

Day08 (H1)

Calc class
CalcTest class
X class

20150819

Copyright (c) 2015 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

CalcTest.java

class Calc

static calcAvg()

static findMax()

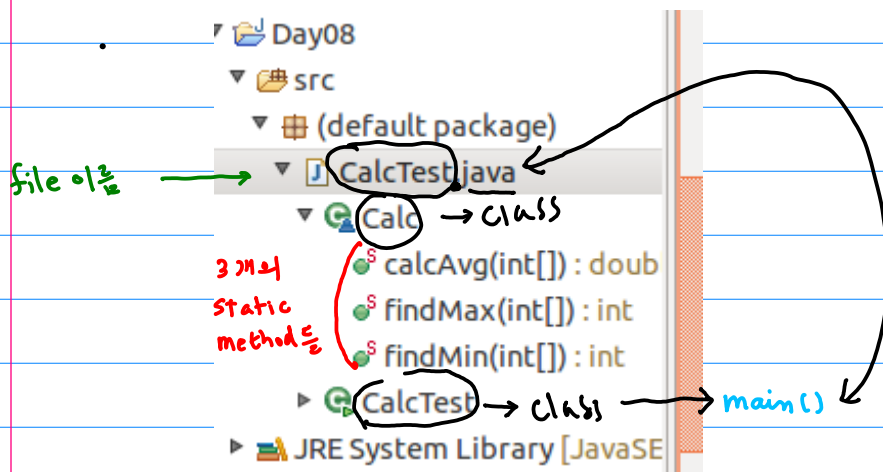
static findMin()

class

CalcTest

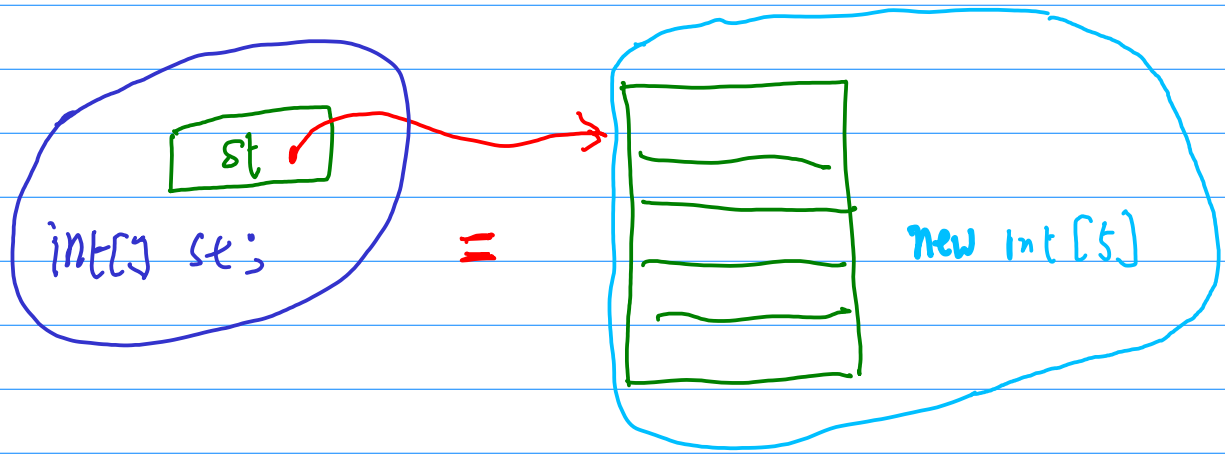
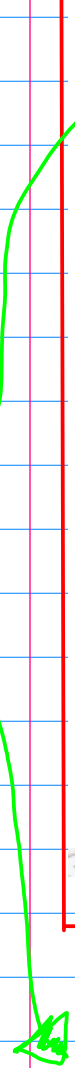
CalcTest.java

```
class Calc {  
    public static double calcAvg(int[] a) {  
        int i, Sum = 0; double Avg;  
        for (i=0; i<a.length; ++i) Sum += a[i];  
        Avg = (double) Sum / a.length;  
        return Avg;  
    }  
  
    public static int findMax(int[] a) {  
        int i, max = -1;  
        for (i=0; i<a.length; ++i)  
            if (max < a[i]) max = a[i];  
        return max;  
    }  
  
    public static int findMin(int[] a) {  
        int i, min = 999;  
        for (i=0; i<a.length; ++i)  
            if (min > a[i]) min = a[i];  
        return min;  
    }  
}
```



