調用指定的屬性:

*/\*  
如何操作运行时类中的指定的属性 -- 需要掌握  
 \*/*@Test  
public void testField1*()* throws Exception *{* Class clazz = Person.class;  
 *//创建运行时类的对象* Person p = *(*Person*)* clazz.newInstance*()*;  
 *//1. getDeclaredField(String fieldName):获取运行时类中指定变量名的属性* Field name = clazz.getDeclaredField*(*"name"*)*;  
 *//2.保证当前属性是可访问的* name.setAccessible*(*true*)*;  
 *//3.获取、设置指定对象的此属性值* name.set*(*p,"Tom"*)*;  
 System.*out*.println*(*name.get*(*p*))*;  
*}*

調用指定的方法:

*/\*  
 如何操作运行时类中的指定的方法 -- 需要掌握  
 \*/* @Test  
 public void testMethod*()* throws Exception *{* Class clazz = Person.class;  
 *//创建运行时类的对象* Person p = *(*Person*)* clazz.newInstance*()*;  
 */\*  
 1.获取指定的某个方法  
 getDeclaredMethod():参数1 ：指明获取的方法的名称 参数2：指明获取的方法的形参列表  
 \*/* Method show = clazz.getDeclaredMethod*(*"show", String.class*)*;  
 *//2.保证当前方法是可访问的* show.setAccessible*(*true*)*;  
 */\*  
 3. 调用方法的invoke():参数1：方法的调用者 参数2：给方法形参赋值的实参  
 invoke()的返回值即为对应类中调用的方法的返回值。  
 \*/* Object returnValue = show.invoke*(*p,"CHN"*)*; *//String nation = p.show("CHN");* System.*out*.println*(*returnValue*)*;  
  
 System.*out*.println*(*"\*\*\*\*\*\*\*\*\*\*\*\*\*如何调用静态方法\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"*)*;  
 *// private static void showDesc()* Method showDesc = clazz.getDeclaredMethod*(*"showDesc"*)*;  
 showDesc.setAccessible*(*true*)*;  
 *//如果调用的运行时类中的方法没有返回值，则此invoke()返回null  
// Object returnVal = showDesc.invoke(null); //靜態方法已經知道是哪個類,所以可以寫null效果一樣*

Object returnVal = showDesc.invoke*(*Person.class*)*;//靜態方法要用類來調用  
 System.*out*.println*(*returnVal*)*;*//null  
  
 }*

調用指定的構造器:

*/\*  
如何调用运行时类中的指定的构造器  
 \*/*@Test  
public void testConstructor*()* throws Exception *{* Class clazz = Person.class;  
 *//private Person(String name)  
 /\*  
 1.获取指定的构造器  
 getDeclaredConstructor():参数：指明构造器的参数列表  
 \*/* Constructor constructor = clazz.getDeclaredConstructor*(*String.class*)*;  
 *//2.保证此构造器是可访问的* constructor.setAccessible*(*true*)*;  
 *//3.调用此构造器创建运行时类的对象* Person per = *(*Person*)* constructor.newInstance*(*"Tom"*)*;  
 System.*out*.println*(*per*)*;  
*}*