



脚本之家  
www.jb51.net

买源码，上源市场 脚本之家旗下

开始下载 只需3步  
即可完成快速安装和扫描

点按免费下载  
开始快速扫描  
获得一台干净的

免费下载

网站首页 网页制作 网络编程 脚本专栏 脚本下载 数据库 CMS教程 电子书籍 平面设计 媒体动画 其它

C#教程 vb vb.net C语言 Java编程 Delphi java Android IOS Swift Scala 易语言 其它相关

阿里云 云数据库Redis版 最高10万QPS，宕机秒级拉起

腾讯云 新注册用户域名抢购1元起 .com首年28元 .cn首年19元

立即抢购

高防备案CDN

【3000个备案老域名】100元起 每天更新

阿里云/腾讯云/百度云等/量大从优

4核独服199/16核独服360|创梦网络

产品发布、创业开店、需求任务找大师兄

鼎点网络百兆独享服务器仅需999元

5M独享云主机599/年

95IDC香港沙田CN2服务器 599/月

30余款阿里云产品免费6个月>>

【亿恩】DELL品牌服务器，月付799元起

云服务器5折，天天抽红包抵扣

新注册用户域名抢购1元起

港湾网络-徐州百独16核32G 999/月~

[香港双高防]无视CC\*DDOS/堪比广东！

群英云服务器送10M带宽30G防御,49元起

枫信科技-江苏双线10M保证-399/元

1G香港云49元/美国云49元/韩国云89元

服务器租用/托管-域名空间/认准腾佑科技

\*\*香港高防10M大带宽独服，低至999元\*\*

中原地区核心数据中心，首月托管免费

免备vps20/百独799/双线350/45互联

畅游网络 百独服务器 包跑满 998元

好系统,装机首选好系统

海西数据全球服务器租用

美国/香港/国内/高防服务器VPS★好优云

酷播资源网-万部VIP电影免费采集保证不卡

知了云,OpenStack云服务器★5折优惠★

云彩网络 100G防服务器450元

韩国香港美国站群服务器 巨牛网络

服务器租用 199元起

九九数据 — 工信部认可正规资质IDC接入商

浦东数据中心上海电信4星云主机30元/月起

新睿云 XRcloud.net

2核4G/50G磁盘/4MB带宽

60天 免费试用

10万级IOPS，全冗余架构

免费领取

西部数码 www.west.cn

云主机 仅需48元

阿里云

一键部署云服务器环境

官方源镜像、安全稳定、免配置

0元起

新天域互联 simcentric solutions

广告

免备案 香港空间

广告

Java 常见异常(Runtime Exception )详细介绍并总结

转载

2016-10-11 作者：绿一茶下 我要评论

这篇文章主要介绍了Java 常见异常(Runtime Exception )详细介绍并相关资料,大家在开发Java 应用软件的时候经常会遇到各种异常这里帮大家整理了一部分，并解释如何解决，需要的朋友可以参考下

ERROR

The requested URL could not be retrieved

本文重在Java中异常机制的一些概念。写本文的目的在于方便我很长时间后若是忘了这些东西可以通过这篇文章迅速回忆起来。

1. 异常机制

1.1 异常机制是指当程序出现错误后，程序如何处理。具体来说，异常机制提供了程序退出的安全通道。当出现错误后，程序执行的流程发生改变，程序的控制权转移到异常处理器。

1.2 传统的处理异常的办法是，函数返回一个特殊的结果来表示出现异常（通常这个特殊结果是大家约定俗称的），调用该函数的程序负责检查并分析函数返回的结果。这样做有如下的弊端：例如函数返回-1代表出现异常，但是如果函数确实要返回-1这个正确的值时就会出现混淆；可读性降低，将程序代码与处理异常的代码混杂在一起；由调用函数的程序来分析错误，这就要求客户程序员对库函数有很深的了解。

