



计算机与信息科学系列规划教材

JISUANJI YU XINXI KEXUE XILIE GUIHUA JIAOCAI

Java

核心技术

周忠宝 郑龙 王浩 周滔 编著

湖南大学出版社



责任编辑 黄 旺

封面设计  风格八號
www.fy8.com



打开手机淘宝
扫一扫



湖南大学出版社
当当网图书旗舰店

ISBN 978-7-5667-1767-2



9 787566 717672 >

定价：68.00元

计算机与信息科学系列规划教材

Java 核心技术

编 著 周忠宝 郑 龙
王 浩 周 滔

湖南大学 出版社

内 容 简 介

本书对 Java 核心技术进行全面、详细的讲解。全书共 10 章理论知识和 10 个上机实操,主要介绍了 Java 语言的发展与运行机制、Java 语言基础、类和对象、继承与多态、Java 基础类的应用、集合框架、异常处理、文件输入与输出、Java 的高级特性、JDBC 基本应用等知识,且每章都配有丰富的实例、要点和作业,帮助读者理解和掌握书中的内容。本书适合作为计算机相关专业“Java 技术”课程的进阶培训教材,也可以作为程序设计员或对 Java 技术感兴趣读者的参考用书。

图书在版编目(CIP)数据

Java 核心技术/周忠宝等编著. —长沙:湖南大学出版社,2019. 11
(计算机与信息科学系列规划教材)
ISBN 978-7-5667-1767-2

I. ①J… II. ①周… III. ①JAVA 语言—程序设计—高等学校—教材
IV. ①TP312. 8

中国版本图书馆 CIP 数据核字(2019)第 167789 号

Java 核心技术

Java HEXIN JISHU

编 著:周忠宝 郑 龙 王 浩 周 滔
责任编辑:黄 旺 责任校对:尚楠欣
印 装:北京虎彩文化传播有限公司
开 本:787mm×1092mm 16 开 印张:23.25 字数:552 千
版 次:2019 年 11 月第 1 版 印次:2019 年 11 月第 1 次印刷
书 号:ISBN 978-7-5667-1767-2
定 价:68.00 元

出 版 人:雷 鸣
出版发行:湖南大学出版社
社 址:湖南·长沙·岳麓山 邮 编:410082
电 话:0731-88822559(发行部),88820006(编辑室),88821006(出版部)
传 真:0731-88649312(发行部),88822264(总编室)
网 址:<http://www.hnupress.com>
电子邮箱:274398748@qq.com

版权所有,盗版必究

湖南大学版图书凡有印装差错,请与发行部联系

计算机与信息科学系列规划教材 编委会

主任:周忠宝

编委:周忠宝 郑 龙 何敏藩

邢立宁 罗俊海 姚 锋

叶朝晖 邓劲生 姚煊道

邹 伟 王 浩 张 章

肖 丹 蔡 琴 付 艳

周 滔 周 舟

编著与设计单位:湖南大学

国防科技大学

电子科技大学

佛山科学技术学院

长沙学院

深圳华大乐业教育科技有限公司

前言

时光荏苒,一转眼中国互联网已走过了 30 余年的历程。人工智能、云计算、移动支付,这些互联网产物不仅迅速占据了我们的生活,刷新了我们对科技发展的认知,而且也提高了我们的生活质量。人们谈论的话题也离不开这些,例如:人工智能是否会替代人类成为工作的主要劳动力;数字货币是否会代替纸币流通于市场;虚拟现实体验到底会有多真实多刺激。从这些现象中不难发现,互联网的辐射面在不断扩大,计算机科学与信息技术发展的普适性在不断增强,信息技术润物无声地全面融入并颠覆了我们的生活。

1987 年,我国网络专家钱天白通过拨号方式在国际互联网上发出了中国有史以来第一封电子邮件,“越过长城,走向世界”,从此,我国互联网时代开启。30 余年间,人类社会仍然遵循着万物自然生长规律,但互联网的枝芽却依托人类的智慧于内部结构中迅速蔓延,并且每一次主流设备、主流技术的迭代速度在明显加快。如今,人们的生活是“拇指在手机屏幕方寸间游走的距离,已经超过双脚走过的路程”。

据估计,截至 2017 年 6 月,中国网民规模已达到 7.5 亿人,占全球网民总数的五分之一,而且这个数字还在不断地增加。

然而,面对快速发展的互联网,每一个互联网人亦感到焦虑,感觉它运转的速度已经接近我们追赶的极限。信息时刻在更新,科技不断被颠覆,想象力也一直被挑战,面对这些,人们感到不安的同时又对未来的互联网充满期待。

