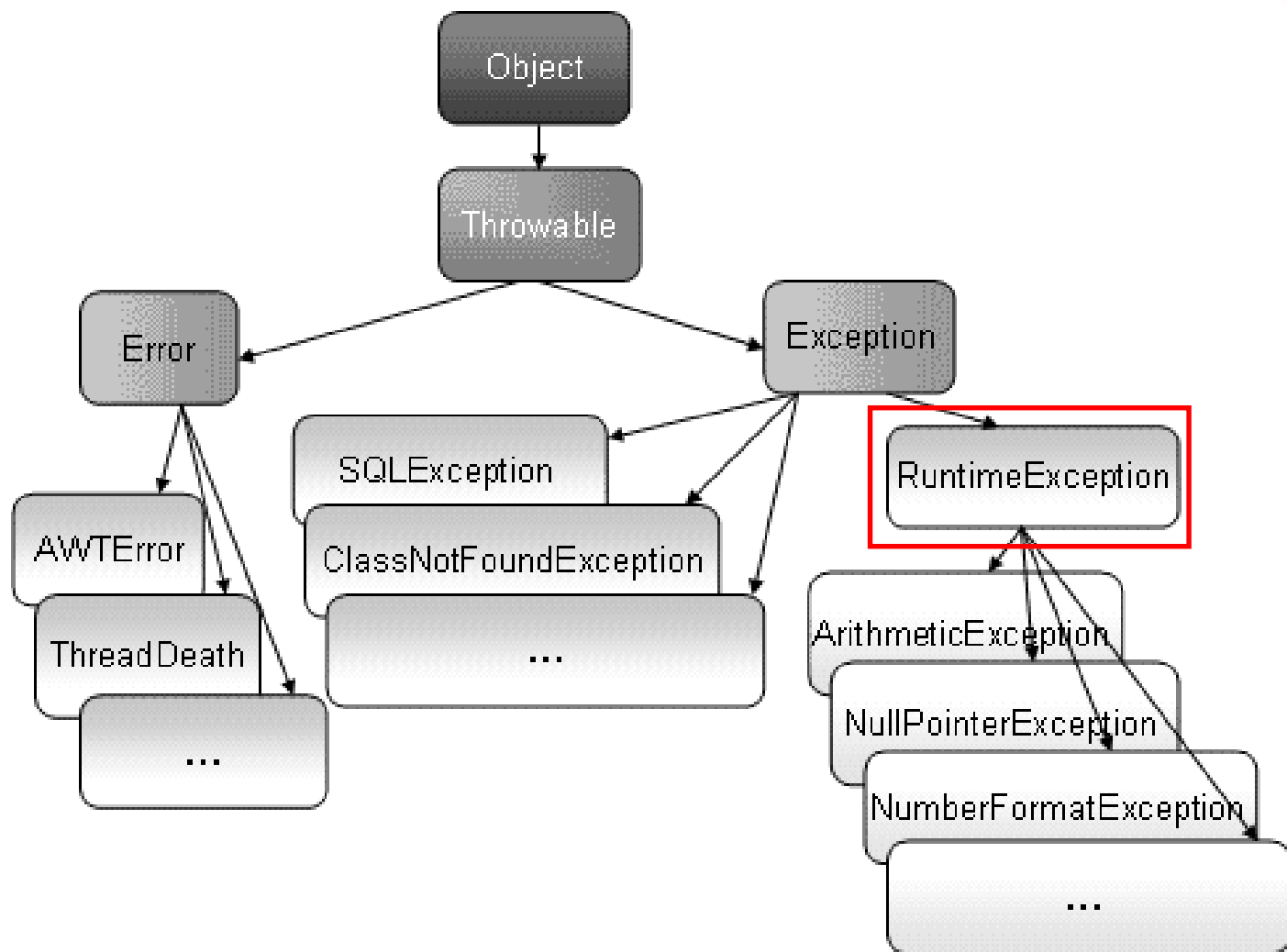


第十二章 异常处理与程序调试

异常的分类



常见的异常类型



Exception

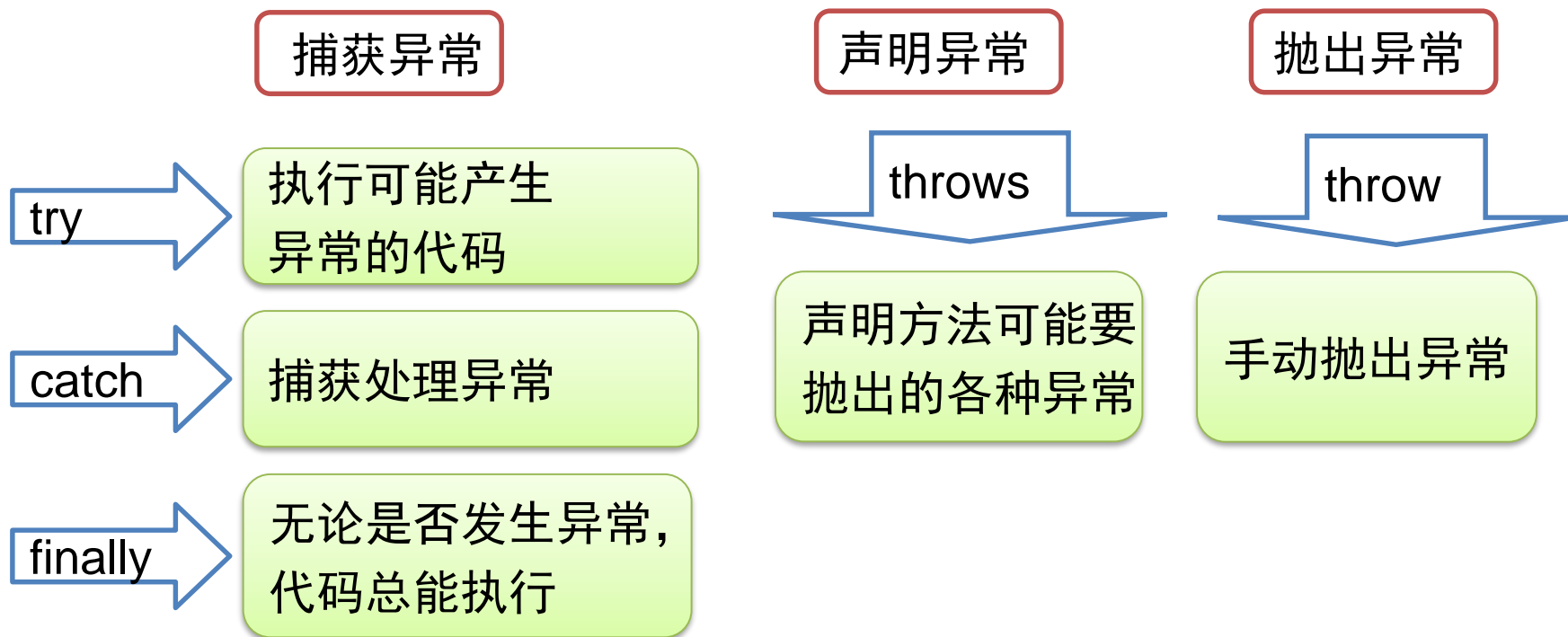
异常层次结构的根类

方 法 名	说 明
ArithmeticException	算术错误情形，如以零作除数
ArrayIndexOutOfBoundsException	数组下标越界
NullPointerException	尝试访问 null 对象成员
ClassNotFoundException	不能加载所需的类
InputMismatchException	欲得到数据类型与实际输入类型不匹配
IllegalArgumentException	方法接收到非法参数