1.3 异常处理的流程

大家感兴趣的内容

1 java使double保留两位小数的多方

2 JAVA8 十大新特性详解

3 java.net.SocketException: Conn

4 java写入文件的几种方法分享

5 JAVA 十六进制与字符串的转换

6 Java环境变量的设置方法(图文教程

7 java list用法示例详解

8 java中File类的使用方法

9 JavaWeb实现文件上传下载功能实例

10 JAVA中字符串函数subString的用法

阿里云 建网站 搭应用 首选云服务器 0.73元/日起

腾讯云 搭建微信小程序 就选腾讯云 一站式部署 共享9亿客户 一键获取

1.3.1 遇到错误，方法立即结束，并不返回一个值；同时，抛出一个异常对象

1.3.2 调用该方法的程序也不会继续执行下去，而是搜索一个可以处理该异常的异常处理器，并执行其中的代码

## 2 异常的分类

### 2.1 异常的分类

2.1.1 异常的继承结构：基类为Throwable，Error和Exception继承Throwable，RuntimeException和IOException等继承Exception，具体的RuntimeException继承RuntimeException。

2.1.2 Error和RuntimeException及其子类成为未检查异常（unchecked），其它异常成为已检查异常（checked）。

### 2.2 每个类型的异常的特点

2.2.1 Error体系 Error类体系描述了Java运行系统中的内部错误以及资源耗尽的情形。应用程序不应该抛出这种类型的对象（一般是由虚拟机抛出）。如果出现这种错误，除了尽力使程序安全退出外，在其他方面是无能为力的。所以，在进行程序设计时，应该更关注Exception体系。

2.2.2 Exception体系 Exception体系包括RuntimeException体系和其他非RuntimeException的体系

2.2.2.1 RuntimeException RuntimeException体系包括错误的类型转换、数组越界访问和试图访问空指针等等。处理RuntimeException的原则是：如果出现RuntimeException，那么一定是程序员的错误。例如，可以通过检查数组下标和数组边界来避免数组越界访问异常。

2.2.2.2 其他（IOException等等）这类异常一般是外部错误，例如试图从文件尾后读取数据等，这并不是程序本身的错误，而是在应用环境中出现的外部错误。

### 2.3 与C++异常分类的不同

2.3.1 其实，Java中RuntimeException这个类名起的并不恰当，因为任何异常都是运行时出现的。（在编译时出现的错误并不是异常，换句话说，异常就是为了解决程序运行时出现的错误）。

2.3.2 C++中logic\_error与Java中的RuntimeException是等价的，而runtime\_error与Java中非RuntimeException类型的异常是等价的。

## 3 异常的使用方法

### 3.1 声明方法抛出异常

3.1.1 语法：throws（略）

3.1.2 为什么要声明方法抛出异常？方法是否抛出异常与方法返回值的类型一样重要。假设方法抛出异常确没有声明该方法将抛出异常，那么客户程序员可以调用这个方法而且不用编写处理异常的代码。那么，一旦出现异常，那么这个异常就没有合适的异常控制器来解决。

3.1.3 为什么抛出的异常一定是已检查异常？RuntimeException与Error可以在任何代码中产生，它们不需要由程序员显示的抛出，一旦出现错误，那么相应的异常会被自动抛出。而已检查异常是由程序员抛出的，这分为两种情况：客户程序员调用会抛出异常的库函数（库函数的异常由库程序员抛出）；客户程序员自己使用throw语句抛出异常。遇到Error，程序员一般是无能为力的；遇到RuntimeException，那么一定是程序存在逻辑错误，要对程序进行修改（相当于调试的一种方法）；只有已检查异常才是程序员所关心的，程序应该且仅应该抛出或处理已检查异常。

3.1.4 注意：覆盖父类某方法的子类方法不能抛出比父类方法更多的异常，所以，有时设计父类的方法时会声明抛出异常，但实际的实现方法的代码却并不抛出异常，这样做的目的就是为了方便子类方法覆盖父类方法时可以抛出异常。

### 3.2 如何抛出异常

3.2.1 语法：throw（略）



### 最近更新的内容

Spring.Net控制反转IoC入门使用  
Java 方法签名详解及实例代码  
JAVA中使用双括号来初始化静态常量...  
Log4j定时打印日志及添加模块名配置...  
Java 正则表达式详细介绍  
Activiti流程图查看实例  
java抓取网页或文件中的邮箱号码  
java 日期各种格式之间的相互转换实...  
JAVA中String类与StringBuffer类的区...  
Spring Boot集成springfox-swagger...

