

Java中“异常机制”深入研究（一）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\\_Java\\_E4\\_B8\\_AD\\_E2\\_80\\_9C\\_c104\\_145060.htm](https://www.100test.com/kao_ti2020/145/2021_2022_Java_E4_B8_AD_E2_80_9C_c104_145060.htm) 由于本文旨在探讨Java"异常机制"的深层原理，因此关于"异常"的使用方法都不做详细说明。首先看一段非常熟悉的用于打开一个文件的C程序段：

```
FILE *fp. fp=fopen(filename,"rw"). if(fp==NULL){ printf("cannot open file\n"). exit(0). }
```

在这段程序中，if条件语句中的一段用来处理没有找到指定文件，或者其它原因无法正确打开指定文件。可是如果遇到一个责任心不强的程序员，他可能认为出现找不到文件的可能性很小，或者由于思路集中在程序功能的实现上而忘记了处理这种情况。这时程序同样可以正确编译，而且一般情况下也不会出现问题。但此时这段程序可以肯定说是不够健壮的，而且一旦这段程序发生了错误也会让程序员很难发现错误出在哪里。在C语言以及其它大多数高级语言中都可以举出很多这种例子。也就是一个函数在使用的时候，可能会出现并没有达到这个函数的使用目的的情况，哪怕在这段程序的特定使用环境下发生这种异常情况的可能性只有万分之一。常用处理的方法就是，程序员在需要使用某个函数时必须充分了解可能会有什么原因导致该函数不能正确执行，然后加入相应的条件判断语句来进行处理。后面将有一个例子说明这个问题。而Java的"异常机制"就是在处理上述问题中给了程序员非常简单而灵活的方式。一般来说，其它高级语言主要是让函数使用者来关注该函数可能会出现异常情况，而java则是把这件事情交给方法(和函数对应的概念，在Java中称方法)的设计者来做。这对于方法的使用

者来说带来的方便是不会因为责任心不强，或者办事丢三那四，会忘了在使用方法时处理可能发生的异常情况。而麻烦就是，在使用一个可能会发生异常的方法时，绝对不能视而不见，而必须做出相应的处理。也就是说象上述C程序段中，如果忘了if程序块，这个程序甚至还能蒙过一个外行上司，但当使用Java来完成这个功能时，只要用到的方法使用了"异常"机制，如果不对可能产生"异常"的方法进行相应处理，java编译器是不会让其通过的。一、"异常类"的组织形式 Java系统类中的方法产生的异常都被组织成"异常类"（还有Error类，不在本文讨论范围），此方法和它相关的"异常类"通过throws关键字关联在一起，并且这些类都必须是Exception类的子类。任何一个自己开发的类的方法中如果可能会产生某种异常，也可以将这种异常组织成一个"异常类"，但这个"异常类"同样必须是Exception的子类，或孙子类等等。例1：

```
/*isLegal于检查数据是否合法，当>0时视为合法，返回合法值，*否则视为不合法，抛出"异常"。*/ int isLegal(int dt) throws LowZeroException // 这种定义本文中均称为方法与"异常"通过throws建立了关联 if(dt>=0){ return data. } else throw new LowZeroException(). } /*自己写的异常类，继承自Exception*/ class LowZeroException extends Exception { public LowZeroException(){ super(). } } 仔细观察方法isLegal()，它体现出的最值得注意的特色是，它有两种方式的函数出口，一种是通过return语句，返回的是方法本身定义的类型实例，另一种是通过throw，返回的是"异常类"的对象实例，Java中称之为抛出"异常"。对比一下C中如何处理同样的问题的： int isLegal(int dt) { if(dt>=0){ return data. } else return -1. // 通过一个
```

特定值来表明出错 } 由于C只能通过return返回函数值，所以在处理异常情况时则可能通过以上方式来处理。当然这就要求isLegal()函数的使用者必须知道函数中使用返回值-1来表明出现不合法数据的情况。对比这两种处理方法，可以知道java的"异常机制"把处理异常事件的职能和方法本身的职能通过两个不同出口分离开来。 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)