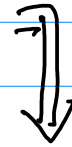
CalcTest.java

```
public class CalcTest {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int[] st = new int[5];  
  
        st[0] = 80; st[1] = 95; st[2] = 99;  
        st[3] = 92; st[4] = 85;  
  
        System.out.println("st.length= " + st.length);  
  
        int i;  
        for (i=0; i<5; ++i) {  
            System.out.println("st[" + i + "]= " + st[i])  
        }  
  
        double Avg = Calc.calcAvg( st );  
        int max = Calc.findMax( st );  
        int min = Calc.findMin( st );  
  
        System.out.println("Avg= " + Avg);  
        System.out.println("max= " + max);  
        System.out.println("min= " + min);  
    }  
}
```



int[] st = new int[5];

CalcTest class의 main 함수의 일부



```
class CalcTest {  
    ..... main ( ..... ) {  
  
        double Avg = calcAvg( st );  
        int max = Calc.findMax( st );  
        int min = Calc.findMin( st );  
  
    }  
}
```

함수들은

CalcTest class 내의
~~method~~

Calc.findMax()

Calc class 내

member 함수임을 표시

Static method 이어서

~~객체 이름~~. findMax() 라고

class 이름. findMax() 일

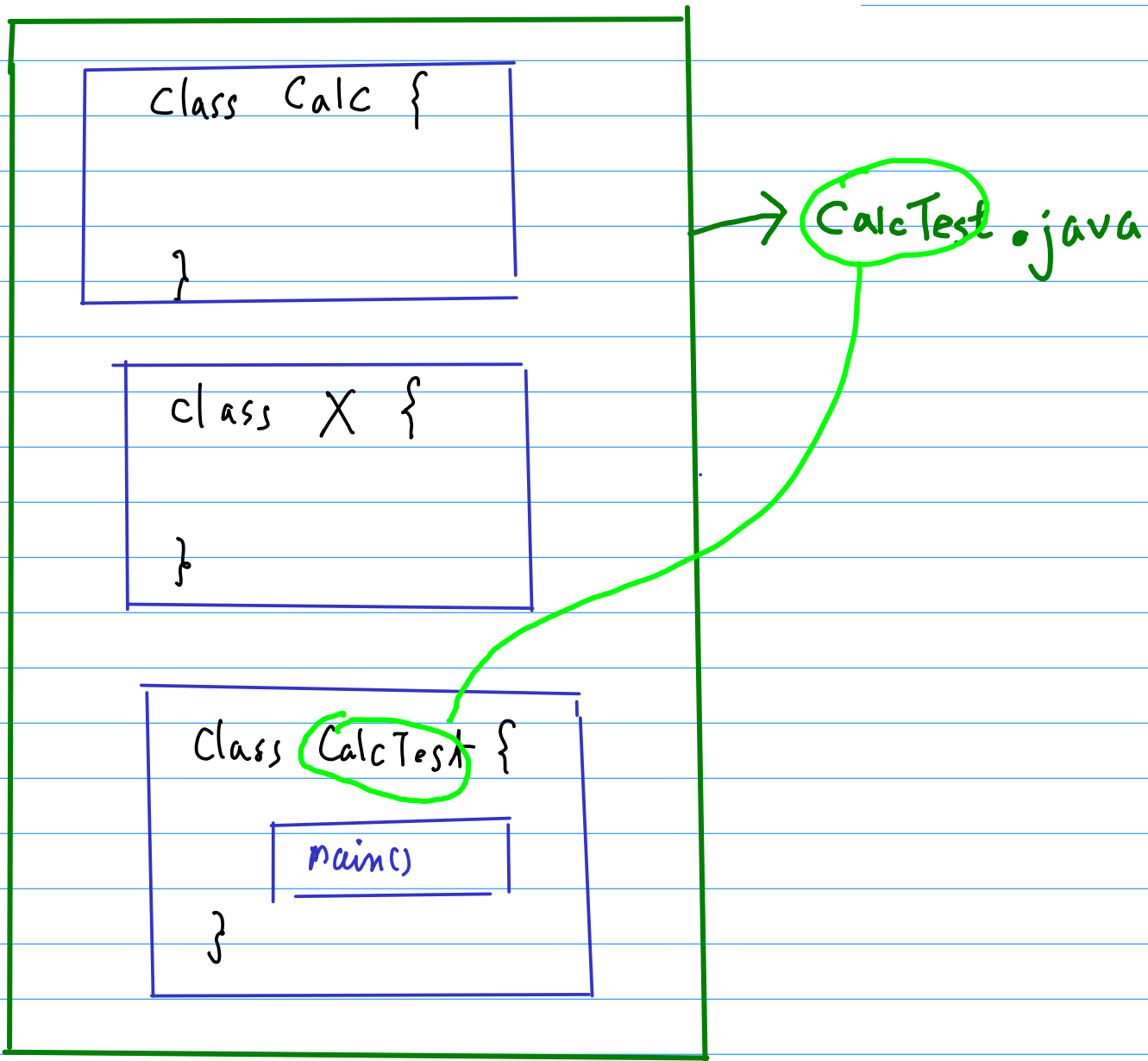
X class

2개의 입력

```
class X {  
    public static void pr(String s, int a) {  
        System.out.println(s + a);  
    }  
}
```

class 이름
한글이름
"max="

max
↓



Project 01 $\frac{2}{2}$

The screenshot shows a project named 'Day08' with a source folder 'src' containing a 'default package'. Inside this package are three Java files: 'Calc.java', 'CalcTest.java', and 'X.java'. Each file contains one or more classes with static methods. Handwritten blue annotations identify these as 'Static method'.