众生网络  
品牌服务器租用

集思网络  
VPS主机

枫信科技  
IDC服务商

### 常用在线小工具

CSS代码工具  
JavaScript代码格式化工具  
在线XML格式化/压缩工具  
php代码在线格式化美化工具  
sql代码在线格式化美化工具  
在线HTML转义/反转义工具  
在线JSON代码检验/检验/美化/格式化  
JavaScript正则在线测试工具  
在线生成二维码工具(加强版)  
更多在线工具



3.2.2 抛出什么异常？对于一个异常对象，真正有用的信息时异常的对象类型，而异常对象本身毫无意义。比如一个异常对象的类型是ClassCastException，那么这个类名就是唯一有用的信息。所以，在选择抛出什么异常时，最关键的就是选择异常的类名能够明确说明异常情况的类。

3.2.3 异常对象通常有两种构造函数：一种是无参数的构造函数；另一种是带一个字符串的构造函数，这个字符串将作为这个异常对象除了类型名以外的额外说明。

3.2.4 创建自己的异常：当Java内置的异常都不能明确的说明异常情况的时候，需要创建自己的异常。需要注意的是，唯一有用的就是类型名这个信息，所以不要在异常类的设计上花费精力。

3.3 捕获异常如果一个异常没有被处理，那么，对于一个非图形界面的程序而言，该程序会被中止并输出异常信息；对于一个图形界面程序，也会输出异常的信息，但是程序并不中止，而是返回用户界面处理循环中。

3.3.1 语法：try、catch和finally（略）控制器模块必须紧接在try块后面。若掷出一个异常，异常控制机制会搜寻参数与异常类型相符的第一个控制器随后它会进入那个catch 从句，并认为异常已得到控制。一旦catch 从句结束对控制器的搜索也会停止。

3.3.1.1 捕获多个异常（注意语法与捕获的顺序）（略）

3.3.1.2 finally的用法与异常处理流程（略）

3.3.2 异常处理做什么？对于Java来说，由于有了垃圾收集，所以异常处理并不需要回收内存。但是依然有一些资源需要程序员来收集，比如文件、网络连接和图片等资源。

3.3.3 应该声明方法抛出异常还是在方法中捕获异常？原则：捕捉并处理哪些知道如何处理的异常，而传递哪些不知道如何处理的异常

3.3.4 再次抛出异常

3.3.4.1 为什么要再次抛出异常？在本级中，只能处理一部分内容，有些处理需要在更高级的环境中完成，所以应该再次抛出异常。这样可以使每级的异常处理器处理它能够处理的异常。

3.3.4.2 异常处理流程对应与同一try块的catch块将被忽略，抛出的异常将进入更高的一级。

#### 4 关于异常的其他问题

4.1 过度使用异常首先，使用异常很方便，所以程序员一般不再愿意编写处理错误的代码，而仅仅是简简单单的抛出一个异常。这样做是不对的，对于完全已知的错误，应该编写处理这种错误的代码，增加程序的鲁棒性。另外，异常机制的效率很差。

4.2 将异常与普通错误区分开对于普通的完全一致的错误，应该编写处理这种错误的代码，增加程序的鲁棒性。只有外部的不能确定和预知的运行时错误才需要使用异常。

4.3 异常对象中包含的信息一般情况下，异常对象唯一有用的信息就是类型信息。但使用异常带字符串的构造函数时，这个字符串还可以作为额外的信息。调用异常对象的getMessage()、toString()或者printStackTrace()方法可以分别得到异常对象的额外信息、类名和调用堆栈的信息。并且后一种包含的信息是前一种的超集。

