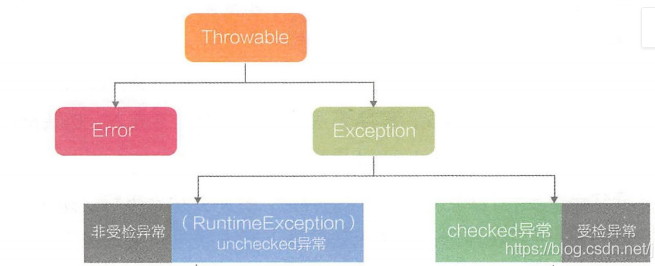
Java异常总结

## 设计模式

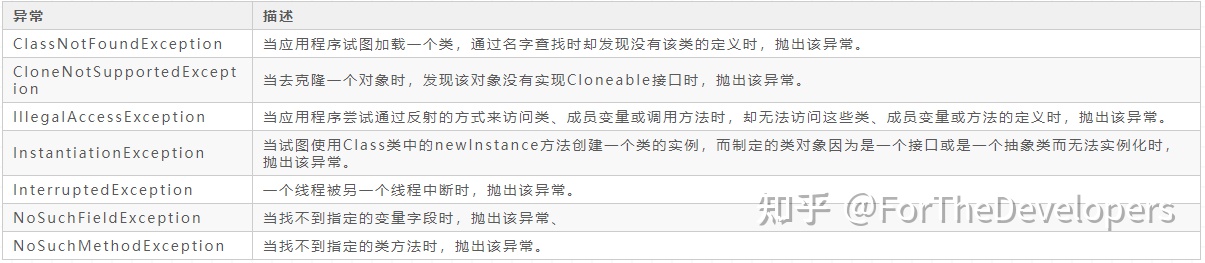
Java异常使用了链式处理既**职责链模式**

## 异常结构

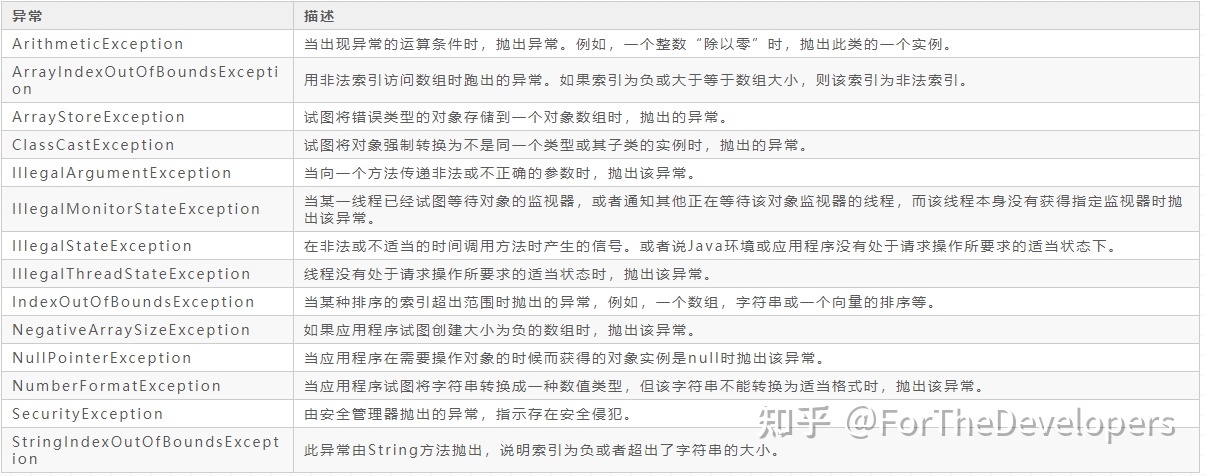


Error：是程序中无法处理的错误，表示运行应用程序中出现了严重的错误。此类错误一般是JVM出现问题。有Virtual MachineError（虚拟机运行错误）、NoClassDefFoundError（类定义错误）等。比如说当jvm耗完可用内存时，将出现OutOfMemoryError。此类错误发生时，JVM将终止线程。非代码性错误。因此，当此类错误发生时，应用不应该去处理此类错误。

Checked异常：Exception中除RuntimeException极其子类之外的异常。编译器会检查此类异常，如IOException，程序必须对该异常进行处理，要么使用try-catch捕获，要么使用throws语句抛出，否则编译不通过。



Runtime异常：RuntimeException类极其子类表示JVM在运行期间可能出现的错误。编译器不会检查此类异常，并且不要求处理异常，比如用空值对象的引用（NullPointerException）、数组下标越界（ArrayIndexOutBoundException）。编译器**不会检查此类异常**且能够通过编译器的编译，当然也可以捕获该异常进行处理。



### 总结：Checked异常与Runtime异常的区别：

1. 编译器是否对异常进行检查，检查：checked异常必须处理，不检查：runtime异常可以不处理。
2. 代码中要求添加try catch、 throws xxxException的异常，均为checked异常， 相反 既 runtime异常。

## throw 与throws的区别？

1. throws 用在方法名后边，后边跟的是**类名**
2. thow 用在方法体中，后边跟的是**对象**。



Java异常链与如何查看异常



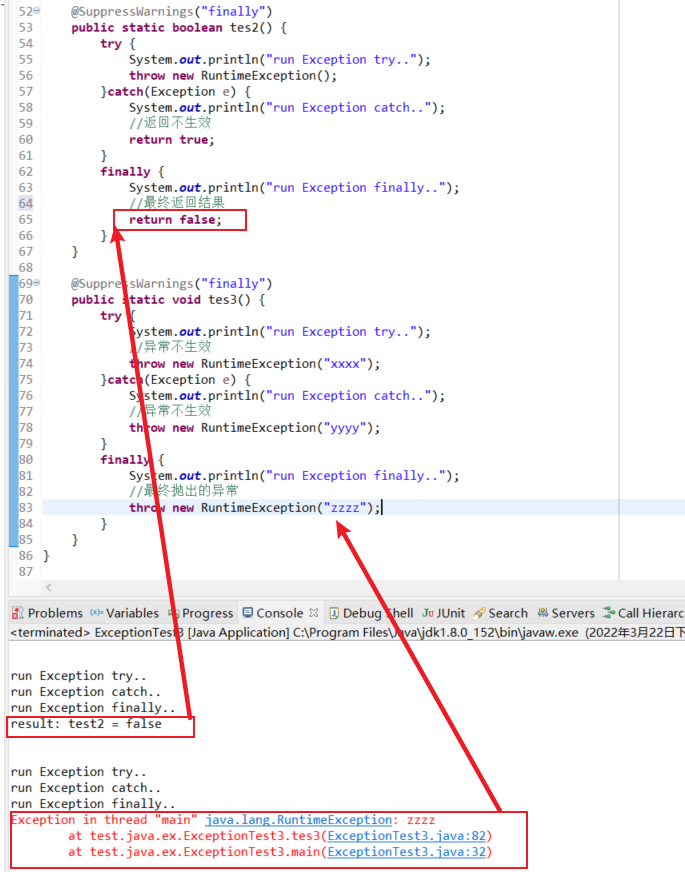
## Finally语句块执行

当程序遇到return或throw语句时，该方法并不会立即结束。会查找finally是否存在。当存在finally语句块时则执行finally块代码，没有则立即执行return或throw 语句。

注意：

1.finally语句块始终都会执行。

2.当finally中存在 return或throw时，结果都已finally中的返回为准，即便try、catch语句块中也存在return或throw语句。



## try - with - resources详解

https://blog.csdn.net/tianzhonghaoqing/article/details/118720380