package AnnotationAssignments;

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

import java.lang.reflect.InvocationTargetException;

import java.lang.reflect.Method;

public class Annotation1 {

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

TestClass t = new TestClass();

Method methods = t.getClass().getMethod("testMethod");

Test name = methods.getAnnotation(Test.class);

methods.invoke(t);

System.out.println(name.testCase());

}

}

@Target(ElementType.METHOD)

@Retention(RetentionPolicy.RUNTIME)

@interface Test{

String testCase() ;

}

class TestClass{

@Test(testCase = "Having Annotation Test")

public void testMethod()

{

System.out.println("Method is of type test Case");

}

private void tempmethod() {

System.out.println("Hello");

}

}