常用异常：

**UnsupportedOperationException不支持的操作**

**IllegalArgumentException非法参数**

**IndexOutOfBoundsException索引出界**

**IllegalStateException非法状态**

异常跟普通的警告等有一定的区别。当应用程序发生异常时，会中断正在执行的程序的正常指令流。也就是说，发生异常后面的代码将得不到正确的执行。甚至还会触发数据库的回退操作。

在Java开发平台中，异常包括预定义异常与自定义异常。这两种异常的类型互为补充。作为一个合格的程序开发人员，要善于在应用程序中使用异常。这可以提高应用程序的交互性。同时，也是保证应用程序正常运行的前

提。故异常的处理对于开发一个优秀的应用程序来说非常的重要。为此笔者认为程序开发人员应该对Java应用程序的常见异常有一个深入的了解。只有在了解这些常见异常的情况下，才能够做好自定义异常的处理。

## 一、常见异常的类型与原因。

对于Java应用程序的常见异常，笔者认为程序开发人员要从两个方面去了解。一是要知道有哪些常见的Java应用程序异常，二是需要知道哪些原因可能会造成这个异常。这不仅需要程序管理人员在日常工作中要注意积累，在必要的环境下还需要去从其它渠道收集资料。笔者对此就进行一个分析，希望能够对各位程序开发人员有一定的帮助。

### 1、SQLException：操作数据库异常类。

现在的Java应用程序大部分都是依赖于数据库运行的。当Java应用程序与数据库进行沟通时如果产生了错误，就会触发这个类。同时会将数据库的错误信息通过这个类显示给用户。也就是说，这个操作数据库异常类是数据库与用户之间异常信息传递的桥梁。如现在用户往系统中插入数据，而在数据库中规定某个字段必须唯一。当用户插入数据的时候，如果这个字段的值跟现有的纪录重复了，违反了数据库的唯一性约束，此时数据库就会跑出一个异常信息。这个信息一般用户可能看不到，因为其发生在数据库层面的。此时这个操作数据库异常类就会捕捉到数据库的这个异常信息，并将这个异常信息传递到前台。如此的话，前台用户就可以根据这个异常信息来分析发生错误的原因。这就是这个操作数据库异常类的主要用途。在Java应用程序中，所有数据库操作发生异常时，都会触发这一个类。所有此时Java应用程序本身的提示信息往往过于笼统，只是说与数据库交互出现错误，没有多大的参考价值。此时反而是数据库的提示信息更加有使用价值。

### 2、ClassCastException：数据类型转换异常。

在Java应用程序中，有时候需要对数据类型进行转换。这个转换包括显示的转换与隐式的转换。不过无论怎么转换，都必须符合一个前提的条件，即数据类型的兼容性。如果在数据转换的过程中，违反了原则，那么就会触发数据类型转换异常。如现在在应用程序中，开发人员需要将一个字符型的日期数据转换为数据库所能够接受的日期型数据，此时只需要在前台应用程序中进行控制，一般不会有问题。但是，如果前台应用程序缺乏相关的控制，如用户在输入日期的时候只输入月、日信息，而没有年份的信息。此时应用程序在进行数据类型转换的时候，就会出现异常。根据笔者的经验，数据类型转换异常在应用程序开发中使一个出现的比较多的异常，也是一个比较低级的异常。因为大部分情况下，都可以在应用程序窗口中对数据类型进行一些强制的控制。即在数据类型进行转换之前，就保证数据类型的兼容性。如此的话，就不容易造成数据类型的转换异常。如在只允许数值类型的字段中，可以设置不允许用户输入数值以外的字符。虽然说有异常处理机制，可以保证应用程序不会被错误的运行。但是在实际开发中，还是要尽可能多的预见错误发生的原因，尽量避免异常的发生。

