

try、catch、finally代码块中处理 return 时返回的不同结果

=====

> 注：无论 try 子句中执行了`return`、`break` 还是 `continue` 语句，或是程序出现异常，finally 子句块都会被执行`。除非直接调用`System.exit()` 来终止程序。

来看这样一段有意思的代码~

```
...
public class Demo_try_catch_finally {
    public static int get(){
        int x = 1;
        try{
            x = 2;
            return x;
        }catch (Exception e){
            x = 3;
            return x;
        }finally {
            x = 4;
        }
    }
    public static void main(String[] args) {
        System.out.println(get());
    }
}
...
```

它的返回结果是多少呢？ 4？ 3？ 2？ 1？ 总之，总会有一个是对的！

咱们来 debug 一下，你会发现第一次执行的是`x=2, return x;` 这里没有问题，我们按`F7`查看程序执行的下一步。

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/293f923cc25a43a9b0de546d0e523aef~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=587&h=472&s=356936&e=png&b=375b5f)

程序并没有跟我们想象的那样，执行return语句后，直接返回。如下图所示，它直接去执行finally中的代码。

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/aaff5a64f2074b44996a580167cd3c18~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=589&h=458&s=350191&e=png&b=375b5f)

执行完finally中的代码后，你会发现debug又回到了try代码块中，并且这时候x是等于4的。感觉没毛病了，大家肯定都会觉得返回结果一定是4没错了，收工~

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/c5182afbdd7243189b4e90f097975876~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=664&h=462&s=396488&e=png&b=36585d)

看看控制台，直接裂开!!! 居然是2!!!

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/57f4cd244928409a932073d110689180~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=778&h=187&s=190505&e=png&b=56495e)

****为什么程序的返回结果是2而不是4呢? ****

来看看官方的[JVM规范](http://cxyroad.com/"https://docs.oracle.com/javase/specs/jvms/se7/html/jvms-4.html#jvms-4.10.2.5")是如何解释的:

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/691b6db0379c4398bc2fe2759d650964~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1333&h=676&s=90247&e=png&b=fefdf d)

翻译成中文大概意思如下:

- > 若try子句执行return语句，编译后的代码将执行以下操作：
- >
- >
- > * 将返回值保存到局部变量中；
- > * 执行jsr指令转移到执行finally子句；
- > * 执行完finally子句返回时，返回保存在局部变量中的值。

原文链接: <https://juejin.cn/post/7367298026655989787>