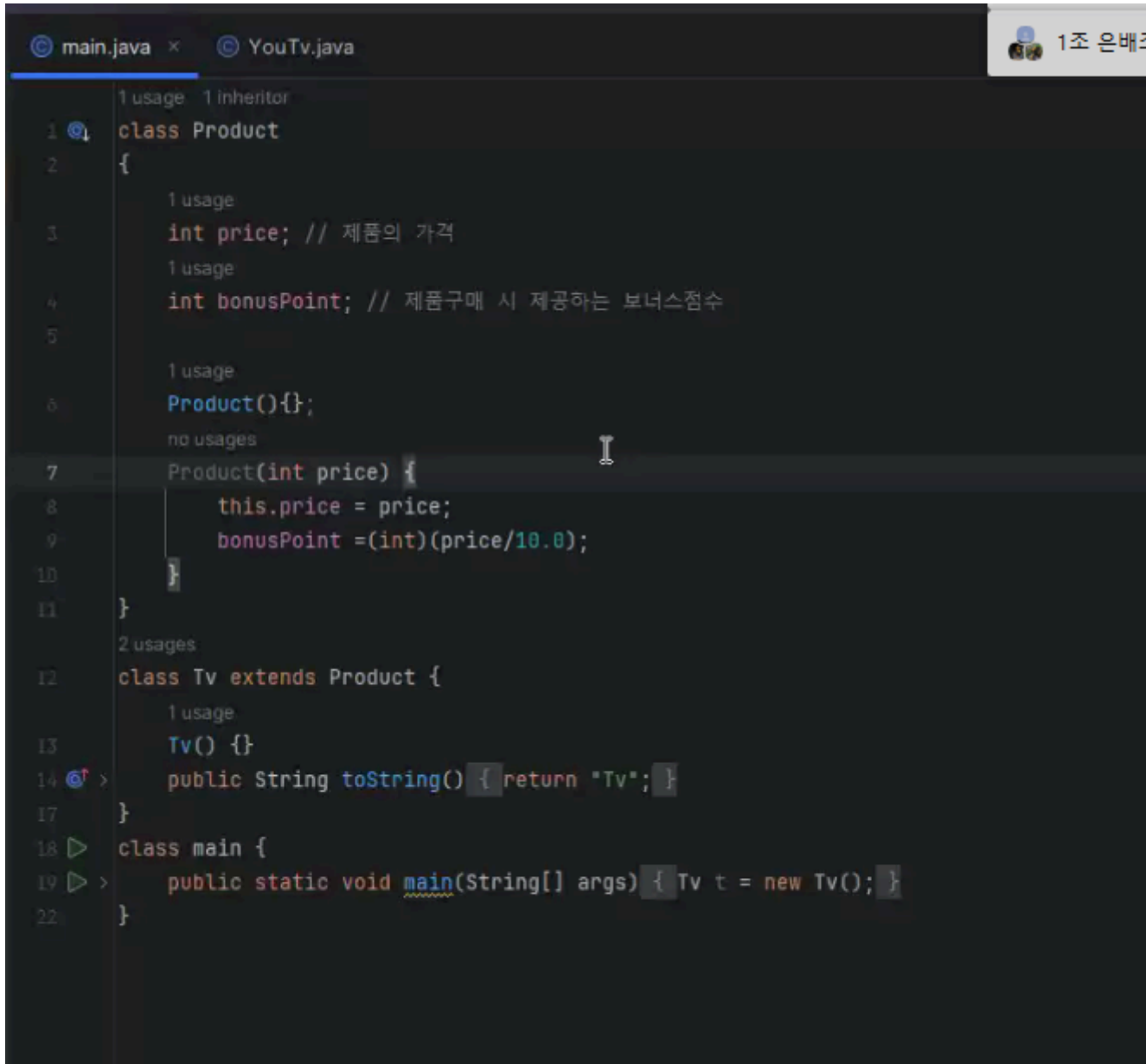


231220

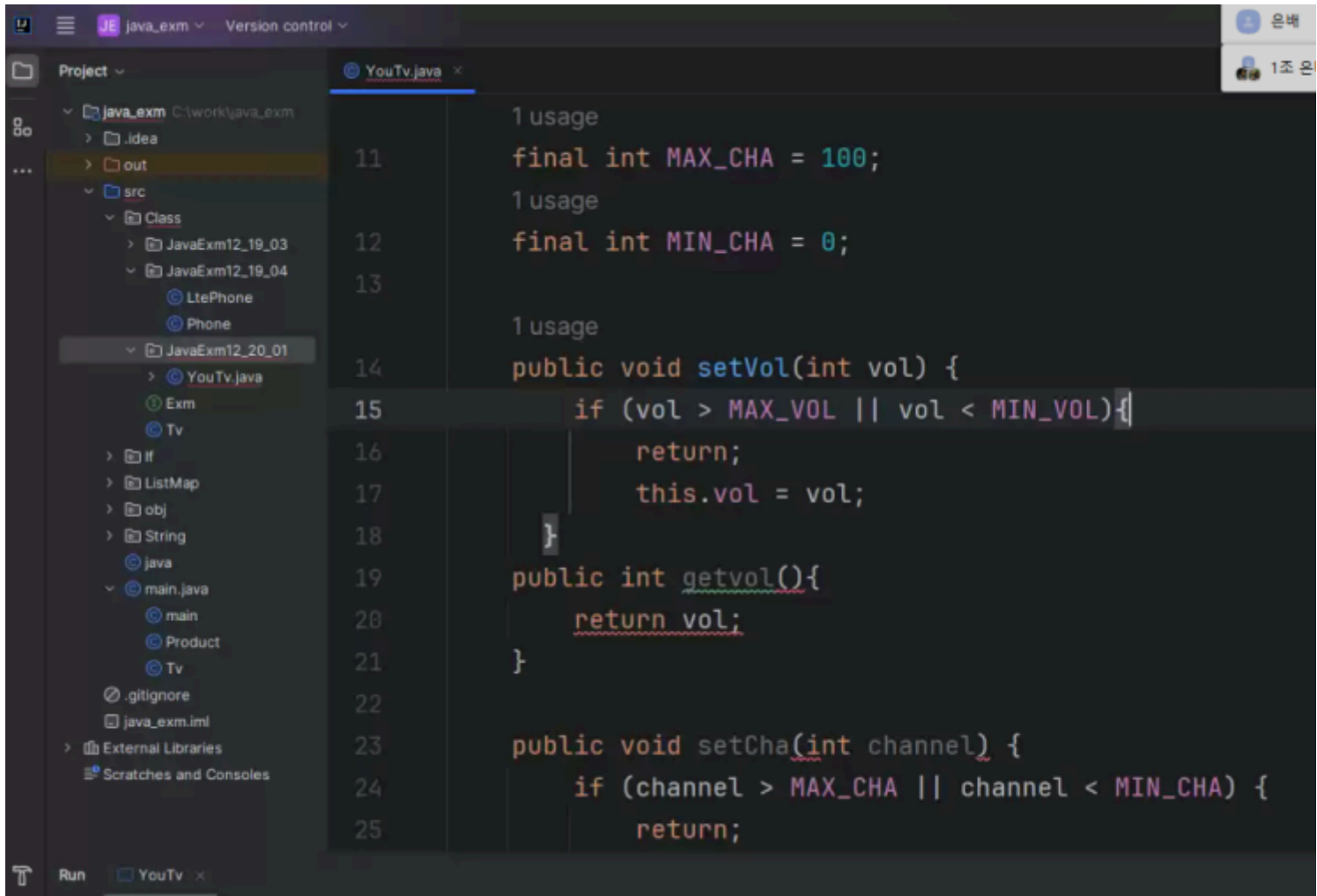
자바스프링

2023/12/20 18:33

<http://blog.naver.com/sophia2164/223298673193>

A screenshot of a Java IDE with two tabs: 'main.java' and 'YouTv.java'. The 'main.java' tab is active, showing a Java class hierarchy. At the top, '1 usage 1 Inheritor' is displayed. The code defines a 'Product' class with attributes 'price' and 'bonusPoint', a no-args constructor, and a constructor that takes a 'price' parameter. It then defines a 'Tv' class that extends 'Product' and overrides the 'toString()' method. Finally, it defines a 'main' class with a 'main' method that creates a new 'Tv' object. The code is as follows:

```
1 usage 1 Inheritor
1 class Product
2 {
3     1 usage
4     int price; // 제품의 가격
5     1 usage
6     int bonusPoint; // 제품구매 시 제공하는 보너스점수
7
8     1 usage
9     Product(){};
10
11     no usages
12     Product(int price) {
13         this.price = price;
14         bonusPoint =(int)(price/10.0);
15     }
16 }
17 2 usages
18 class Tv extends Product {
19     1 usage
20     Tv() {}
21     public String toString() { return "Tv"; }
22 }
23
24 class main {
25     public static void main(String[] args) { Tv t = new Tv(); }
26 }
```



```
