JAVA 异常

异常处理是程序设计中一个非常重要的方面，也是程序设计的一大难点，从C开始，你也许已经知道如何用if...else...来控制异常了，也许是自发的，然而这种控制异常痛苦，同一个异常或者错误如果多个地方出现，那么你每个地方都要做相同处理，感觉相当的麻烦!

　　Java语言在设计的当初就考虑到这些问题，提出异常处理的框架的方案，所有的异常都可以用一个类型来表示，不同类型的异常对应不同的子类异常(这里的异常包括错误概念)，定义异常处理的规范，在1.4版本以后增加了异常链机制，从而便于跟踪异常!这是**Java**语言设计者的高明之处，也是Java语言中的一个难点，下面是我对Java异常知识的一个总结，也算是资源回收一下。

Java中异常的基类为Throwable, Error和Exception是其子类，通过捕获Throwable类型的一场可以捕获Error。Error和RuntimeException类型的异常。即使不捕获编译器也可以正常编译通过。

Error 是 Throwable 的子类，表示仅靠程序本身无法恢复的严重错误，用于指示合理的应用程序不应该试图捕获的严重问题。

Exception 类及其子类是 Throwable 的一种形式，它指出了合理的应用程序想要捕获的条件，表示程序本身可以处理的异常。

还有一点，不可忽视：finally语句在任何情况下都必须执行的代码，这样可以保证一些在任何情况下都必须执行代码的可靠性。比如，在数据库查询异常的时候，应该释放JDBC连接等等。finally语句先于return语句执行，而不论其先后位置，也不管是否try块出现异常。finally语句唯一不被执行的情况是方法执行了System.exit()方法。System.exit()的作用是终止当前正在运行的 Java 虚拟机。

Try块和finally块同时抛出异常的情形：

一个方法里面try块和finally块同时抛出异常，调用主线程中捕获的异常为finally块中的异常，try块的异常貌似丢失了，要注意这个问题java的bug。

Exception类可以分为两种：运行时异常和受检查异常。

　　1、运行时异常

　　RuntimeException类及其子类都被称为运行时异常，这种异常的特点是Java编译器不去检查它，也就是说，当程序中可能出现这类异常时，即使没有用try...catch语句捕获它，也没有用throws字句声明抛出它，还是会编译通过。例如，当除数为零时，就会抛出java.lang.ArithmeticException异常。

　　2、受检查异常

　　除了RuntimeException类及其子类外，其他的Exception类及其子类都属于受检查异常，这种异常的特点是要么用try...catch捕获处理，要么用throws语句声明抛出，否则编译不会通过。

　　3、两者的区别

　　运行时异常表示无法让程序恢复运行的异常，导致这种异常的原因通常是由于执行了错误的操作。一旦出现错误，建议让程序终止。

　　受检查异常表示程序可以处理的异常。如果抛出异常的方法本身不处理或者不能处理它，那么方法的调用者就必须去处理该异常，否则调用会出错，连编译也无法通过。当然，这两种异常都是可以通过程序来捕获并处理的