

22. 10 .12 - 연산자 종류

자바에서는 논리 취급 안한다 = 트루스 펼지 지원 xxx

자바 ,C 언어는 $20 < \text{BAR} < 30$ 이렇게 안된다
⇒ 비교연산자는 반드시 이항연산자로만 사용된다.



비교 연산자 (이항연산)

>	<=
<	<=
==	
!=	

1. 이항연산을 한다 ⇒ 피연산자를 같은 타입으로 변환한 후에 비교한다
2. 대상 결과 값의 자료형이 항상 “**불린 형**(true , false) ” 이 나온다

특정 구간을 지정하려 할 때

논리 연산자를 결합하여 사용 해야 한다.

⇒ `bar>20 && bar <30`



논리 연산자

“&&” 가 “||” 보다 세다!!!!

AND	OR
&&	

⇒ & 이렇게 한개는 비트 연산자의 AND

논리 연산자 사용시 대입 연산자 넣지 말기

```
package Test;

import java.util.Scanner;

// Comment (주석)
// 이유?
// 1) 소스 코드를 설명하기 위해서
// 2) 디버깅
// 주석 작성 방법
// 1) // -> 한 줄
// 2) /* */ -> 다중라인

// Variable + Array
// Variable
// 1) 선언방법
// - 자료형 변수명;
// - 자료형 변수명 = 초기값;
// 2) 자료형
// 3) 접근범위 (Scope)
// 4) 생명주기 (Lifecycle)

// Constant
// 자료를 저장하는 의미에서는 변수와 동일
// 단 한 번 저장된 값은 프로그램 종료까지 절대 변하지 않는다.
// 두 가지 종류가 있음
// 1) General constant(일반상수, 이름있는 상수)
// 2) Literal constant(리터럴 상수, 이름없는 상수)

// Operator
// Flow control
// Function

// FishCake 클래스를 정의
class FishCake {

    // Member variable (멤버변수)
    String name;

    // Member method (멤버메서드)
    void prtNameOfFish() {
        System.out.println(name);
    }
}
```

```

    }

    // Constructor (생성자)
    FishCake() {
        System.out.println("초기화 작업이 실행 됩니다.");
    }
}

public class MyFoo {
    public static void main(String args[]) {

        // 짝수 이면서, 7의 배수이면서, 10의 배수인 값
        // bar % 2 == 0 && bar % 7 == 0 && bar % 10 == 0

        // 20 < bar < 30
        int bar = 1;

        if ( 3 > 2 || (bar = bar + 1) > 2 )
            System.out.println("true");

        System.out.println(bar);

        //System.out.println( bar > 20 && bar <= 30 );
        // System.out.println( false && false );
        // System.out.println( true && false );
        // System.out.println( false && true );
        // System.out.println( true && true );
        //
        // System.out.println( false || false );
        // System.out.println( true || false );
        // System.out.println( false || true );
        // System.out.println( true || true );
        //
        // 비교 연산자
        // >   >=
        // <   <=
        // ==
        // !=

        // 논리연산자
        // &&   -> AND
        // ||   -> OR

    }
}

```

성적 입력 코드

예외를 제일 위에 조건으로 작성하기

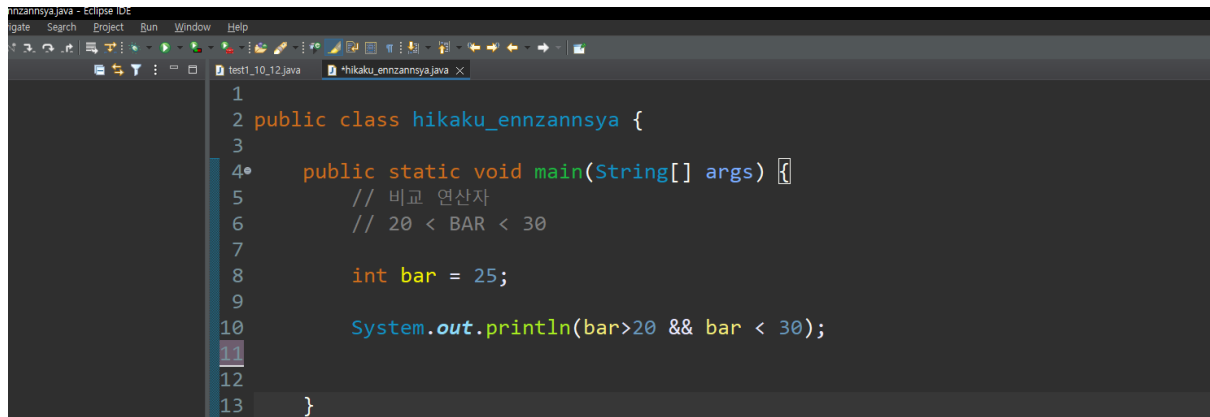
```

public class MyFoo {
    public static void main(String args[]) {

        // 1. 0~100 사이의 정수를 키보드로부터 입력 받아 변수에 저장하고
        Scanner scn = new Scanner(System.in);
        System.out.println("0 ~ 100 사이 정수를 입력하세요");
        int inputValue = scn.nextInt();
        String result;

        // 2. 아래 등급표의 등급을 출력하라
        // 2-1.
        //100 초과 0 미만 -> "입력 값 오류"
        if(inputValue > 100 || inputValue < 0) {
            result = "입력 값 오류";
        }
        // 90 이상 -> A
        else if(inputValue >= 90) {
            result = "A";
        }
        // 80 이상 -> B
        else if(inputValue >= 80) {
            result = "B";
        }
        // 70 이상 -> C
        else if(inputValue >= 70) {
            result = "C";
        }
        // 60 이상 -> D
        else if(inputValue >= 60) {
            result = "D";
        }
        // 60 미만 -> F
        else {
            result = "F";
        }
        System.out.println(result);
    }
}

```



```
1
2 public class hikaku_enzannnya {
3
4     public static void main(String[] args) {
5         // 비교 연산자
6         // 20 < BAR < 30
7
8         int bar = 25;
9
10        System.out.println(bar>20 && bar < 30);
11
12
13    }
```

삼항연산자

// if else 와 같은 기능을 가진다.

```
System.out.println((inputValue == 1) ? "남성" : "여성");
```

// 비트연산자 [컴퓨터 구조론과 관계가 있다]

```
// 복합 연산자 ( 두개 이상 결합 )
// 서로다른 연산자가 결합이 되어 사용 되는 것
// 대입연산자 와 산술연산자 또는 비트 연산자 가 같이 결합이 되어 쓰인다.

int bar = 10;

bar = bar + 10;
bar += 10;

bar = bar - 10;
bar -= 10;

bar = bar * 10;
bar *= 10;

bar = bar / 10;
bar /= 10;

bar = bar % 10;
bar %= 10;
```

```

public class monndai {

    public static void main(String[] args) {

        // for 문 사용
        // 1 ~ 100 정수 중 짝수이면서
        // 8의 배수 또는 5의 배수인 값만 출력하라

        // 1. 1 ~ 100 까지 반복 할 for 문 작성
        for (int Num = 1 ; Num < 100 + 1 ; ++Num) {
            // 2. 조건식 ( 짝수 )
            if (Num % 2 == 0) {
                // 3. 8의 배수 또는( || ) 5의 배수
                if (Num % 8 == 0 || Num % 5 == 0) {
                    // 4. 結果だし
                    System.out.println(Num);
                }
            }
        }

    }

}

```