

转自: <http://www.cnblogs.com/rollenholt/archive/2011/09/02/2163758.html>

【案例】

返回一个类实现的接口:

```
package Reflect;
interface China{
    public static final String name="Rollen";
    public static int age=20;
    public void sayChina();
    public void sayHello(String name, int age);
}

class Person implements China{
    public Person() { }
    public Person(String sex){
        this.sex=sex;
    }
    public String getSex() {
        return sex;
    }
    public void setSex(String sex) {
        this.sex = sex;
    }
    @Override
    public void sayChina(){
        System.out.println("hello ,china");
    }
    @Override
    public void sayHello(String name, int age){
        System.out.println(name+" "+age);
    }
    private String sex;
}

class hello{
    public static void main(String[] args) {
        Class<?> demo=null;
        try{
            demo=Class.forName("Reflect.Person");
        }catch (Exception e) {
            e.printStackTrace();
        }
        //保存所有的接口
        Class<?> intes[]=demo.getInterfaces();
    }
}
```

```

        for (int i = 0; i < intes.length; i++) {
            System.out.println("实现的接口    "+intes[i].getName());
            //~out : Reflect.China
        }
    }
}

```

（注意，以下几个例子，都会用到这个例子的Person类，所以为节省篇幅，此处不再粘贴Person的代码部分，只粘贴主类hello的代码）

• 【案例】：取得其他类中的父类

```

class hello{
    public static void main(String[] args) {
        Class<?> demo=null;
        try{
            demo=Class.forName("Reflect.Person");
        }catch (Exception e) {
            e.printStackTrace();
        }
        //取得父类
        Class<?> temp = demo.getSuperclass();
        System.out.println("继承的父类为：    "+temp.getName());
        //~out : 继承的父类为：    java.lang.Object
    }
}

```