

基于 JAVA 语言的异常处理探讨分析

杨波

成都九洲电子信息系统股份有限公司, 四川 成都 610000

[摘要] java 语言的关键性技术之一就是异常处理, 这一该技术能够快速、准确地识别和处理语言程序中的异常, 从而保持 Java 程序语言的足够可靠性。在提出 Java 语言异常处理的优点后, 对 Java 语言可能出现的异常进行分类, 对于异常处理技术的原理进行解释, 并简单分析了两种异常处理方式。本文从 JAVA 语言的概念和特点等方面, 对 JAVA 语言的异常进行了分析, 阐述了 JAVA 语言异常种类, 介绍了几种常见异常处理方式。

[关键词] JAVA 语言; 软件编程; 异常; 异常处理

DOI: 10.33142/sca.v5i2.6150

中图分类号: TP312JA

文献标识码: A

Discussion and Analysis of Exception Handling Based on JAVA Language

YANG Bo

Chengdu Jiuzhou Electronic Information System Co., Ltd., Chengdu, Sichuan, 610000, China

Abstract: One of the key technologies of Java language is exception handling, which can quickly and accurately identify and handle exceptions in language programs, so as to maintain the sufficient reliability of Java program language. After putting forward the advantages of exception handling in JAVA language, this paper classifies the possible exceptions in Java language, explains the principle of exception handling technology, and briefly analyzes two exception handling methods. From the concept and characteristics of Java language, this paper analyzes the exceptions of JAVA language, expounds the types of JAVA language exceptions, and introduces several common exception handling methods.

Keywords: JAVA language; software programming; abnormal; exception handling

1 JAVA 语言介绍

1.1 Java 语言概念

Java 语言, 是目前编程中最成熟的语言之一。在设计 Java 程序的过程中, 程序员应该在语言使用中保持严谨。不能出差错。但是, 在实际的应用中不能够精确的保证。例如, 在运行程序的过程中, 可能存在指定文件的位置出现错误, 文件名不一致、数组范围越界、0 和其他不可用的除数而导致错误报告。Java 语言的发展, 研究了方便的异常处理技术, 能够快速、准确地识别和处理语言程序中的异常, 从而保持 Java 程序语言足够的稳定性、可靠性。

1.2 JAVA 语言面向对象程序设计的特点

Java 语言有着广泛的应用, 是市场编程的主流语言之一。用 java 语言描述数据结构, 编程思路清晰灵活。例如, 只有泛型类可以用来描述复杂的数据结构, 这大大提高了程序的通用性和可重用性。Java 语言放弃了多重继承机制, 具有自动垃圾收集机制, 降低了编程的复杂性; 同时, Java 语言提供了丰富的系统类和接口。例如, 使用 java 集合类可以轻松描述和操作链表。

面向对象方法逐渐的渗透到了计算机的领域当中, 包括各个分支当中。面向对象方法针对的包括一些较为详细的技术, 软件开发策略, 还包括采用较为完整的方式认识

软件系统和现实世界。主要是从哪些方面和角度深入问题, 并且解决问题, 利用软件系统等。Java 编程语言的核心功能是处理“对象”。^[5]编程语言中提到的对象, 应该从两个方面来理解和操作: 状态参数和操作过程。一般来说, Java 编程中有三种处理对象的方法: 封装、继承和多态。封装是将处理后的对象属性私有化, 并在处理后添加 set 和 get 等程序; 继承可分为“状态继承”和“操作继承”。状态继承说明访问权限不是自由的, 尤其是当子类面对父类时, 操作继承的重点是操作的定义; 多态方法指的是同一对象不少于两个响应的相同办法。这三种方法相辅相成^[6]。

1.2.1 面向对象编程

Java 语言具有面向对象和跨平台的特点。信息技术发展和应用的主要目标是提高工作和生活的便利性。Java 可以在许多编程语言中脱颖而出, 因为它使用起来非常方便和简单。在计算机软件开发过程中, 相应人员可以使用 Java 的源代码转换并形成相应的二进制节点代码, 从而促进其在不同领域的合理、稳定使用, 提高不同领域软件开发人员的效率, 促进从复杂到简单的变革和优化的全面完成。Java 语言是迄今为止最好的面向对象语言。编程时, 代码和数据可以写在每个对象上, 这不仅简化了思维和操作, 而且提高了程序的安全性。

