# Throwable 类

## 概述

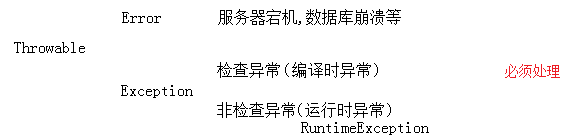
Throwable 类是 Java 语言中所有错误或异常的超类

## 分类（子类）

Throwable有两个直接子类，它们是：

Error类：Error类的异常通常为内部错误，因此在正常情况下并不期望用户程序捕获它们；

Exception类：绝大部分用户程序应当捕获的异常类的根类；一些常用的异常类都直接或间接派生自Exception类，因此我们可以认为绝大部分的异常都属于Exception。

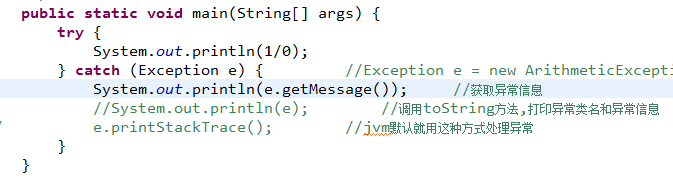


## 常见方法

getMessage() 获取异常信息，返回字符串。

toString() 获取异常类名和异常信息，返回字符串。

printStackTrace() 获取异常类名和异常信息，以及异常出现在程序中的位置。返回值void。



分别打印日志







# 异常

## 概述

异常就是Java程序在运行过程中出现的错误。异常可分为两大类：编译时异常和运行时异常；

编译时异常一般是指语法错误，可以通过编译器的提示加以修正，这里我们不予讨论；

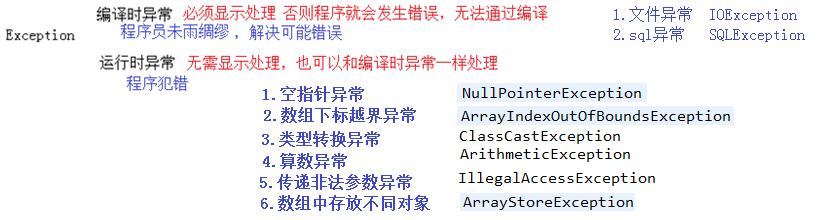
运行时异常包括：运行错误：如数组下标越界，除数为0等；逻辑错误：如年龄超过200岁等。

