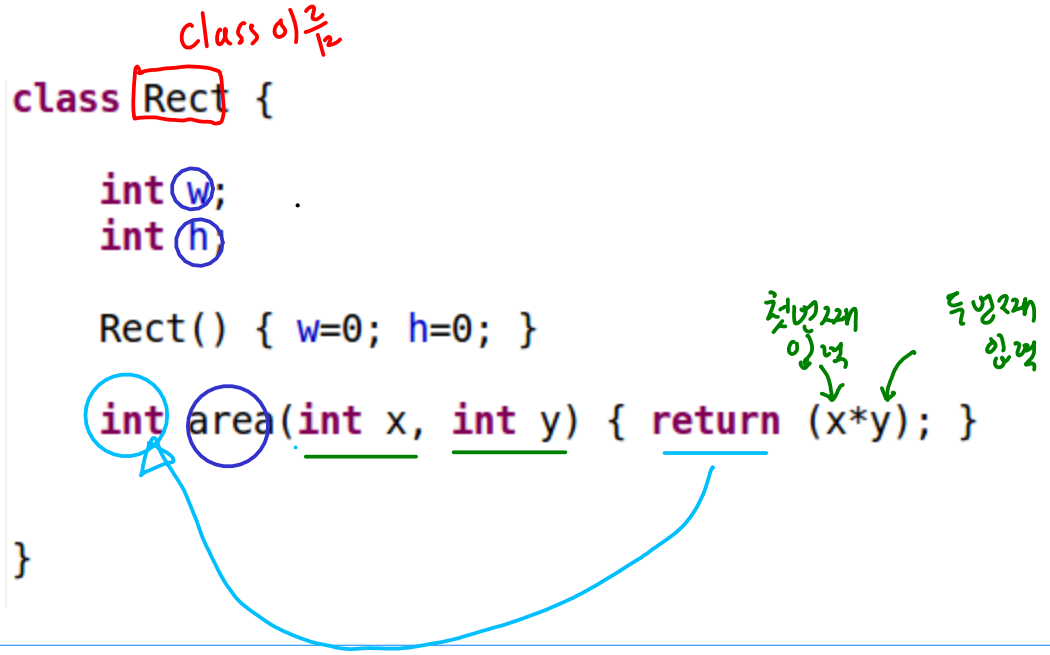
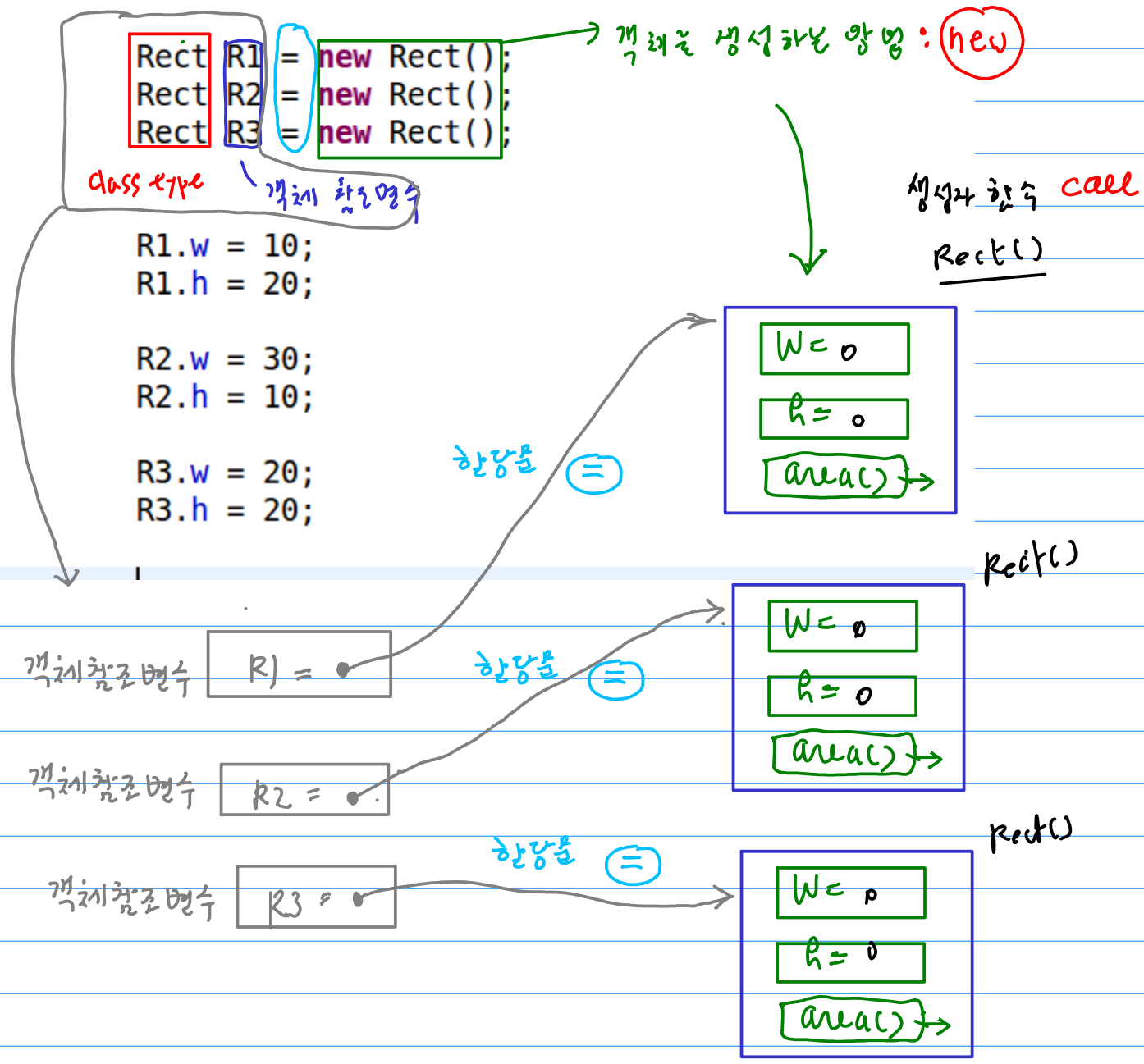
Day04 (H1)