1.2.2 独立平台

Java 编程语言广泛适用的主要原因是它有自己独立的平台。由于这方面的特点，它在编写过程中不受其他程序操作的干扰，从而保证了编写过程的良好独立性、快速性和整体稳定性。这方面显示的特性也是其他编程语言所没有的优点。平台独立性是 Java 语言的一个关键特性。此功能的主要体现是 Java 虚拟机。Java 编程语言的独立平台功能清晰，符合平台编程环境的相关标准和体系。Java 编程能否顺利进行，取决于 Java 虚拟机在编程过程中能否正常运行。同时，Java 语言在应用过程中具有很强的独立性。例如，在计算机软件开发过程中，用户可以使用 Java 语言将软件开发的所有步骤都放在 Java 虚拟机的链接中，并根据 Java 虚拟机的运行情况实时检测计算机软件的运行状态。Java 编程语言的这种运行模式使用户摆脱了传统软件开发的束缚，最终保证了软件开发的质量和结果。

1.2.3 移植作用

一般来说，Java 编程语言可以在图像、图形和 HTML 等网站上浏览、下载所需的工具。浏览器在接收到一定的指令之后进行工作，并且实现一定的资源下载工作。在 Java 编程中使用这些工具能够提升计算机的实用性、流畅性，并且能够为数据设置更高的要求。计算机软件开发是一项系统工程，通常有许多操作阶段。从整体的角度进行开发和编制，不仅增加了相应人员需要承担的工作量，而且找不到合适的切入点，导致工作效率不断降低，耽误了正常进度。通过使用 Java 编程语言，可以科学地处理上述问题。技术人员在编写程序时主要使用模块化程序，并对整个程序进行大量分解，以便负责编程的人员可以对其中的一些程序进行操作，以防止混乱。此外，Java 编程语言具有子模块连接的优势，可以确保开发工作能够真正有效地实现。

1.2.4 多线程

与其他编程语言相比，Java 具有良好的开放性和包容性，使其可扩展性不断增强和扩展。如果应用软件想要正常工作，它不能与具有多线程功能的程序分离。要采用 Java 语言的同步源代码，更好的提高数据的准确，保持数据的共享和优势。除此以外，Java 业提升了计算机使用人员的专业。利用多线程控制的情况下，实现更多行为模式。因此，开展网络交互可以逐步降低软件开发的难度。

1.2.5 灵活性和安全性

首先，在灵活性方面，与其他编程语言相比，Java 语言的编写结构更简单。它没有太多的操作符号或复杂的程序。使用 java 语言是最及时的，可以在一定程度上保证计算机软件的开发质量和效率。Java 语言不会影响计算机系统中程序的正常运行，但可以提高计算机软件的开发效率。因此，人们可以同时更有效地使用计算机。其次，

从安全性方面来看，使用 Java 可以处理和开发计算机软件，但这需要基于公钥技术的先进和前瞻性功能，这也是计算机软件开发中最大的创新。为了使计算机软件系统具有较高的安全性，我们需要分析 Java meta 的适用性。一旦出现外部访问问题，将自动弹出警告消息，防止外部访问。通过这种方式，它可以确保计算机内部的数据信息不会受到来源的威胁。

2 Java 语言异常的结构与分类

在使用计算机编程语言的过程中，Java 被称为该领域的“宠儿”，因为它的应用范围非常广泛，可以在各种不同的平台上稳定运行。它不仅是最常用的，也是编程中最有效的。即便如此，Java 语言中仍可能出现异常。

Java 语言异常的定义，即因程序执行错误导致的异常对象。它通常对应于 Java 语言中异常处理的技术机制。Java 语言异常处理技术机制可用于识别、过滤和解决异常。Java 语言异常处理技术机制具有很高的效率。它不仅可以使用 Java 语言编写的程序更加完善、可靠和健壮，还可以帮助解决异常问题。以前的其他异常处理技术通常用于执行跳转指令或结束程序进程，或在异常发生时调用较低级别的函数处理来处理异常。

