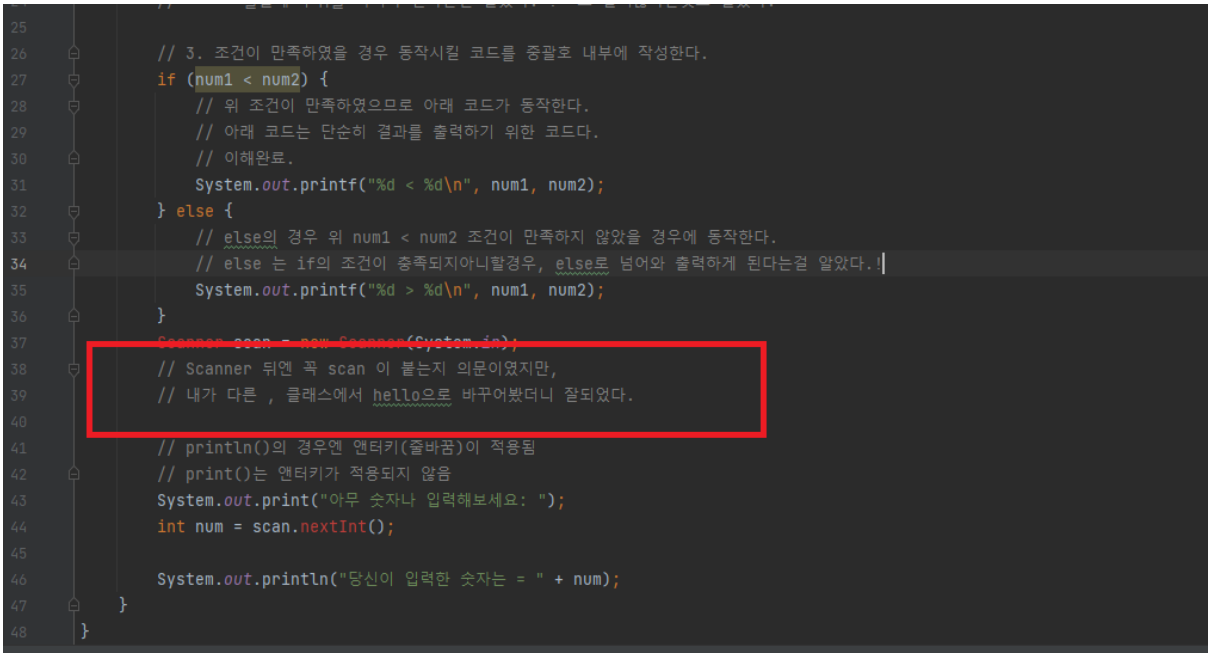
[한상우]5월11일 자바3일차 복습.

빨간 네모창안에 저의 생각이 적혀있습니다.
추가적으로 적은 내용은 pdf파일안에 있습니다!



```
// 2. 소괄호 내부에 조건을 적는다.
//    * 조건에는 다음과 같은 케이스들이 존재한다.
//      >, <, >=, <=, ==, !=
//      A > B: A가 B 보다 크면 참(1) 아니면 거짓(0)
//      A < B: A가 B 보다 작으면 참(1) 아니면 거짓(0)
//      A >= B: A가 B 보다 크거나 같으면 참(1) 아니면 거짓(0)
//      A <= B: A가 B 보다 작거나 같으면 참(1) 아니면 거짓(0)
//      A == B: A와 B가 같다면 참(1) 아니면 거짓(0)
//      A != B: A와 B가 같지 않다면 참(1) 아니면 거짓(0)

//      위에 케이스들을 몰랐었지만, 이번 수업을통해 ==는 서로 같다는것이고, =는 대입연산을 한다는것이여서
//      =를쓸때 주위를 하여야 한다는걸 알았다. != 도 같지않다는것도 알았다.

// 3. 조건이 만족하였을 경우 동작시킬 코드를 중괄호 내부에 작성한다.
if (num1 < num2) {
    // 위 조건이 만족하였으므로 아래 코드가 동작한다.
    // 아래 코드는 단순히 결과를 출력하기 위한 코드다.
    // 이해완료.
    System.out.printf("%d < %d\n", num1, num2);
} else {
    // else의 경우 위 num1 < num2 조건이 만족하지 않았을 경우에 동작한다.
    // else 는 if의 조건이 충족되지아니할경우, else로 넘어와 출력하게 된다는걸 알았다.!
    System.out.printf("%d > %d\n", num1, num2);
}
```

