

给JAVA设计开发新手一些建议和意见(2)Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/560/2021_2022__E7_BB_99J

AVA_E8_AE_BE_c104_560206.htm 【处理好你的异常】异常处理是Java编程中非常重要的一个部分。建议在使用异常之前阅读或者。下面从书中摘出几条建议：***绝对不要忽略异常***
千万不要隐藏异常** ***仅在不正常的情况下使用异常*** ***对可恢复的情况使用可检查异常，对程序错误使用运行时异常(RuntimeException)*** ***给方法引发的异常做文档*** ***在详细信息里面包括失败捕获信息*** ***使用finally避免资源泄漏***。。。。
在这里特别提出的是，在开发中要特别处理NULL的情况，否则经常引发NullPointerException异常，在Java里这是一个最令人头疼的异常了。如果你的程序因为一个NULL值，而报了几十个NullPointerException的话，不但得让人烦死，而且还非常难以找到错误所在。所以在Java中一定要注意这个问题。如果你的函数不允许Null值，那么可以截获它，抛出一个异常，或者给客户更友好的提示，难道不好吗？让我们来看一个例子：

```
public String getName(User aUser) { //如果aUser为Null，会发生什么情况 return aUser.getName(). }
```

很明显，如果参数为Null，就会抛出异常。应该改为：

```
public String getName(User aUser) { if(null==aUser) { return "" }. } else { return aUser.getName(). } }
```

或者你要求参数不能为空，还可以抛出一个异常，强制使用者不能传入空值。还有经常被忽略的是RuntimeException和普通异常的区别，在Java中，这是一个特殊的异常类，程序中如果遇到这个异常，用户可以不截获它，而如果是其他的普通异常，就不许要截获它。我们的代

码经常这么写: `try { //your code here } catch(Exception e) { //do warn }` 这样写的话, 就截获了所有异常, 当然也包括了 `RuntimeException`。在很多情况下, 这是不合适的处理方式, 我们只应截获必要的异常, 而应该忽略 `RuntimeException`。关于 `RuntimeException`, 在 Spring 中还有更好的利用方式, 建议阅读 Spring 框架中在事务中对异常的处理代码, 例如对 Jdbc 抛出的 `SQLException` 的转换。关于异常处理, 我提出几点建议: *捕获异常而且再次抛出时要包含原来的异常信息* 不要忘了 `RuntimeException`, 除非必要, 否则不要用 `catch(Exception e)` 的方式捕获所有异常。*不要用异常做流程控制, 异常的性能代价比较高昂。(对此, 可能有人不同意。此处不详细讨论)* 不要把异常处理都抛给别人, 本函数有能力处理的就不要抛出。在此建议读者详细阅读或者。【过度依赖】在定位错误的时候, 经常遇到浏览了七八个文件还是没有找到什么地方执行了真正需要的函数, 这个时候就非常郁闷。A调用了B, B调用了C, C调用了D。。。。。。让人找不到北 面对这样的程序, 存在的问题不仅仅是定位错误麻烦, 而且如果需要维护这样的函数库/框架, 恐怕你的有非常高的统御能力才行, 否则打死我也不去维护。那么我们自己最好不要写这样的程序出来给人用。 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