当 Java 语言编程过程中出现异常时，我们需要找出问题并编写检测程序。根据问题的不同类型，可以分为两种情况：一种是用 C 语言编写的异常。当这类程序出现问题时，通常使用 if 条件语句和调用函数来编写检测程序。用户可以根据函数的返回值判断程序的问题类型；另一个是由 Java 语言编写的程序引起的异常问题。使用的检测方法是异常调用。

2.1 Java 语言异常的结构

Java 中系统类 throwable 的子类覆盖了系统中的所有异常类。project 对象生成 throwable，然后 throwable 生成两个子类：exception 和 error。异常主要指在程序运行过程中可能捕捉到的异常。exception 类还派生自 runtimeException，这是一个极其重要的子类。在 Java 中，exception 类和 runtimeException 类被称为 Java 程序运行期间生成的异常。例如，除数零或下标超出范围等异常是在操作过程中生成的错误。错误通常是由系统异常引起的，这些异常不仅不可恢复，而且不愿意被程序检查和获取。主要指 Java 虚拟机 (JVM) 异常或系统异常，代表严重的系统异常。

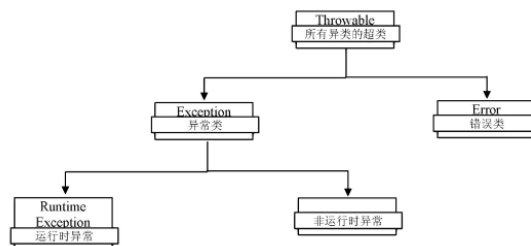


图 1 异常的结构

2.2 Java 语言异常的分类

Java 语言异常有两种类型,一种是系统定义的,另一种是用户定义的。系统定义的异常类型很多,主要包括非法输入输出导致的 `IOException`、数据类型匹配导致的 `NumberFormatException` 异常、算术错误导致的算术异常,如除数 0 导致的算术异常、下标超出范围导致的 `IndexOutOfBoundsException`、中断导致的 `InterruptedException`、等,当系统定义的异常类型不能满足用户需求时,用户自行定义异常。这个过程异常对象由程序员根据程序的需要创建。为了使程序自动抛出异常,程序员在创建过程中首先应该创建异常的类型。使用 `exception` 创建一个新类。通过继承,创建的子类可以拥有与父类相同的处理代码和接口。

2.3 常见异常和错误

(1) 可丢弃: 定义为两个子类: 异常和错误; 在异常和错误情况下, 分别分布多个子类别。

(2) 错误: 程序之外的异常, 通常发生在系统或虚拟机中, 程序本身无法捕获。

(3) 异常: 程序本身可以处理的异常。

(4) `RuntimeException`: JVM 工作过程中的异常, 如空指针异常。

(5) 已检查: 指与异常、错误、运行时异常等不同的所有其他异常。此类异常发生在程序之外, 无法由程序本身捕获和处理。因此, 在编译过程中需要对它们进行检查。

(6) 未选中: 包括运行时异常、运行时异常及其子类和错误。

3 异常处理方式

在现阶段, Java 编程语言中存在不同的例外情况。因此, 需要根据实际使用处理公司的实际情况。异常处理方法主要包括抛出异常和捕获并处理异常。当程序遇到当前环境无法处理的问题时, 它会将问题抛出到上层, 并寻找能够解决问题的环境, 直到找到并执行包含适当异常处理的方法。如果找不到, 则终止程序。

在开发计算机软件时, 针对特定的开发内容生成 Java 编程语言的异常处理机制, 并且存在一对一的对应关系。这种机制需要在处理 Java 编程语言的异常问题时预先设置可能的异常类型。在设置异常问题的前提下, 可以发挥程序异常处理机制的功能, 对满足异常处理机制触发条件的异常问题采取应对措施, 确保计算机软件的正常运行。这也表明, 在使用 java 编程语言时, 设计者应充分利用其灵活性和可扩展性的优势, 以便在程序出现异常问题时采取积极有效的措施。

