

异常的处理机制:

1) 当程序在运行过程中出现了异常, JVM自动创建一个该类型的异常对象。同时把这个异常对象交给运行时系统。(抛出异常)

2) 运行时系统接受到一个异常对象时, 它会再产生异常的代码附近查找相应的处理方式。

3) 异常的处理方式有两种:

捕获并处理: 在异常的代码附近显示用try/catch进行处理(不合理), 运行时系统捕获后会查询相应的catch处理块, 在catch处理块中对该异常进行处理。

用throws向外声明(合理的处理方法): 查看发生异常的方法是否有向上声明异常, 有向上声明, 向上级查询处理语句, 如果没有向上声明, JVM中断程序的运行并处理。

ArithmeticException异常是继承RuntimeException运行时异常。

Java里面异常分为两大类: checked exception(检查异常)和unchecked exception(未检查异常), 对于未检查异常也叫RuntimeException(运行时异常), 对于运行时异常, java编译器不要求你一定要把它捕获或者一定要继续抛出, 但是对checked exception(检查异常)要求你必须要在方法里面或者捕获或者继续抛出。

对未检查异常(unchecked exception), 包括RuntimeException与其子类, 以及错误(Error)的几种处理方式:

- 捕获
- 继续抛出
- 不处理

对检查异常(checked exception, 除了RuntimeException与其子类, 以及错误(Error), 其他的都是检查异常), 用throw关键字来手动抛出一个异常, 如果手动抛出的异常是一个检查异常, 那么必须进行处理, 处理方式有两种:

- 显示用try/catch进行处理(不合理)
- 用throws向外声明(合理的处理方法)

注意, 对于检查的异常必须处理, 或者必须捕获或者必须抛出

自定义异常：当需要一些跟特定业务相关的异常信息类时。

可以继承Exception来定义一个受检异常。也可以继承自RuntimeException或其子类来定义一个非受检异常。

Exception异常进行划分，它可分为运行时异常和非运行时异常。

什么是运行时异常？

都是RuntimeException类及其子类异常，如NullPointerException(空指针异常)、IndexOutOfBoundsException(下标越界异常)等，这些异常是非检查异常，程序中可以选择捕获处理，也可以不处理。这些异常一般是由程序逻辑错误引起的，程序应该从逻辑角度尽可能避免这类异常的发生。

运行时异常的特点是Java编译器不会检查它，也就是说，当程序中可能出现这类异常，即使没有用try-catch语句捕获它，也没有用throws子句声明抛出它，也会编译通过。

什么是非运行时异常？

是RuntimeException以外的异常，类型上都属于Exception类及其子类。从程序语法角度讲是必须进行处理的异常，如果不处理，程序就不能编译通过。如IOException、SQLException等以及用户自定义的Exception异常，一般情况下不要自定义检查异常。