//



```
        // 3. 조건이 만족하였을 경우 동작시킬 코드를 중괄호 내부에 작성한다.
        if (num1 < num2) {
            // 위 조건이 만족하였으므로 아래 코드가 동작한다.
            // 아래 코드는 단순히 결과를 출력하기 위한 코드다.
            // 이해완료.
            System.out.printf("%d < %d\n", num1, num2);
        } else {
            // else의 경우 위 num1 < num2 조건이 만족하지 않았을 경우에 동작한다.
            // else 는 if의 조건이 충족되지아니할경우, else로 넘어와 출력하게 된다는걸 알았다.!
            System.out.printf("%d > %d\n", num1, num2);
        }
        Scanner scan = new Scanner(System.in);
        // Scanner 뒤엔 꼭 scan 이 붙는지 의문이였지만,
        // 내가 다른 , 클래스에서 hello으로 바꾸어봤더니 잘되었다.

        // println()의 경우엔 엔터키(줄바꿈)이 적용됨
        // print()는 엔터키가 적용되지 않음
        System.out.print("아무 숫자나 입력해보세요: ");
        int num = scan.nextInt();

        System.out.println("당신이 입력한 숫자는 = " + num);
    }
}
```

//

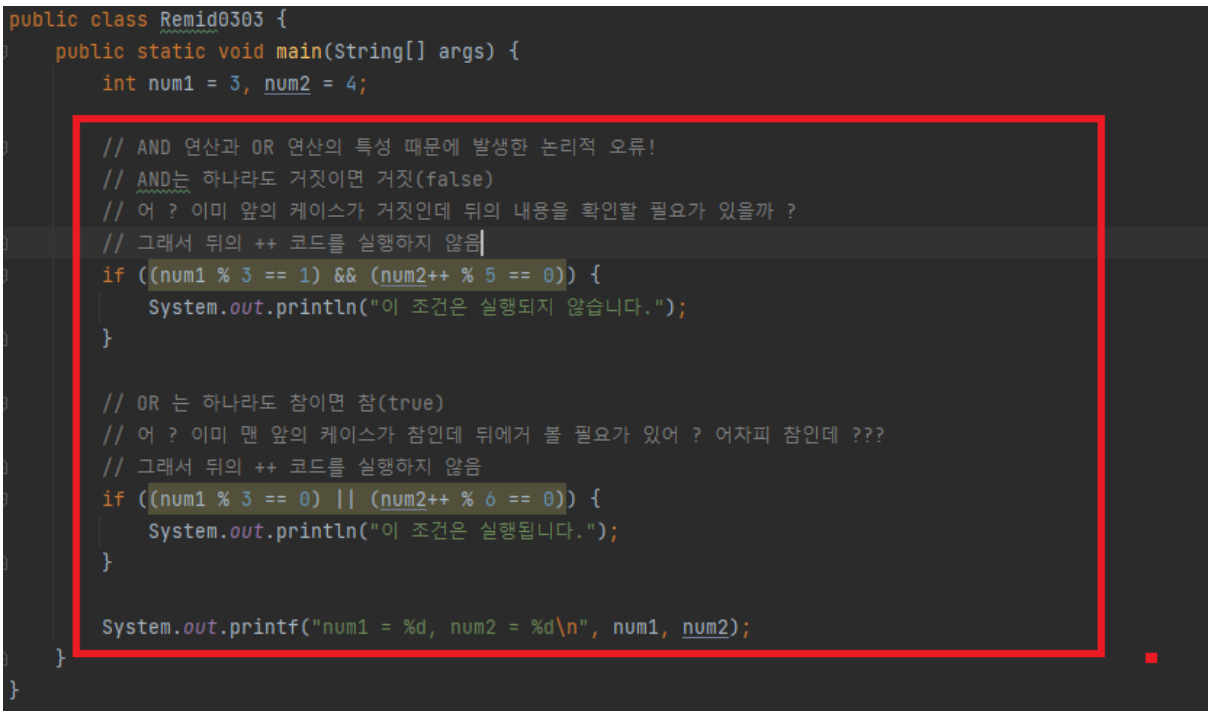
Scanner 함수뒤에 변수는 내가 지정해서 쓰는것. 하지만, 나의목적을 뚜렷히 알고자,
이것을 스캔한다. 해서 scan 이라는 변수로써 쓰면 나중에 코드 가시성이 좋아질것이다.

