

【案例】接下来让我们取得其他类的全部属性吧，最后我讲这些整理在一起，也就是通过class取得一个类的全部框架

```
class hello {
    public static void main(String[] args) {
        Class<?> demo = null;
        try {
            demo = Class.forName("Reflect.Person");
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println("=====本类属性=====");
        // 取得本类所有的属性,但是不包括父类的声明字段
        Field[] field = demo.getDeclaredFields();
        for (int i = 0; i < field.length; i++) {
            // 权限修饰符
            int mo = field[i].getModifiers();
            String priv = Modifier.toString(mo);
            // 属性类型
            Class<?> type = field[i].getType();
            System.out.println(priv + " " + type.getName() + " "
                + field[i].getName() + ";");
        }
        System.out.println("=====实现的接口或者父类的属性=====");
        // 取得public修饰的属性
        Field[] filed1 = demo.getFields();
        for (int j = 0; j < filed1.length; j++) {
            // 权限修饰符
            int mo = filed1[j].getModifiers();
            String priv = Modifier.toString(mo);
            // 属性类型
            Class<?> type = filed1[j].getType();
            System.out.println(priv + " " + type.getName() + " "
                + filed1[j].getName() + ";");
        }
    }
}
```

【运行结果】：

```
=====本类属性=====
private java.lang.String sex;
=====实现的接口或者父类的属性=====
public static final java.lang.String name;
public static final int age;
```

【案例】其实还可以通过反射调用其他类中的方法：

```
class hello {  
    public static void main(String[] args) {  
        Class<?> demo = null;  
        try {  
            demo = Class.forName("Reflect.Person");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        try{  
            //调用Person类中的sayChina方法  
            Method method = demo.getMethod("sayChina");  
            method.invoke(demo.newInstance());  
            //调用Person的sayHello方法  
            method = demo.getMethod("sayHello", String.class,int.class);  
            method.invoke(demo.newInstance(),"Rollen",20);  
        }catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

【运行结果】：

```
hello ,china  
Rollen 20
```