- Calc.java**
 - Calc class**
 - calcAvg(int[]): double } Static method
 - findMax(int[]): int } Static method
 - findMin(int[]): int } Static method
- CalcTest.java**
 - CalcTest class**
 - main(String[]): void } Static method
- X.java**
 - X class**
 - pr(String, double): void } Static method
 - pr(String, int): void } Static method

JRE System Library [JavaSE-1.7]

RectTest.java

```
StringTest.java | CalcTest.java | CalcTest.java | Calc.java | X.java | *RectTest.java x
```

```
class Rect {  
    int w;  
    int h;  
    static int count;  
  
    Rect() { w=0; h=0; count++; }  
    Rect(int x, int y) { w=x; h=y; count++; }  
  
    int area() { return w*h; }  
}
```

```
public class RectTest {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        System.out.println("count=" + Rect.count);  
        Rect R1 = new Rect(10, 20);  
        Rect R2 = new Rect(30, 40);  
        Rect R3 = new Rect(50, 50);  
        Rect R4 = new Rect(60, 60);  
        System.out.println("count=" + Rect.count);  
    }  
}
```

정리 4개 한문다

Writable Smart Insert 25


```

System.out.println("count=" + Rect.count);
Rect R1 = new Rect(10, 20);
Rect R2 = new Rect(30, 40);
Rect R3 = new Rect(50, 50);
Rect R4 = new Rect(60, 60);
System.out.println("count=" + Rect.count);

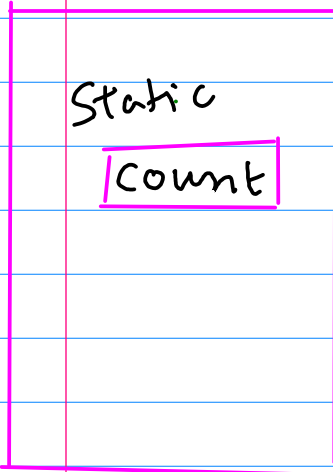
```

각각 4개

객체별
다른 w, h 값

AREA() 기가이 코드

1개만 존재



Rect.count

class 이름