else if 를 추가적으로 넣어 입력값이 정확하게 나오도록 바꾸었습니다.

```java
import java.util.Scanner;

public class Remind0302 {
    public static void main(String[] args) {
        System.out.println("두 개 숫자를 입력 받아 비교해봅니다.");
        Scanner scan = new Scanner(System.in);

        System.out.print("첫 번째 숫자를 입력하세요: ");
        int num1 = scan.nextInt();

        System.out.print("두 번째 숫자를 입력하세요: ");
        int num2 = scan.nextInt();

        // 문제: 같은 숫자를 넣으면 원하는 동작을 하지 않는다.
        // 원인: 현재 조건은 둘 중 하나가 무조건 큰 경우만 보고 있음
        // 즉 두 숫자가 같은 경우를 전혀 고려하고 있지 않다.
        // 그러므로 이 부분에 대한 대응이 필요하다!


        if (num1 > num2)
        {
            System.out.printf("%d > %d\n", num1, num2);
        }else if(num1 == num2)// 수업도중 강사님이, 서로같은 숫자일땐, else로 빠져나가 같은숫자여도 %d = %d  로나오지않고
                              // else 값이 도출되어  내가 else if 달아서 num1과  num2 가 같을땐 %d = %d 로 출력하도록 수정하였다.
            {
            System.out.printf("%d = %d\n",num1,num2);
        }
        else
        {
            System.out.printf("%d < %d\n", num1, num2);
        }
    }
}
```

```java
import java.util.Scanner;

public class Remind03 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("n의 수중 4의배수를 구할것이다. n을 기입하시오.");
        int loop = scan.nextInt();
        int i = 1 ;
        while(i <= loop)

        {
            if(i % 4 == 0)

            {
                System.out.println("1~부터"+ loop+"까지의4의배수를 출력합니다"+ i);
                //제가 식을 하나 다시 복습하여 만들어봤습니다.
                // 맨위에 있는, 룹에다가 내가 입력한숫자를 대입 한다,
                // while의 식 while(){}i++로 ()은 와일의 작동 식이고 {}안에는 와일이 출력할 칸
                // {}밖에는 while이 무한대로 돌려지지않게 정해진 숫자만큼돌게끔 증감식 혹은 증가식인 i++을 넣는다.
                // 다시 문제로 돌아가서 만약 제가 50을 대입하였다면. while 식에 의하여, i는 50에 가까워지려고 하겠다.
                // 1부터 시작하여 , if라는 필터링안에 들어갈것이다, 만약 if안의식(i % 4 == 0)이  해당 즉 i가 4의 배수이면
                // if 의 식에 충족되어 4때 8 때 12 때.... 48때 걸려서 출력이 될것이다.
                // 고로 System.out.println("1~부터"+ loop+"까지의4의배수를 출력합니다"+ i) 출력란엔
                // 최초 걸리는 4가 걸릴것이고 다음은 8,12... 로 50아래의 4의배수가 걸릴것이다.

            }
            else if(loop < 4 )
            {
                System.out.println("입력된값의 범위가 4이내로 출력이 되지않습니다.");
            }
            //  저는 여기서 입력값이 1,2,3,일때 아무것도 뜨지않는 완성적인 코드가 아니라고 판단하여,
            //  else if를 넣어주어 loop의 값이 4미만일땐, 해당 문구를 넣었습니다.
            i++;
```

/ 처음엔 이해가 전혀되질않았는데,
/ 차근차근 하나씩 식을 풀어가며 생각하고 적어보고
/while 문 밖에는 증감식 있어야 무한도출 되지않는다는 개념도 이해했습니다.
대입 연산자에서 이 코드의 도출값이



```java
public class Remid0303 {
    public static void main(String[] args) {
        int num1 = 3, num2 = 4;

        // AND 연산과 OR 연산의 특성 때문에 발생한 논리적 오류!
        // AND는 하나라도 거짓이면 거짓(false)
        // 어 ? 이미 앞의 케이스가 거짓인데 뒤의 내용을 확인할 필요가 있을까 ?
        // 그래서 뒤의 ++ 코드를 실행하지 않음
        if ((num1 % 3 == 1) && (num2++ % 5 == 0)) {
            System.out.println("이 조건은 실행되지 않습니다.");
        }

        // OR 는 하나라도 참이면 참(true)
        // 어 ? 이미 맨 앞의 케이스가 참인데 뒤에거 볼 필요가 있어 ? 어차피 참인데 ???
        // 그래서 뒤의 ++ 코드를 실행하지 않음
        if ((num1 % 3 == 0) || (num2++ % 6 == 0)) {
            System.out.println("이 조건은 실행됩니다.");
        }

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

대입 연산자에서 이 코드의 도출값이
이조건은 실행됩니다.
num1 =3 num2 = 4로나왔는데,
앤드연산자의특성하나라도 거짓이면거짓(false), 오어 연산자의 특성 하나라도 참이면 참(true)이여서 앤드는 앞에식이 거짓이라면 뒤에 식을 풀지않고 넘겨버리며,
오어의 앞에식이 참이면 뒤에식을 풀지않는 특성으로 나타난 일이다.

9번 첼린지문제는,
1000개의 데이터가 있다.
여기서 C에 해당하는 데이터는 30개 있다.
F에 해당하는 데이터는 500개 있다.
B에 해당하는 데이터는 240개 있다.
A에 해당하는 데이터는 700개 있다.
D에 해당하는 데이터는 350개 있다.
C 혹은 F 둘 중 하나의 케이스를 판정하고자 한다.
어떻게 하면 가장 효율적으로 이 케이스들을 찾아낼 수 있을까 고민해보자!

c F의 교집합을 찾는것으로 이해하였다.
고로 효율적으로 찾는다면,
c데이터 를 먼저 간추리고 F를 그사이에서 찾으면 될것같은데
아마 답은 아닐거같다