

java编程中异常处理的优劣观 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_java\\_E7\\_BC\\_96\\_E7\\_A8\\_8B\\_c104\\_252300.htm](https://www.100test.com/kao_ti2020/252/2021_2022_java_E7_BC_96_E7_A8_8B_c104_252300.htm) 关键字: 异常,通

告(throws),throw,try,catch,RuntimeException,不可检

查(Unchecked)异常,可检查(Checked)异常 Java编程中的异常处理是一个很常见的话题了，几乎任何一门介绍性的Java课程都会提到异常处理。不过，我认为很多人其实并没有真正掌握正确处理异常情况的方法和策略，最多也就不过了解个大概，知道点概念。本文就对三种不同程度和质量的Java异常处理进行了讨论，所阐述的处理异常的方式按手法的高下分为：好，不好和恶劣三种。同时向你提供了一些解决这些问题的技巧。首先解释一些java异常处理中必须搞清楚的定义和机制。Java语言规范将自Error类或RuntimeException类衍生出来的任何违例都称作“不可检查”(Unchecked)异常.其他所有异常则称作“可检查”(Checked)异常。所谓可检查异常，是指我们应该自行处理的异常。至于处理的手段，要么加以控制(try catch)，要么通告(throws)他们有可能产生。通常，应捕捉那些已知如何处理的异常，而通告那些不知如何处理的异常。而对那些不可检查异常来说，他们要么在我们的控制之外(Error),要么是我们首先就不该允许的情

况(RuntimeException).至于异常的指定，Java的规则非常简单：一个方法必须通告自己可能产生的所有可检查异常。编写自己的方法时，并不一定要通告出方法实际可能产生的每一个异常对象，要想理解什么时候必须要方法的throws从句来通告异常，就必须知道对一个异常来说，他只有可能在下面四

种情况下才会产生：1. 调用了可能产生异常的方法。比如BufferedReader类的readLine方法。该方法通告java.io.IOException异常 2. 侦测到一个错误，并用throw语句产生异常。 3. 出现一个编程错误。比如a[-1] = 0。 4. Java产生内部错误。如果出现头两种情况之一，必须告诉打算使用自己方法的人：假如使用这个方法，可能造成一个异常的产生(即在方法头上使用throws)，一个简单的记忆方法：只要含有throw,就要通告throws。如果一个方法必须同时处理多个异常，就必须在头内指出所有异常。就像下例展示的那样，用逗号对他们进行分割：1234567class Animation { public Image loadImage(String s) throws EOFException, MalformedURLException { ..... } }然而，我们不需要通告内部java错误，也不应该通告自RuntimeException衍生出来的异常。好异常处理 好异常处理提供了处理程序错误的统一机制。事实上，Java语言通过向调用者提出异常警告的方式而显著地提升了软件开发中的异常处理能力。这种方式把Java语言中的“方法（method）”进行了扩展和增强，使之包括了自身的错误条件。下面就让我们看一个例子，这个例子说明了这种情况。以下是FileInputStream构造器之一的原型：public FileInputStream(String name) throws FileNotFoundException Java的方法和构造器必须声明他们在被调用时可能“抛出”的异常，采用的关键字就是“throws”。这种在方法原型中出现的异常提示增加了编程的可靠性。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)