### 3、NumberFormatException：字符串转换为数字类型时抛出的异常。

在数据类型转换过程中，如果是字符型转换为数字型过程中出现的问题，对于这个异常在Java程序中采用了一个独立的异常，即NumberFormatException。如现在讲字符型的数据“123456”转换为数值型数据时，是允许的。但是如果字符型数据中包含了非数字型的字符，如123#56，此时转换为数值型时就会出现异常。系统就会捕捉到这个异常，并进行处理。

Java应用程序中常见的异常类还有很多。如未找到相应类异常、不允许访问某些类异常、文件已经结束异常、文件未找到异常、字段未找到异常等等。一般系统开发人员都可以根据这个异常名来判断当前异常的类型。虽然不错，但是好记性不如烂笔头。程序开发人员在必要的时候（特别是存在自定义异常的时候），最后手头有一份异常明细表。如此的话，无论是应用程序在调试过程中发现问题，还是运行过程中接到用户的投诉，都可以及时的根据异常名字来找到异常发生的原因。从而可以在最短时间内解决异常，恢复应用程序的正常运行。这个措施笔者用了很多年，非常的有效。

## 二、异常管理的实用建议。

对于操作数据库异常来说，Java应用程序只提供了一个异常类。故光凭Java应用程序的错误信息，往往不能够帮助应用程序人员排除错误的原因。只能够指名是应用程序错误还是数据库错误导致的这个异常。为了更进一步指明问题的原因，在数据库层面定义异常的时候，最好能够说明具体的原因。如前台应用程序可能会调用数据库的函数或者过程。此时在数据库的函数或者过程中做好能够说明某个异常的具体原因。如根据某个基础表生成另一张表

的时候，某个字段不能够为空等等。将这些异常信息说明清楚后，如果真的遇到类似的异常时，操作数据库异常类就会将数据库的异常信息反会前台用户。从而有利于用户寻找问题的原因，并在最短时间内改正。当然，这需要Java程序员与数据库设计人员进行协调。

其次需要注意的是，异常并不是常态。也就是说，大部分异常可以通过前提的合理预见与预防，来消除。如设计到四则运算，可以在前台应用程序窗口中限制在除数字段内输入0值等手段来消除应用程序运行中可能产生的异常。不过这往往要求应用程序开发人员有比较丰富的工作经验以及由比较严密的思维逻辑。虽然这有一定的难度，但是笔者认为程序开发人员还是应该往这方面努力，而不要老是让用户作为你的试验品，让用户来发现应用程序中的设计Bug.笔者认为，只有一些实在是程序人员无法控制的因素才允许抛出异常。如果应用程序开发人员能够意识到这种错误、但是仍然没有引起重视或者采取有效的措施防止出现这种异常，那么笔者是不允许的。

ArithmeticException（除数为0的异常），BufferOverflowException（缓冲区上溢异常），BufferUnderflowException（缓冲区下溢异常），IndexOutOfBoundsException（出界异常），NullPointerException（空指针异常），EmptyStackException（空栈异常），IllegalArgumentException（不合法的参数异常），NegativeArraySizeException, NoSuchElementException, SecurityException, SystemException, UndeclaredThrowableException

#### 1. java.lang.NullPointerException

异常的解释是"程序遇上了空指针"，简单地说就是调用了未经初始化的对象或者是不存在的对象，即把数组的初始化和数组元素的初始化混淆起来了。数组的初始化是对数组分配需要的空间，而初始化后的数组，其中的元素并没有实例化，依然是空的，所以还需要对每个元素都进行初始化（如果要调用的话）

#### 2. java.lang.ClassNotFoundException

异常的解释是"指定的类不存在"。

#### 3. java.lang.ArithmeticException

这个异常的解释是"数学运算异常"，比如程序中出现了除以零这样的运算就会出这样的异常。

#### 4. java.lang.ArrayIndexOutOfBoundsException

异常的解释是"数组下标越界"，现在程序中大多都有对数组的操作，因此在调用数组的时候一定要认真检查，看自己调用的下标是不是超出了数组的范围，一般来说，显示（即直接用常数当下标）调用不太容易出这样的错，但隐式（即用变量表示下标）调用就经常出错了，还有一种情况，是程序中定义的数组的长度是通过某些特定方法决定的，不是事先声明的，这个时候，最好先查看一下数组的length，以免出现这个异常。

