

计算机二级辅导:java异常处理详解计算机等级考试 PDF转换
可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645376.htm 第一部分选

择checked or unchecked 这里需要对异常的理解。什么算异常？java的异常处理机制是用来干什么的？异常和错误有什么区别？异常机制就是java的错误处理机制！java中的异常意味着2点：第一，让错误处理代码更有条理。这使得正常代码和错误处理代码分离。第二，引入了context的概念，认为有些错误是可以被处理的。问题就出在这儿了。java的checked异常指的就是在当前context不能被处理的错误！这句话其实是对上面2点的总结。首先checked异常是一种错误，其次这种错误可以被处理（或修复）。checked异常就是可以被处理（修复）的错误，unchecked异常其实就是无法处理（修复）的错误。说到这儿，应该清楚了。别的语言没有checked异常，就是说它们认为错误都无法被修复，至少在语言级不提供错误修复的支持。java的catch clause干的就是错误修复的事。我的理解是，用好java的异常，其实就是搞清楚什么时候该用checked异常。应该把unchecked异常当作缺省行为。unchecked异常的意思是：当我做这件事时，不可思议的情况发生了，我没办法正常工作下去！然后抛出一个unchecked异常，程序挂起。而checked异常的意思是：当我做这件事时，有意外情况发生，可以肯定的是，活是没法干了，但是要不要挂起程序，我这个函数没法做主，我只能汇报上级！其实，从上面的分析可以看出，java引入checked异常只是让程序员多了一个选择，它并不强迫你使用checked异常。如果你

对什么时候应该使用checked异常感到迷惑，那么最简单的办法就是，不要使用checked异常！这里包括2个方面：第一，你自己不必创建新的异常类，也不必在你的代码中抛出checked异常，错误发生后只管抛出unchecked异常；第二，对已有API的checked异常统统catch后转为unchecked异常！使用unchecked异常是最省事的办法。用这种方法也可以享受“正常代码和错误处理代码分离”的好处。因为我们在调用方法时，不用根据其返回值判断是否有错误出现，只管调用，只管做正事就ok了。如果出现错误，程序自然会知道并挂起。这样的效果是怎样的呢？第一，我们的业务代码很清晰，基本都是在处理业务问题，而没有一大堆判断是否有错的冗余代码。（想想看，如果没有throw异常的机制，你只能通过函数的返回值来判断错误，那么你在每个调用函数的地方都会有判断代码！）第二，我们的代码假设一切正常，如果确实如此，那么它工作良好。但是一旦出现任何错误，程序就会挂起停止运行。当然，你可以查看日志找到错误信息。那么使用checked异常又是怎样的呢？第一，你需要考虑更多的问题。首先在设计上就会更加复杂，其次就是代码更加冗长。设计上复杂体现在以下方面：1 对异常（错误）的抽象和理解。你得知道什么情况才能算checked异常，使得上级确实能够处理（修复）这种异常，并且让整个程序从这种设计中确实得到好处。2 对整个自定义checked异常继承体系的设计。正如那篇文章所说，总不能在一个方法后面抛出20个异常吧！设计自定义checked异常，就要考虑方法签名问题，在合适的时候抛出合适的异常（不能一味的抛出最具体的异常，也不能一味抛出最抽象的异常）第二，业务代码相比较使

用unchecked的情况而言，不够直接了当了。引入了throws签名和catch clause，代码里有很多catch，方法签名也和异常绑定了。第三，有了更强的错误处理能力。如果发生了checked异常，我们有能力处理（修复）它。表现在不是任何错误都会导致程序挂起，出现了checked异常，程序可能照样运行。整个程序更加健壮，而代价就是前面2条。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