

# 异常处理

## 异常分类



图1 Java异常类层次结构图

## 系统错误-Error

Java运行系统中的内部错误，这些错误发生时，JVM一般会终止线程。

## 异常——Exception

### 1. 运行时异常——RuntimeException

运行过程中产生的异常，一般有程序逻辑错误引起的，应从逻辑角度避免这样子的错误。

### 2. 非运行时异常

必须进行处理的异常，如果不处理，程序就不能编译通过。

## 捕捉处理异常

```
try{
    //程序代码块
}
catch(Exception e) {
    //对exception处理
}
finally{
    //代码块
}
```

其中finally代码块是异常处理结构最后执行部分，无论程序是否发生异常，finally代码块中的代码都将执行，因此，在finally代码块是通常防止释放资源、关闭对象的代码。

## 处理异常的方法

```
getMessage(): 获取有关异常世家的信息
toString(): 获取异常类型与性质
printStackTrace (): 获取异常事件发生时执行的堆栈内容
```

当使用多个catch块时，异常类顺序是先子类后父类，否则，可能导致后面子类的异常由父类捕获而得不到详细的信息。

## 在方法中抛出异常

如果某个方法可能会发生异常，但不想在当前方法中处理这个异常，则可以使用throws、throw关键字在方法中抛出异常。

## 使用throws关键字抛出异常

throws关键字通常应用在声明方法时，用来指定方法可能抛出的异常，多个异常可使用逗号分割。

使用throws为方法抛出异常时，如果子类继承父类，子类重写方法抛出异常也要和原父类方法抛出的异常相同或是其异常的子类，除非throws异常是RuntimeException。

如果方法抛出了异常，在调用该方法时，必须为捕捉的方法处理异常，当然，如果使用throws关键字将异常抛给上一级后，不想处理该异常，可以继续向上抛出，但最终要有能够处理该异常的代码。

## 使用throw关键字抛出异常

throw关键字通常用于在方法中制造一个异常，程序在执行到throw语句时立即终止，它后面的语句都不执行。使用throw关键字抛出异常

```
throw new 异常类型名 (异常信息)
```

throw通常用来抛出用户自定义异常。