

# 异常

try-catch-finally

```
try{
    //可能会抛出异常的代码
}catch(Exception e){
    //对异常进行处理
}finally{
    //一定会执行的代码
}
```

## 异常类

- Error：系统错误，程序无法处理。
- Exception：程序运行时出现的错误，程序可以处理。

Error 和 Exception 都是 Throwable 的子类，Throwable、Error、Exception 都是存放在 java.lang 包中。

Error 常见的子类：VirtualMachineError、AWTError、IOException。

VirtualMachineError 的子类：StackOverflowError、OutOfMemoryError，用来描述内存溢出等系统问题。

Exception 常见的子类：IOException 和 RuntimeException

IOException 存放在 java.io 包中，RuntimeException 存放在 java.lang 包中。

IOException 常见的子类：FileLockInterruptedException、FileNotFoundException、FileNotFoundException，这些异常通常都是处理通过 IO 流进行文件传输的时候发生的错误。

RuntimeException 常见的子类：

- ArithmeticException：表示数学运算异常。
- ClassNotFoundException：表示类未定义异常。
- IllegalArgumentException：表示参数格式错误。
- ArrayIndexOutOfBoundsException：表示数组下标越界。
- NullPointerException：表示空指针异常。
- NoSuchMethodException：表示方法未定义异常。
- NumberFormatException：表示将其他数据类型转为数值类型发生的类型不匹配异常。

## throw 和 throws

throw 和 throws 是 Java 在处理异常时使用的两个关键字，都可以用来抛出异常，但是使用的方式和表示的含义完全不同。

Java 中抛出异常有 3 种方式：

- try-catch
- 使用 throw 是开发者主动抛出异常，即读到 throw 代码就一定抛出异常，基本语法：throw new Exception()，是一种基于代码的逻辑而主动抛出异常的方式。

```
public class Test {
    public static void main(String[] args) {
        int[] array = {1,2,3};
        test(array,2);
    }

    public static void test(int[] array,int index) {
        if(index >= 3 || index < 0) {
            try {
                throw new Exception();
            } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } else {
            System.out.println(array[index]);
        }
    }
}
```

- try-catch 和 throw 都是作用于具体的逻辑代码，throws 是作用于方法的，用来描述方法可能会抛出的异常。

如果方法 throws 的是 RuntimeException 异常或者其子类，外部调用时可以不处理，JVM 会处理。

如果方法 throws 的是 Exception 异常或者其子类，外部调用时必须处理，否则报错。

```
public class Test {
    public static void main(String[] args) throws Exception {
        test("123");
    }

    public static void test(String str) throws Exception {
        int num = Integer.parseInt(str);
    }
}
```