ClassCastException	对象强制类型转换出错
NumberFormatException	数字格式转换异常，如把"abc"转换成数字

异常处理



- Java的异常处理是通过5个关键字来实现的：`try`、`catch`、`finally`、`throw`、`throws`

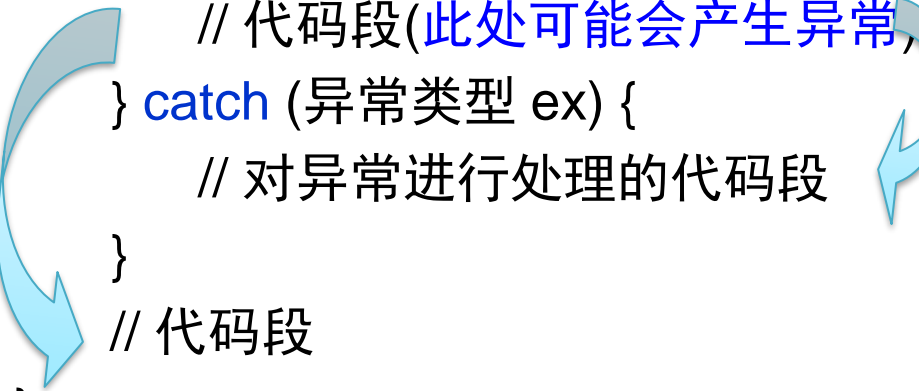


try-catch块



□ 基本语法：

```
public void method(){  
    try {  
        // 代码段(此处可能会产生异常)  
    } catch (异常类型 ex) {  
        // 对异常进行处理的代码段  
    }  
    // 代码段  
}
```