44         t.setCha(10);
45         System.out.println(t.getCha());
46         t.setCha(50);
47         System.out.println(t.getCha());
48         t.setCha(25);
49         System.out.println(t.getCha());
50         t.setCha(16);
51         System.out.println(t.getCha());
52
53         t.gotoPrevchannel();
54         System.out.println(t.getCha());
55         t.gotoPrevchannel();
56         System.out.println(t.getCha());
57     }
58 }
59
```

객체 변수 value 가 100 보다 큰 값은 가질 수 없도록 제한하는 MaxLimitCalculator 클래스를 만들어 보자  
즉, 다음과 같이 동작해야 한다.

```
MaxLimitCalculator cal = new MaxLimitCalculator();  
cal.add(50); // 50 더 하 기  
cal.add(60); // 60 더 하 기  
System.out.println(cal.getValue()); // 100 출 력
```

단, MaxLimitCalculator 클래스는 반드시 다음의 Calculator 클래스를 상속해서 만들어야 한다.

```
class Calculator {  
    int value;  
    Calculator() {  
        this.value = 0;  
    }  
    void add(int val) {  
        this.value += val;  
    }  
    int getValue() {  
        return this.value;  
    }  
}
```

```

1 package Class.JavaExm12_20_01;
2
3 class Calculator {
4     int value;
5     Calculator() {
6         this.value = 0;
7     }
8     void add(int val) {
9         this.value += val;
10    }
11    int getValue() {
12        return this.value;
13    }
14 }
