

```

package test;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
public class Test2 {

    public static void main(String[] args)
throws ClassNotFoundException, NoSuchMethodException, SecurityException, InstantiationException, IllegalAccessException, IllegalArgumentException, InvocationTargetException {

    // 클래스객체얻어오기 , 클래스 정보를 담은 객체이다
    Class clazz1 = Class.forName("test.Member");

    System.out.println("Wn=====매개변수가 없는 생성자 정보 가져오기=====Wn");
    Constructor constructor = clazz1.getDeclaredConstructor();
    //System.out.println("Constructor: " + constructor.getName());

    System.out.println("Wn=====매개변수가 없는 생성자 정보로 객체 생성하기=====Wn");
    //생성자정보를 얻어온다음 객체를 생성할 수 있다.
    Member m =(Member) constructor.newInstance();
    m.method2(5);
    System.out.println("Wn=====매개변수가 있는 생성자 정보로 객체생성하기=====Wn");
    //매개변수가 있는 생성자 정보 가져오기
    Constructor constructor2 = clazz1.getDeclaredConstructor(String.class,String.class);
    //System.out.println("Constructor(String,String): " + constructor2.getName());
    Member m2 =(Member) constructor2.newInstance("victoai","1234");
    System.out.println( m2 );

    System.out.println("Wn=====클래스정보로 부터 매서드 정보가져오기=====Wn");
    Method method2 = clazz1.getDeclaredMethod("method2", int.class);
    System.out.println( method2);
    System.out.println("Wn=====클래스정보로 부터 매서드 정보가져온 다음 호출하기=====Wn");
    //매소드를 객체를 이용하여 매소드 실제 실행, 호출, invoke
    method2.invoke( clazz1.getDeclaredConstructor().newInstance() ,6);

    System.out.println("Wn=====매소드를 객체를 이용하여 매소드 실제 호출, invoke=====Wn");
    Method method1 = clazz1.getDeclaredMethod("printlnfo");
    System.out.println( method1);
    //매소드를 객체를 이용하여 매소드 실제 실행, 호출, invoke
    method1.invoke( clazz1.getDeclaredConstructor(String.class,String.class).newInstance("hong","5678") );

    System.out.println("Wn=====클래스객체로 부터 필드정보 얻어오기=====Wn");
    Field fields[] = clazz1.getDeclaredFields();
    for (Field field : fields) {
        System.out.println( field);
    }

}

}

```