20150817

Class and Objects
Method
Static Method
Constructor


class 이름
class Rect {
int w;
int h;
Rect() { w=0; h=0; }
int area(int x, int y) { return (x*y); }
}
첫번째 인자
두번째 인자

```java
public class RectTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Rect R1 = new Rect();
        Rect R2 = new Rect();
        Rect R3 = new Rect();

        R1.w = 10;
        R1.h = 20;

        R2.w = 30;
        R2.h = 10;

        R3.w = 20;
        R3.h = 20;
```



```java
public class RectTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Rect R1 = new Rect();
        Rect R2 = new Rect();
        Rect R3 = new Rect();

        R1.w = 10;
        R1.h = 20;

        R2.w = 30;
        R2.h = 10;

        R3.w = 20;
        R3.h = 20;
```

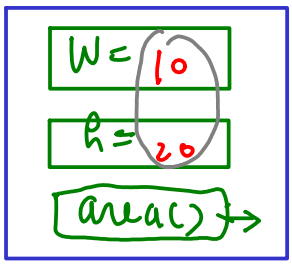
class type

객체 참조변수

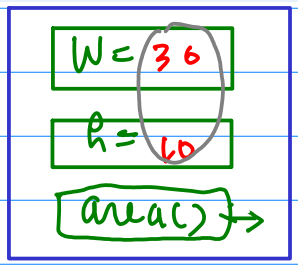
객체를 생성하는 방법

생성자 함수

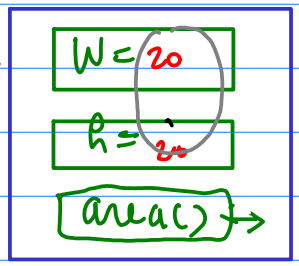
Rect()



Rect()



Rect()



R1 =

R2 =

R3 =

R1 이 가리키는 객체의 w & h 수정

R2 가 가리키는 객체의 w & h 수정

R3 가 가리키는 객체의 w & h 수정

생성자 함수

class 이름

```
class Rect {
    int w;
    int h;

    Rect() { w=0; h=0; }

    void display() {
        System.out.print("w=");
        System.out.println(w);
        System.out.print("h=");
        System.out.println(h);
    }
}
```

class 이름과 같은 함수 = 생성자 함수
⇒ 초기화
⇒ return 값 X
⇒ 그래서 void X

void 차이점

일반 method

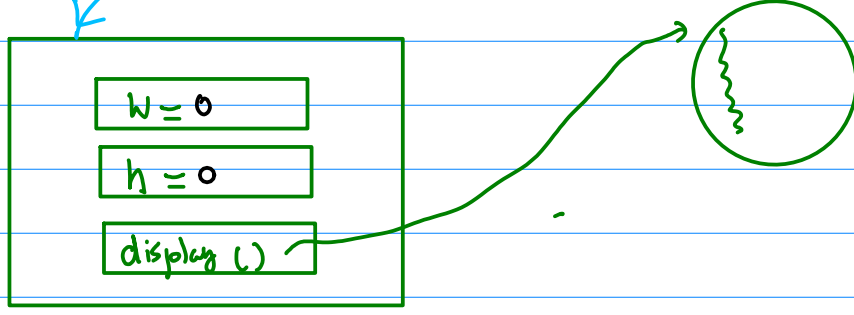
member 속성 (field)는 내가 변수를 받을 필요 없이 직접 사용 가능

Rect R1 = new Rect();

R1 = 참조변수 (어떤 객체를 가리키는)

Rect 클래스의 객체를 memory에 생성함.

생성자 함수 Rect()를 부름



```java
class Rect {
    int w;
    int h;

    Rect() { w=0; h=0; }

    int area() { return w*h; }

    void display() {
        System.out.print("w=");
        System.out.println(w);
        System.out.print("h=");
        System.out.println(h);

        int A;
        A = area();
    }
}
```

A = 20 * 30;

① 계산

A = 600 ② assign

A = area();

① A에 할당할 값을 계산

: area() 함수가 돌려주는 값

```java
class Circle {
    int r;

    Circle(     ) { r = 0; }
    Circle(int n) { r = n; }

    double area() { return 3.14*r*r; }

    void display() {
        System.out.println( "radius=" + r      );
        System.out.println( "Area="   + area() );
    };


}
```

생성자 ① no input

생성자 ② one input

```java
public class CircleTest {

    /**
     * @param args
     */
    public static void main(String[] args) {

        Circle C1 = new Circle(    );
        Circle C2 = new Circle( 20 );
        Circle C3 = new Circle( 30 );

        C1.display();
        C2.display();
        C3.display();


    }

}
```

생성자① call

생성자② call

생성자② call