

23.12.18

🕒 작성 일시	@2023년 12월 18일 오전 10:14
🏷 태그	

```
▶ public class JavaExm12_18_01 {
▶     public static void main(String[] args) {

        int[] data = {70, 60, 55, 75, 95, 90, 80, 80, 85, 100};
        int sum = 0;

        for (int i = 0; i < data.length; i++) {
            sum += data[i];
        }
        sum = sum / data.length;
        System.out.println("평균: " + sum + "점");
        if (sum < 60) {
            System.out.println("불합격");
        } else {
            System.out.println("합격");
        }
    }
}
```

JavaExm12_18_01 x

⋮

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program
평균: 79점
합격

object 객체 \Rightarrow 연동

클래스, 객체, 인스턴스, 상속
인터페이스, 다형성, 추상화 기타등등

원자사항 C, \rightarrow 순차적
코딩순서 순서대로 작동
 \rightarrow 클래스내의 코딩이 차례
따라 실행

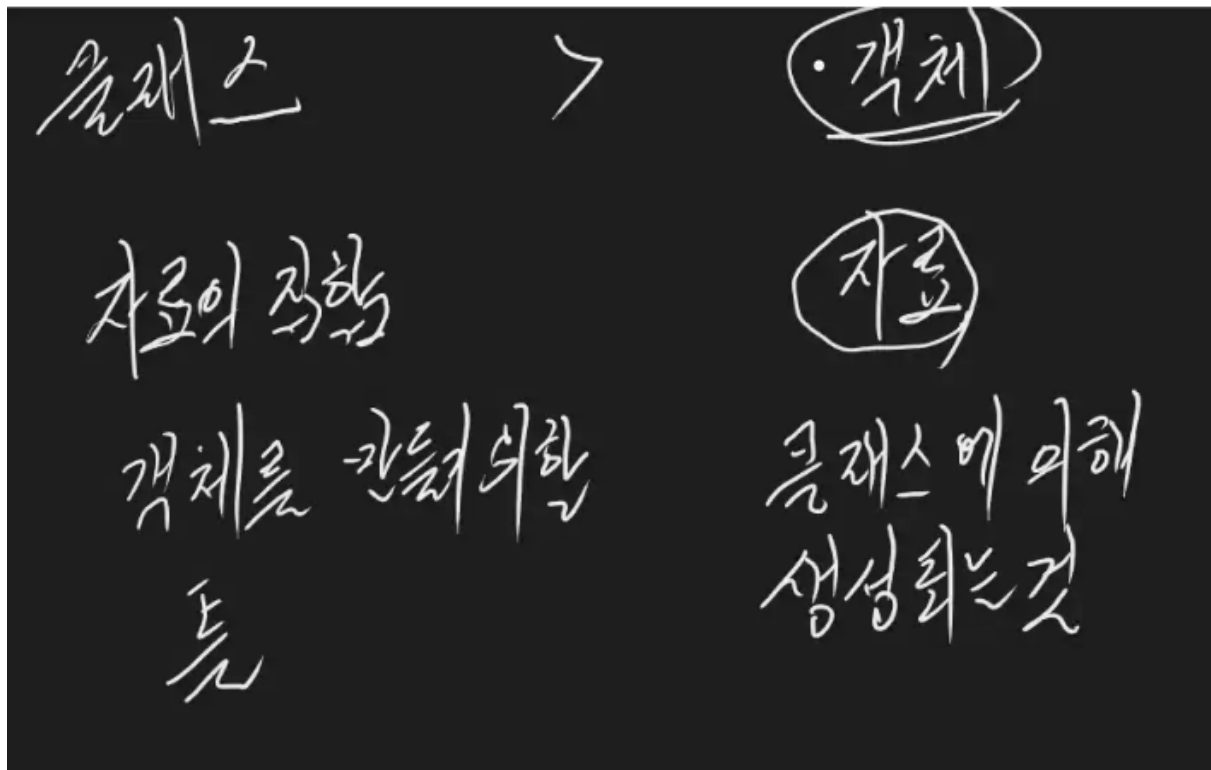
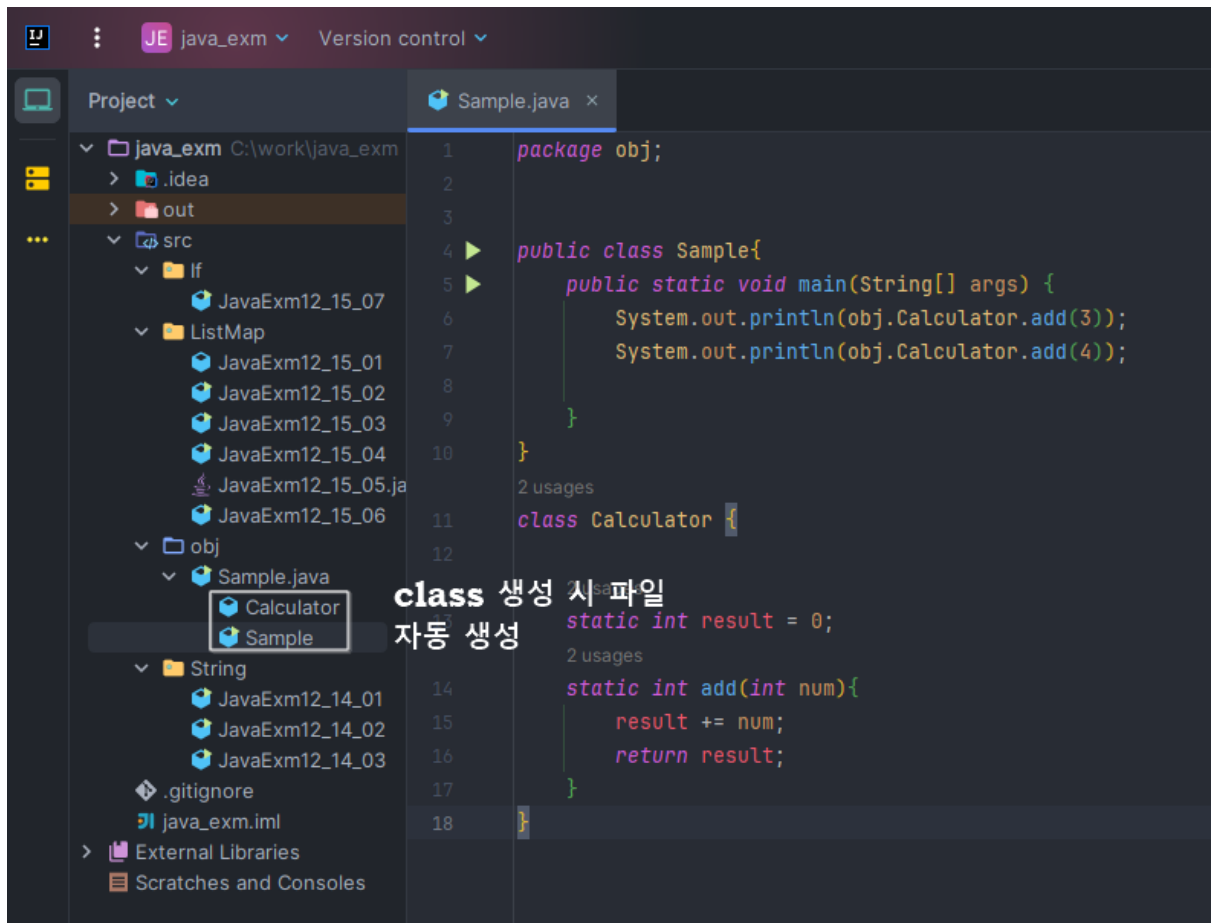
패키지 자동 연동

```

12 // 패키지 안에 있는 것들은 자동으로 연동
13 public class Sample {
14     public static void main(String[] args) {
15         System.out.println(JavaExm12_18_01.add(3));
16         System.out.println(JavaExm12_18_01.add(4));
17     }
18 }

```

public class / class



set name

set name

```
package obj;
```

8 usages

```
class Animal{
```

3 usages

```
String name;
```

2 usages

```
public void setName(String name){
```

```
    this.name =name;
```

```
}
```

```
}
```

```
public class javaExm12_18_04 {
```

```
    public static void main(String[] args) {
```

```
        Animal cat = new Animal();
```

```
        Animal dog = new Animal();
```

```
        cat.setName("baby");
```

```
        dog.setName("big");
```

```
        System.out.println(cat.name);    baby
```

```
        System.out.println(dog.name);    big
```

```
    }
```

```
}
```

