

【案例】调用其他类的set和get方法

```
class hello {
    public static void main(String[] args) {
        Class<?> demo = null;
        Object obj=null;
        try {
            demo = Class.forName("Reflect.Person");
        } catch (Exception e) {
            e.printStackTrace();
        }
        try{
            obj = demo.newInstance();
        }catch (Exception e) {
            e.printStackTrace();
        }
        setter(obj, "Sex", "男", String.class);
        getter(obj, "Sex");
    }

    /**
     * @param obj 操作的对象
     * @param att 操作的属性
     */
    public static void getter(Object obj, String att) {
        try {
            Method method = obj.getClass().getMethod("get" + att);
            System.out.println(method.invoke(obj));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * @param obj 操作的对象
     * @param att 操作的属性
     * @param value 设置的值
     * @param type 参数的属性
     */
    public static void setter(Object obj, String att, Object value,
        Class<?> type) {
        try {
            Method method = obj.getClass().getMethod("set" + att, type);
            method.invoke(obj, value);
        }
    }
}
```

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
} // end class
```

【运行结果】：

男

【案例】通过反射操作属性

```
class hello {  
    public static void main(String[] args) throws Exception {  
        Class<?> demo = null;  
        Object obj = null;  
  
        demo = Class.forName("Reflect.Person");  
        obj = demo.newInstance();  
  
        Field field = demo.getDeclaredField("sex");  
        //启用或禁用安全检查,值为 true 则指示反射的对象在使用时  
        //应该取消 Java 语言访问检查。值为 false 则指示反射的对象  
        //应该实施 Java 语言访问检查。  
        field.setAccessible(true);  
        field.set(obj, "男");  
        System.out.println(field.get(obj));  
    }  
} // end class
```