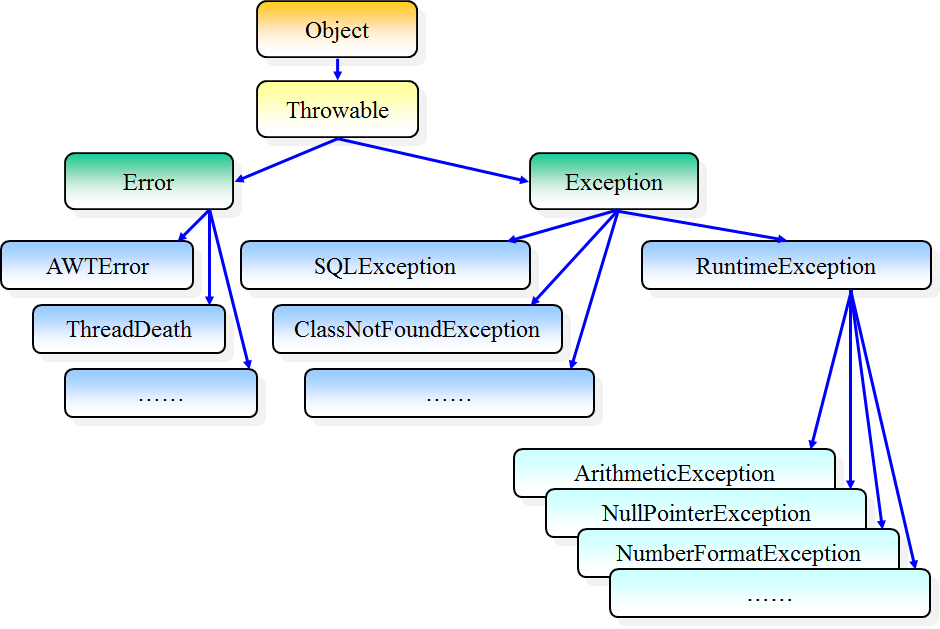
## 构造方法

|  |
| --- |
| [**Exception**](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/Exception.html#Exception())()            构造详细消息为 null 的新异常。 |
| [**Exception**](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/Exception.html#Exception(java.lang.String))([String](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) message)            构造带指定详细消息的新异常。 |
| [**Exception**](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/Exception.html#Exception(java.lang.String, java.lang.Throwable))([String](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) message, [Throwable](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/Throwable.html) cause)            构造带指定详细消息和原因的新异常。 |
| [**Exception**](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/Exception.html#Exception(java.lang.Throwable))([Throwable](mk:@MSITStore:F:\a桌面数据\参考API\jdk\JDK_API_1_6_zh_CN.CHM::/java/lang/Throwable.html) cause)            根据指定的原因和 (cause==null ? null : cause.toString()) 的详细消息构造新异常（它通常包含 cause 的类和详细消息）。 |

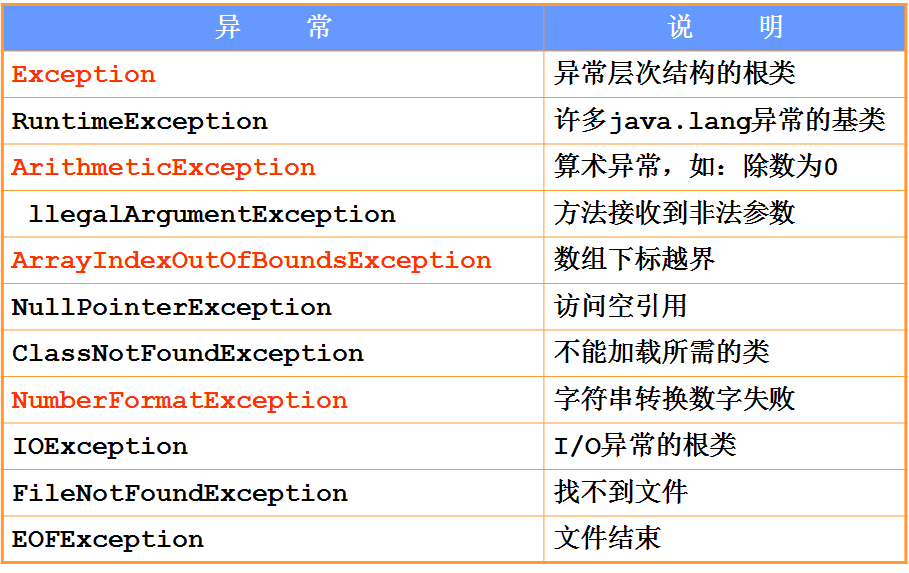
## 分类（子类）

所有的RuntimeException类及其子类的实例被称为运行时异常，其他的异常就是编译时异常





## 常见异常



# 异常处理

## 产生异常后的反应

当异常发生时，程序一般会作出如下反应：

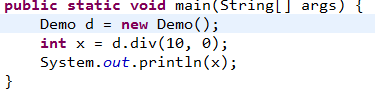
发生异常的部分产生系统定义的错误信息；程序意外终止，并将控制权返回操作系统；程序中所有已分配资源的状态保持不变，这样将会导致资源泄漏。那么我们就必须对有可能产生的异常进行处理。

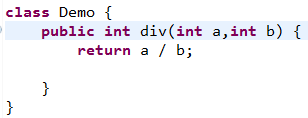
## JVM默认是如何处理异常的

main函数收到这个问题时,有两种处理方式:

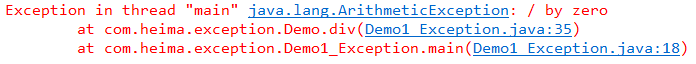
a:自己将该问题处理,然后继续运行

b:如果自己没有针对的处理方式,只有交给调用main的jvm来处理；jvm有一个默认的异常处理机制,就将该异常进行处理.并将该异常的名称,异常的信息.异常出现的位置打印在了控制台上,同时将程序停止运行





由于自己没有处理，jvm处理，🡪jvm打印信息



## 注意事项

1.子类重写父类方法时，子类的方法必须抛出相同的异常或父类异常的子类。(父亲坏了,儿子不能比父亲更坏)

2.如果父类抛出了多个异常,子类重写父类时,只能抛出相同的异常或者是他的子集,子类不能抛出父类没有的异常

3.如果被重写的方法没有异常抛出,那么子类的方法绝对不可以抛出异常,如果子类方法内有异常发生,那么子类只能try,不能throws