



链滴

关于 Return 在 Try-Catch 中顺序问题

作者: [kevin2020](#)

原文链接: <https://ld246.com/article/1513611828931>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

之前也一直没有特别在意，今天正好发现了这个问题
彻底理清顺序问题，先上代码，最下面是结论

```
public class TryCatchTest {
    public static String testReturn(String str) {
        try {
            if (str.equals("正常结束")) {
                return str;          //第一句返回
            }
            if (str.equals("抛出异常")) {
                throw new Exception();
            }
            return str;          //三四句返回
        } catch (Exception e) {
            //      System.out.println("catch");
            //      throw new NullPointerException();
            str="出错了";          //这里要对接下来的返回对象做些处理，确保返回的对象不会影响接下
            的程序，保证健壮性。
        }
        //      finally{
        //          System.out.println("finally");
        //          return "不管怎样都会返回我";
        //      }

        return str+str;          //第二句返回
    }

    public static void main(String[] args) {
        System.out.println(TryCatchTest.testReturn("正常结束")); //返回正常结束
        System.out.println("=====");
        System.out.println(TryCatchTest.testReturn("抛出异常")); //出错了出错了
        System.out.println("=====");
        System.out.println(TryCatchTest.testReturn("A")); //A
        System.out.println("=====");
        System.out.println(TryCatchTest.testReturn("B")); //B
    }
}
```

相信不进catch块都没有悬念，主要是在catch块，finally中存在着一先盲点

1 抛出异常后 会直接进入catch块后 返回try-catch块外的return 所以要注意return的对象是否影响
续程序运行，如果影响，catch中可做适当处理

2 任何执行try 或者catch中的return语句之前，都会先执行finally语句，这点大家都知道。

所以如果finally中有return语句，那么程序就return了，**finally中的return是一定会被return的。**

编译器把finally中的return实现为一个warning，也就是说如果finally中存在return 相当于其余return
失效。

3 由于2，所以即时catch中throw了，在throw前也会先进行finally 如果finally中return，则catch中
常也无法抛出

4 如果在catch中return，finally中对return对象进行改变：基本数据类型的改变无效，对引用改变有
。（引用传递和值传递的区别）

简单例子 伪代码 `int i;try (return i=1;) finally {i=2} ; return`结果为1

如有疏漏之处，欢迎指出

PS：现在的工具也是很智能，catch写了if判断对象是否为空，提示Dead Code警告，才发现最开始经new过了