

读书笔记：《Java2教程》（八）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E8_AF_BB_E4_B9_A6_E7_AC_94_E8_c104_145220.htm

Java异常处理：什么是异常：异常可以看做是运行时的错误，不正常的操作产生的结果等，可能来源于I/O错误，设计逻辑的错误等。异常处理就是一种用于方便捕捉并进入适当处理过程的手段，可以方便我们的程序设计。方法：try{} catch{} finally{} 组合来捕获并处理异常。其中，try { } 用于定义可能产生异常的代码段，catch { } 部分是用来处理异常的，而finally{}部分则做一些清理工作。其处理的逻辑是：1 try中语句在执行中如果没有发生异常，则转到后继续运行2 try段中一旦某一条语句发生了异常，则在try段其余的语句不会被继续执行，而是立即转到catch部分进行异常处理。3 finally段是一个特殊的地方，其中的代码一定会被执行：在没有发生异常的情况下，在try块之后执行；发生异常后，则在符合条件的catch语句段后执行！切记！4 finally段一般用于完成关闭文件，释放网络套接字，或者其它清理工作。5 try{} catch{} finally{} 组合中，以下情况都是真确的：try{} catch{} finally{} 完全版本 try{} catch{} 没有finally{} try{} finally{} 没有catch{} 但是，如果没有catch{} finally{} ，而只有try{} ，这时Java语法所不允许的。注意：try 和 catch 代码段是不可分割的； 对一个catch 代码段，可以存在多个catch 部分，分别用来捕获不同类型的异常，这些catch部分也不能分开； 接下来，我们来思考一个问题：当我们采用try{} finally{}组合时，程序如何来处理异常呢？是的，我们没有提供catch { } 代码段，那么在异常发生以后，

我们没有处理，这个异常怎么样了？答案是，这个异常被转给了该方法的调用者。应该了解的一个常识是，在执行过程方法调用的时候，计算机实际上会做一个保护调用环境，并将其压栈的工作，然后才转到被调用的过程和方法的入口开始执行；执行完毕，将栈中保存的环境进行恢复，回到调用者调用该方法的语句，继续执行。为什么要说这个呢，因为我们的问题和这个背景知识有密切的关系：异常被转移了。这个机制被成为：异常传递机制。如果调用者也不处理，那么异常就继续沿调用顺序继续被传递，直至到达main()函数，如果main()还是不处理，那只能由JVM站出来了，最后的结果就是屏幕上出现一些错误信息：告诉用户，应用发生了异常，是从哪里开始发生的，等等.....，我现在没有办法继续工作了。（谁让你们都不处理的？）Java的异常捕获顺序

Java的Exception 有很多种类，在Java1.4的api document中，列出的就有： Direct Known Subclasses: AclNotFoundException, ActivationException, AlreadyBoundException, ApplicationException, AWTException, BackingStoreException, BadLocationException, CertificateException, ClassNotFoundException, CloneNotSupportedException, DataFormatException, DestroyFailedException, ExpandVetoException, FontFormatException, GeneralSecurityException, GSSEException, IllegalAccessException, InstantiationException, InterruptedException, IntrospectionException, InvalidMidiDataException, InvalidPreferencesFormatException, InvocationTargetException, IOException, LastOwnerException, LineUnavailableException,

MidiUnavailableException, MimeTypeParseException, NamingException, NoninvertibleTransformException, NoSuchFieldException, NoSuchMethodException, NotBoundException, NotOwnerException, ParseException, ParserConfigurationException, PrinterException, PrintException, PrivilegedActionException, PropertyVetoException, RefreshFailedException, RemarshalException, RuntimeException, SAXException, ServerNotActiveException, SQLException, TooManyListenersException, TransformerException, UnsupportedAudioFileException, UnsupportedCallbackException, UnsupportedFlavorException, UnsupportedLookAndFeelException, URISyntaxException, UserException, XAException

很多吧。在我们的程序中，有两个不可能是我们要知道的：我们的程序不可能处理所有的异常；我们的程序也不可能产生所有的异常；我们只关心也只能关心其中一部分我们感兴趣的种类。前面我们也提到了，一个try{}代码段可以有多个catch{}，那么我们怎么安排这多个catch{}呢？原则：先特殊，再一般。就是说，先处理子类异常，再处理父类异常。可能不太好理解，举个例子：

```
public class ReadFile { public static void main() { try { //some statement which concerning about reading files. } catch(FileNotFoundException e) { System.err.println( " File not fond. " ). System.err.println(e.getMessage()). e.printStackTrace(). } catch(IOException e) { System.err.println( " I/O error. " ). System.err.println(e.getMessage()). e.printStackTrace(). } } }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问