通用 Java 程序在运行过程中具有故障诊断和故障修复功能。通过使用 Java 编程语言, 可以科学地处理上述问题。在编写程序时, 技术人员主要使用模块化程序, 并对整个程序进行大量分解, 以便负责编程的人员可以对其

中一些程序进行操作, 以防止混乱。Java 语言的强大实用性也体现在异常的发生上。Java 编程语言具有子模块连接的优势, 可以保证开发工作能够真正有效地实现。

在 Java 语言的异常处理过程中, Java 语言不是将程序作为一个整体来编写, 而是采用不同的部分进行划分模式, 针对不同的模板, 在程序编码中找到相互异常问题对应的方案。在处理异常问题时, Java 语言能够保证不同的模块之间的独立性, 提升工作的效率。因为 Java 语言是封闭的和多样性的特点。因此, 在编写一个模块时, 不会影响到其他模块的程序。

3.1 抛出异常

当程序中发生错误或异常时, 系统将创建一个具有异常类型和程序状态的异常对象, 同时将这个系统返回到正在运行的系统。运行的系统根据创建的异常对象找出处理代码, 针对不符合的代码, 对此进行修改, 之后继续执行。并对其进行修改以执行。一旦程序出现异常, 并且不能够处理时, 这个异常将会被上层的调用函数进行修改。

3.2 捕获异常

试试 {...} 抓住 {...} 最后 {...} Java 规定必须捕获或声明才能抛出可验证的异常。允许忽略不可跟踪的运行时异常和错误。Try:Try-in-Java 是为了确保程序不会崩溃, 在异常发生后可以正常工作。当函数中出现 try 语句时, 需要编写另一个语句来保护它, 直到 try 语句中的所有异常都得到处理。不会生成新的 try 语句。Catch: 捕捉 try 语句块中的异常。最后: 不管发生什么, 它都会被执行。如果 try 中的所有语句都已执行, 则进入 finally 阶段。如果在最后阶段没有例外, 整个试捕就最终完成了。如果在 finally 阶段发生异常, 它将被 catch 再次捕获, 并且 return 将在 try 语句块中重新执行, 直到没有生成新的 try 语句, 并且该语句的执行完成并直接终止。

4 结论

Java 语言异常处理技术将正确的代码从异常代码中分离出来, 这显示了 Java 语言异常处理技术的优势和先进性。Java 语言异常处理技术机制具有很高的效率。它不仅可以使用 Java 语言编写的程序更加完善、可靠和健壮, 还可以帮助解决异常问题。Java 语言的发展, 更加方便了对于一些异常的处理, 保证更加快速, 识别出语言系统中的异常, 以此, 保证 JAVA 语言的稳定性。Java 语言异常处理技术是可以保证识别出错误代码和正常代码, 并分别进行处理, 具有一定的优势。当系统处理一些异常时, 必须注意程序计算的增加可能会降低计算机的运行速度。因此, 对于大多数程序员来说, 应该认真对待用什么方法来完成异常处理, 并在异常处理中保持科学合理性。

[参考文献]

[1] 袁小英. 基于 Java 语言的异常处理机制的研究[J]. 计算机光盘软件与应用, 2011(16): 115-158.

[2] 韩瑞峰. Java 异常处理机制及应用研究[J]. 忻州师范学院学报, 2012, 28(2): 25-27.
[3] 夏宇. 基于 JAVA 常见异常处理研究[J]. 计算机光盘软件与应用, 2012(8): 183-182.
[4] 葛建霞. JAVA 语言在远程网络教学中的应用[J]. 河南

职业技术师范学院学报, 2002(4): 76.

[5] 魏林. 《Java 程序设计》教材二次开发的多元化策略研究[J]. 电脑知识与技术, 2019(32): 76.

作者简介: 杨波 (1985-) 男, 毕业院校, 四川大学, 学历: 本科, 所学专业, 计算机科学与技术。