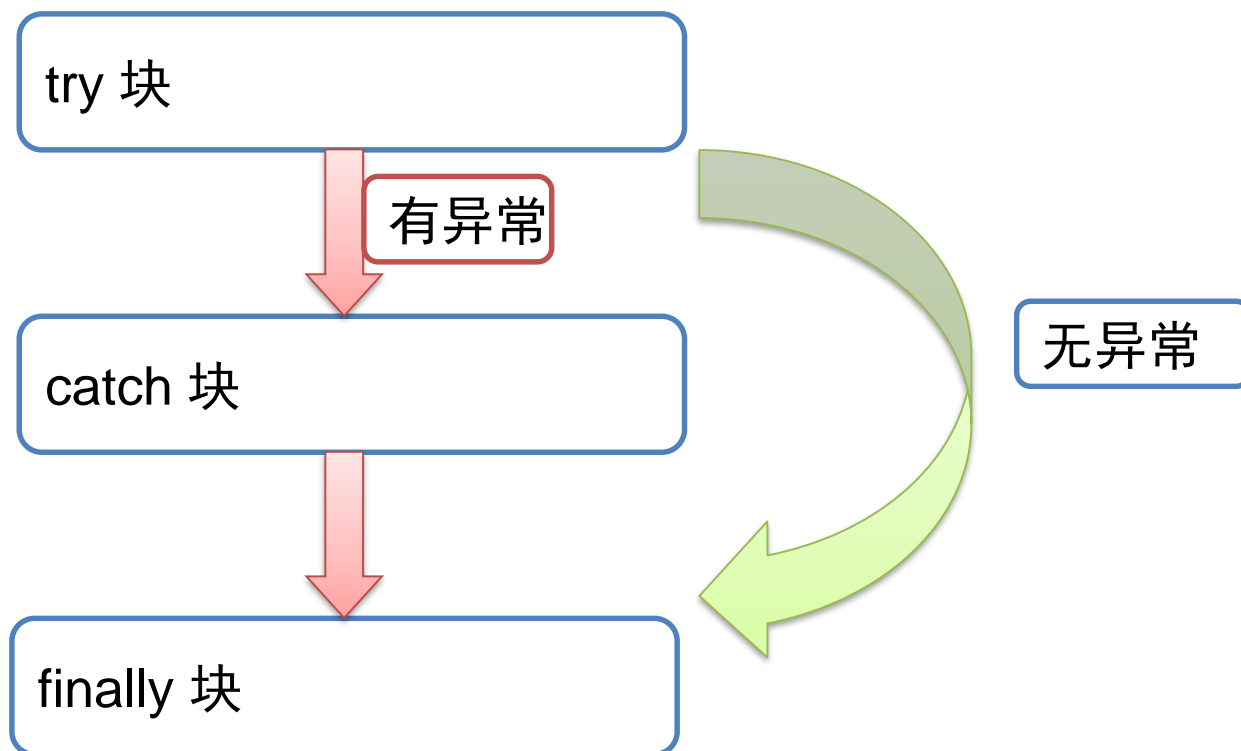
A diagram illustrating the execution flow of a try-catch block. A large blue arrow on the left side of the code block points from the 'try' block down to the 'catch' block, indicating that if an exception occurs, control jumps to the catch block. A smaller blue arrow on the right side points from the 'catch' block back to the 'try' block, indicating that after handling the exception, control returns to the point after the try block.



- 一段代码可能会引发多种类型的异常

```
public void method(){  
    try {  
        // 代码段  
        // 产生异常(异常类型2)  
    } catch (异常类型1 ex) {  
        // 对异常进行处理的代码段  
    } catch (异常类型2 ex) {  
        // 对异常进行处理的代码段  
    } catch (异常类型3 ex) {  
        // 对异常进行处理的代码段  
    }  
    // 代码段  
}
```

try-catch-finally





- ❑ throws
- ❑ Throw
- ❑ 自定义异常



- ❑ 只用于处理非正常的情况；
- ❑ 避免过大的try
- ❑ 使用多重catch
- ❑ 不要忽略catch块儿中的异常
- ❑ 改正代码
- ❑ 文档声明

如何进行程序调试



❑ 步骤1：分析错误，设置断点

❑ 步骤2：启动调试

❑ 步骤3：单步运行

F5：单步跳入

F6：单步跳过

❑ 步骤4：观察变量

❑ 步骤5：发现问题

❑ 步骤6：修正代码，重新运行

❑ 步骤7：解决问题