#### 5. java.lang.IllegalArgumentException

这个异常的解释是"方法的参数错误"，比如g.setColor(int red,int green,int blue)这个方法中的三个值，如果有超过2 5 5的也会出现这个异常，因此一旦发现这个异常，我们要做的，就是赶紧去检查一下方法调用中的参数传递是不是出现了错误。

#### 6. java.lang.IllegalAccessException

这个异常的解释是"没有访问权限"，当应用程序要调用一个类，但当前的方法即没有对该类的访问权限便会出现这个异常。对程序中用了Package的情况下要注意这个异常的, 异常, Java

感谢阅读，希望能帮助到大家，谢谢大家对本站的支持！

#### 您可能感兴趣的文章:

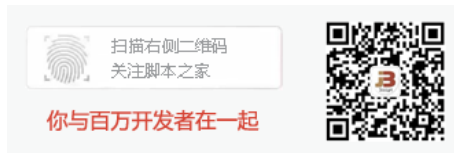
- [Java 进程执行外部程序造成阻塞的一种原因](#)
- [Java基于Runtime调用外部程序出现阻塞的解决方法](#)
- [解析Java的Spring框架的BeanPostProcessor发布处理器](#)
- [oracle如何使用java source调用外部程序](#)
- [Java Process类的详解及实例代码](#)
- [Java Runtime类详解\\_动力节点Java学院整理](#)



Java异常处理运行时异常 ( RuntimeException ) 详解及实例

Java Process详解及实例

Java编程使用Runtime和Process类运行外部程序的方法



微信公众号搜索“脚本之家”，选择关注  
业界资讯、不定期红包、送书等活动等着你

Java 常见异常 RuntimeException

## ERROR

### The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: <http://pos.baidu.com/s?>

#### Read Error

The system returned: (104) Connection reset by peer

An error condition occurred while reading data from the network. Please retry your request.

Your cache administrator is [root](#)

#### 相关文章

## ERROR

### The requested URL could not be retrieved



#### 如何让Jackson JSON生成的数据包含的中文以unicode方式编码

Jackson JSON unicode

2013-12-12



#### java基于jcifs.smb实现远程发送文件到服务器

java 远程发送 服务器

2018-01-01



#### java 学习笔记(入门篇)\_java的安装与配置

java学习笔记 java安装 java配置

2013-01-01



#### 解析WeakHashMap与HashMap的区别详解

WeakHashMap HashMap

2013-05-05



#### 将Sublime Text 2配置为Java的IDE的教程

Sublime Java

2015-07-07



## Java进阶教程之IO基础

Java IO

2014-09-09



## 浅谈JAVASE单例设计模式

JAVASE 单例设计模式

2015-11-11



## java实现文件读写与压缩实例

java 文件

2014-07-07



## 基于spring+hibernate+JQuery开发之电子相册(附源码下载)

spring hibernate JQuery

2013-05-05



## Java Process类的详解及实例代码

Java Process

2017-02-02

# ERROR

## The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: <http://pos.baidu.com/s?>

### Read Error

The system returned: (104) Connection reset by peer

An error condition occurred while reading data from the network. Please retry your request.

Your cache administrator is [root](#)

### 最新评论

评论(0人参与, 0条评论)



来说两句吧...

微博登录

QQ登录

手机登录

还没有评论, 快来抢沙发吧!

Powered by 畅言

关于我们 - 广告合作 - 联系我们 - 免责声明 - 网站地图 - 投诉建议 - 在线投稿



苏公网安备 32031102000137号 苏ICP备14036222号

©CopyRight 2006-2018 JB51.Net Inc All Rights Reserved. 脚本之家 版权所有

