



链滴

1.11异常处理：处理错误

作者：[someone756](#)

原文链接：<https://ld246.com/article/1441540964492>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

 自从编程语言问世以来，错误处理就始是最困难的问题之一。因为设计一个良好的错误处理机制非常困难，所以许多语言直接略去这个问题将其交给程序库设计者处理，而这些设计者也只是提出了一些不彻底的方法，这些方法可用于许多很易就可以绕过此问题的场合，而且其解决方式通常也只是忽略此问题。大多数错误处理机制的主要问在于，它们都依赖于程序员自身的警惕性，这种警惕性来源于一种共同的约定，而不是编程语言所强的。如果程序员不够警惕——通常是因为它们太忙，这些机制就很容易被忽视。

 异常处理将错误处理直接置于编程语言，有时甚至置于操作系统中。异常是一种对象，它从出错点被“抛出”，并被专门设计来处理特定类型错误的相应的异常处理器“捕获”。异常处理就像是与程序正常执行路并行的、在错误发生时执行的另一条路径。因为它是另一条完全分离的执行路径，所以他不会干扰正的执行代码。这往往使得代码编写变得简单，因为不需要被迫定期检查错误。此外，被抛出的异常不方法返回的错误值和方法设置的用来表示错误的标志位那样可以被忽略。异常不能被忽略，所以得保它一定会在某处得到处理。最后需要指出的事：异常提供了一种从错误状况进行可靠恢复的途径。现不再是只能退出程序，你可以经常进行校正，并恢复程序的执行，这些都有助于编写出更健壮的程序

 Java的异常处理在众多的编程语言中格引人注目，因为Java一开始就内置了异常处理，而且强制你必须使用它。它是唯一可以接受错误报告式。如果没有编写正确的处理异常的代码，那么就会得到一条编译是的出错消息。这种有保障的一致有时回事的错误处理非常容易。

 值得注意的是，异常处理不是面向对象特征——尽管在面向对象语言中异常常常被表示成为一个对象。异常处理在面向对象言出现之前就已经存在了。

总结

错误处理机制设计很难，错误处理很容易被忽视

异常是一种对象，出错被“抛出”，并被异常处理器“捕获”

与正常执行路径并行，不会干扰正常的执行代码

异常处理不是面向对象的特征。

-----END-----