메소드 생성 시
this 연산자 사용

2 usages

```
class Exm{
```

1 usage

```
int sum(int a, int b){
```

```
    return a + b;
```

```
}
```

```
}
```

```
public class JavaExm12_18_05 {
```

```
    public static void main(String[] args) {
```

```
        int a=3;
```

```
        int b=10;
```

```
        Exm hap= new Exm();
```

```
        int c = hap.sum(a, b);
```

```
        System.out.println(c);
```

```
    }
```

```
}
```

JavaExm12_18_05 x

:

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaag

13

```

package obj;

public class JavaExm12_18_06 {

    1 usage
    String say(){
        return "hello";
    }

    public static void main(String[] args) {
        JavaExm12_18_06 sam = new JavaExm12_18_06();
        String a = sam.say();
        System.out.println(a); hello
    }

}

```

메서드
 Java 함수, 메서드 공존한다. (동일 명을 쓴다) 구별시켜야 한다.
 중복 이름을 갖을 수 없다. [파이션 제외]

중복 이름을 갖을 수 없다. 파이션
 제외 → 마케대로 원유 → 기여 → 메소드 (과정 생략)
 라테 → 원유 → 기여 → 원유 → 예 결과물
 → 아 → 나으는 과정

메소드 사용이유 반복작업 리피

<pre> public class JavaExm12_18_08 { 2 usages void sayNick(String nick){ if (nick == "호구"){ return; } System.out.println("나의 별명은 "+nick+"이다."); } public static void main(String[] args) { JavaExm12_18_08 prn =new JavaExm12_18_08(); prn.sayNick("아쿠아맨"); 나의 별명은 아쿠아맨이다. prn.sayNick("호구"); } } </pre>	<pre> public class JavaExm12_18_08 { 2 usages void sayNick(String nick){ System.out.println("나의 별명은 "+nick+"이다."); } public static void main(String[] args) { JavaExm12_18_08 prn =new JavaExm12_18_08(); prn.sayNick("아쿠아맨"); prn.sayNick("호구"); } } </pre> <p style="text-align: right;">나의 별명은 아쿠아맨이다. 나의 별명은 호구이다.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

return 값이 없을때 void 사용

```

package obj;

public class JavaExm12_18_07 {

    1 usage
    void sum(int a, int b){
        System.out.println(a +"과 " + b +"의 "+"합은 " + (a+b) +" 입니다.");
    }

    public static void main(String[] args) {
        JavaExm12_18_07 sam = new JavaExm12_18_07(); //sam으로 이름 지정
        sam.sum(a: 3, b: 4);
    }
}

```

3과 4의 합은 7 입니다.

메소드의 종류(4가지)

4가지
1. 인자 호출 모두 있는
2. 인자 호출 모두 없는
3. 인자 호출 부분만 있는
4. 인자 호출 출력만 있는

```

1. int Sum(int a, int b) {
    return a + b;
}

Sum sum = new Sum();
int result = sum.happ(3, 4);

```

계산

계산기

```
package obj;

public class Calculator {
    // 더하기
    1 usage
    int add(int a, int b) {
        return a + b;
    }

    // 빼기
    1 usage
    int subtract(int a, int b) {
        return a - b;
    }

    // 결과 출력
    2 usages
    void displayResult(String operation, int result) {
        System.out.println("연산 결과: " + operation + " = " + result);
    }

    public static void main(String[] args) {
        // 클래스 생성 클래스의 인스턴스(객체지향에서 클래스 기반으로 객체를 만드는 과정)
        Calculator calculator = new Calculator();

        // 덧셈 연산
        int sumResult = calculator.add(a: 5, b: 3);
        calculator.displayResult(operation: "5 + 3", sumResult); // 연산의 종류와 결과 출력

        // 뺄셈 연산
        int subtractResult = calculator.subtract(a: 8, b: 4);
        calculator.displayResult(operation: "8 - 4", subtractResult);
    }
}
```

연산 결과: 5 + 3 = 8

연산 결과: 8 - 4 = 4