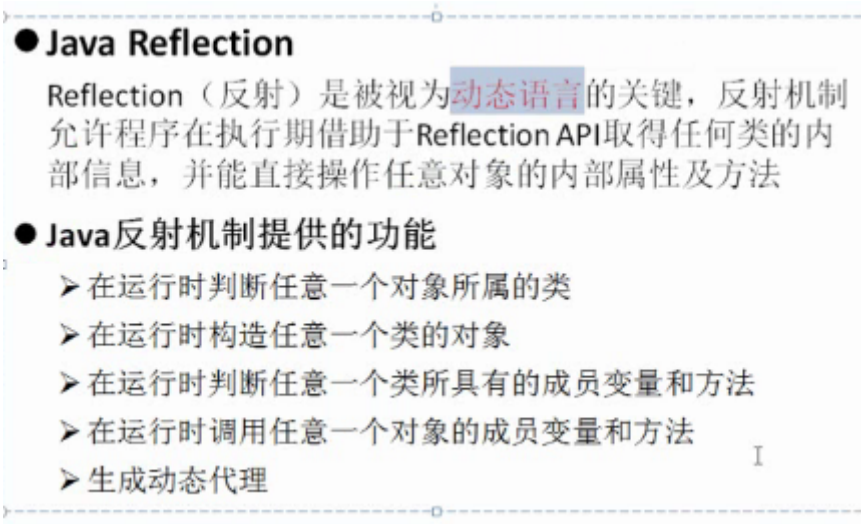

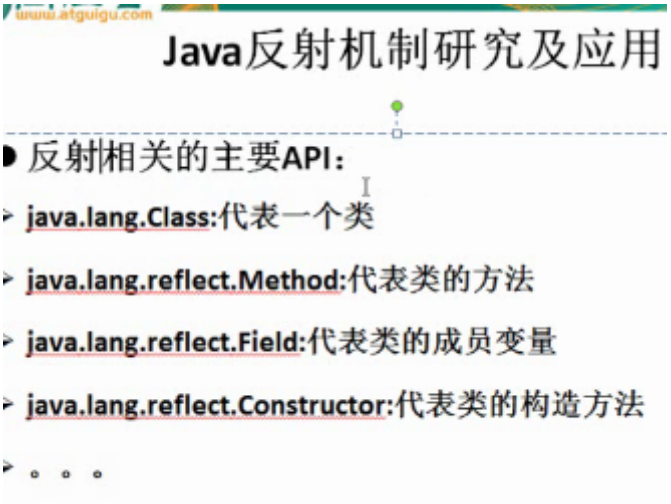

在有反射以前，如果创建一个类的对象，并调用其中的方法和属性

实例：

```
package thread;

import java.lang.reflect.Field;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;

import org.junit.Test;

public class testfanshe {
```

```
	@Test
	public void test2() throws NoSuchFieldException, SecurityException,
InstantiationException, IllegalAccessException, NoSuchMethodException,
IllegalArgumentException, InvocationTargetException{
	//有了反射，可以通过反射创建一个类的对象，并调用其中的结构
		//1,创建class1对应的运行时类fanshe类的对象
		Class class1 = fanshe.class;
		System.out.println("反射调用属性：");
			fanshe fanshe = (fanshe)class1.newInstance();
			System.out.println(fanshe);
			Field field = class1.getDeclaredField("nameString");
			field.set(fanshe, "linmingjun");
			System.out.println(fanshe);
			Field field2 = class1.getDeclaredField("age");
			field2.setAccessible(true);
			field2.set(fanshe, 88);
			System.out.println(fanshe);
			System.out.println("反射调用方法：");
			Method m1 = class1.getMethod("show",String.class);
			m1.invoke(fanshe,"美国");

	}
@Test
public void test1(){
	//在有反射之前，如何创建一个对象，并调用其中的方法和属性
	fanshe fanshe = new fanshe("anqili", 21, "信科16-2");
	fanshe.setAge(32);
	System.out.println(fanshe.getAge());
	System.out.println(fanshe);
}
}
```