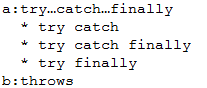
# 处理异常（两种方式）

## 处理异常（两种方式）



## 如何使用异常处理

### 原则

如果该功能内部可以将问题处理，用try；如果处理不了，交由调用者处理,这是用throws，如果JDK没有提供对应的异常，需要自定义异常。

### 区别

后续程序需要继续运行就try

后续程序不需要继续运行就throws

## throws和throw的区别

throws的方式处理异常

\* 定义功能方法时，需要把出现的问题暴露出来让调用者去处理。

\*用在方法声明后面，跟的是异常类名，可以跟多个异常类名，用逗号隔开，那么就通过throws在方法上标识。

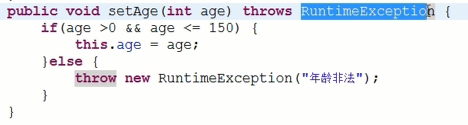
throw的概述

\* 在功能方法内部出现某种情况，程序不能继续运行，需要进行跳转时，就用throw把异常对象抛出。

\* 用在方法体内，跟的是异常对象名，只能抛出一个异常对象名，表示抛出异常，由方法体内的语句处理

# 抛出

## 原生异常



## 自定义异常

### 为什么需要自定义异常

如果JDK没有提供对应的异常，需要自定义异常。就为了区分名字，更快的检索到错误，然而所有工作交给父类来做

### 步骤

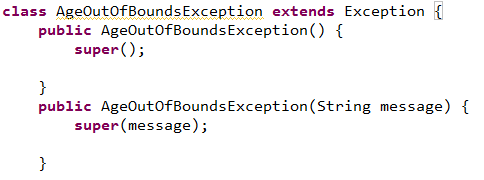
使用自定义的异常一般有如下步骤：

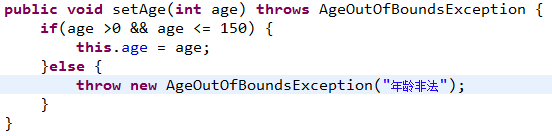
1. 通过继承java.lang.Exception类声明自己的异常类。重写父类构造



2.在方法中生成自定义异常的实例，用throw抛出。

3.在方法的首部用throws声明该方法可能抛出的异常。





继承RuntimeException