15 class MaxLimitCalculator extends Calculator {

```

```
14 }
15 2 usages
16 class MaxLimitCalculator extends Calculator {
17     2 usages
18     void add(int val){
19         this.value += val;
20         if (this.value > 100) {
21             this.value = 100;
22         }
23     }
24 }
25 public class Sample {
26     public static void main(String[] args) {
27         MaxLimitCalculator cal = new MaxLimitCalculator();
28         cal.add(35); // 50 더 하 기
29         cal.add(60); // 60 더 하 기
30         System.out.println(cal.getValue()); // 100 출 력
31     }
32 }
```

다음과 같이 정수 배열 또는 정수의 리스트로 그 평균을 구해 리턴하는 Calculator 클래스를 작성.

```
int[] data = {1, 3, 5, 7, 9};
Calculator cal = new Calculator();
int result = cal.avg(data);
System.out.println(result); // 5 출력
```

```
5
ArrayList<Integer> data = new ArrayList<>(Arrays.asList(1, 3, 5, 7, 9));
Calculator cal = new Calculator();
int result = cal.avg(data);
System.out.println(result); // 5 출력
5
```

```
20 //}
21 class main {
22     public static void main(String[] args) {
23         ArrayList<Integer> a = new ArrayList<>(Arrays.asList(1, 2, 3));
24         ArrayList<Integer> b = a;
25         a.add(4);
26         System.out.println(b.size());
27         System.out.println(a == b);
28     }
```