互联网的魅力正在于此,恰如山之两面,一面阴暗晦涩,一面生机勃勃,一旦跨过山之巅峰,即是不一样的风景。正是这样的挑战会让人着迷,并甘愿为之付出努力。而这个行业还有很多伟大的事情值得去琢磨,去付出自己的匠心。

本系列丛书作为计算机科学与信息科学中的入门与提高教材,在力争保障学科知识广度的同时,也统筹主流技术的深度,既介绍了计算机学科相关主题的历史,也涵盖了国内外最新、最热门课题,充分呈现了计算机科学技术的时效性、前沿性。丛书涉及计算机与信息科学多门课程:Java 程序设计与开发、C# 与 WinForm 程序设计、SQL Server 数据库、Oracle 大型数据库、Spring 框架应用开发、Android 手机 APP 开发、JDBC/Jsp/Servlet 系统开发等;HTML/CSS 前端数据展示、jQuery 前端框架、JavaScript 页面交互效果实现等;大数据基础与应用、大数据技术概论、R 语言预测、PRESTO 技术内幕等;Photoshop 制作与视觉效果设计、网页 UI 美工设计、移动端 UI 视觉效果设计与运用、CorelDraw 设计与创新等。

本系列丛书适合初学者,当然掌握一些计算机基础知识更有利于本系列丛书的学习。开发人员可从本系列丛书中找到许多不同领域的兴趣点和各种知识点的用法。丛书实例内容选取市场流行的应用项目或产品项目,章后部分练习题模拟了大型软件开发企业的实例项目。

本系列丛书在编写过程中,获得了国家自然科学基金委员会与中国民用航空局联合资助项目(U1733110)、湖南省科学“十三五”规划课题(XJK016BGD009)、湖南省教学改革研究课题(2015001)、湖南省自然科学基金(2017JJ1012)、国家自然科学基金(71371067)的资助,并得到了湖南大学、国防科技大学、电子科技大学、佛山科学技术学院、长沙学院和深圳华大乐业教育科技有限公司各位老师的大力支持,同时参考了一些相关著作和文献,在此向这些老师和文献作者深表感谢。

未来互联网信息技术已扑面而来,汹涌胜于往昔,你做好准备了吗?

作者
2017年9月

目次

理论部分

第1章 Java语言简介	2
1.1 Java的由来	2
1.2 Java程序开发环境	7
1.3 JDK(Java开发工具包)	8
1.4 编写Java应用程序	11
1.5 分析Java程序	12
1.6 Java API文档	15
第2章 Java语言基础	17
2.1 Java数据类型.....	18
2.2 变量.....	23
2.3 运算符和表达式.....	26
2.4 Scanner(扫描仪)	33
2.5 选择结构语句.....	33
2.6 循环结构语句.....	37
2.7 数组.....	44
第3章 类和对象	51
3.1 面向对象的基础.....	51
3.2 对象.....	54
3.3 类.....	55
3.4 类的实现.....	57
3.5 类的封装性.....	66
3.6 构造方法.....	69
3.7 对象数组的使用.....	74
3.8 包.....	74
第4章 继承与多态	78
4.1 继承.....	78
4.2 多态.....	84

4.3	权限修饰符	94
4.4	方法修饰符	96
4.5	接口	108
第 5 章	Java 基础类的应用	118
5.1	Java 基础包	118
5.2	包装类	119
5.3	Object 类	122
5.4	String, StringBuffer, StringBuilder 类	124
5.5	System 类	134
5.6	Runtime 类	136
5.7	Math 类	139
5.8	Random 类	141
5.9	日期处理类	144
第 6 章	集合框架	155
6.1	Java 集合框架概述	156
6.2	类集接口	157
6.3	通过迭代方法访问类集	166
6.4	映射接口	168
第 7 章	异常处理	173
7.1	Java 异常	173
7.2	处理 Java 程序中的异常	177
7.3	抛出异常	181
7.4	自定义异常	184
第 8 章	文件输入与输出	190
8.1	File 类的使用	190
8.2	流的概念	194
8.3	字节流和字符流	195
8.4	输入/输出流体系	200
第 9 章	Java 的高级特性	207
9.1	泛型	207
9.2	注解	216
9.3	增强 for 循环	217
9.4	可变参数	219
9.5	枚举	220
9.6	静态导入	222
9.7	自动装箱和拆箱	223

第 10 章 JDBC 基本应用	226
10.1 JDBC 技术概述	226
10.2 JDBC API	228
10.3 JDBC 操作数据库	233
10.4 JDBC 的高级应用	238

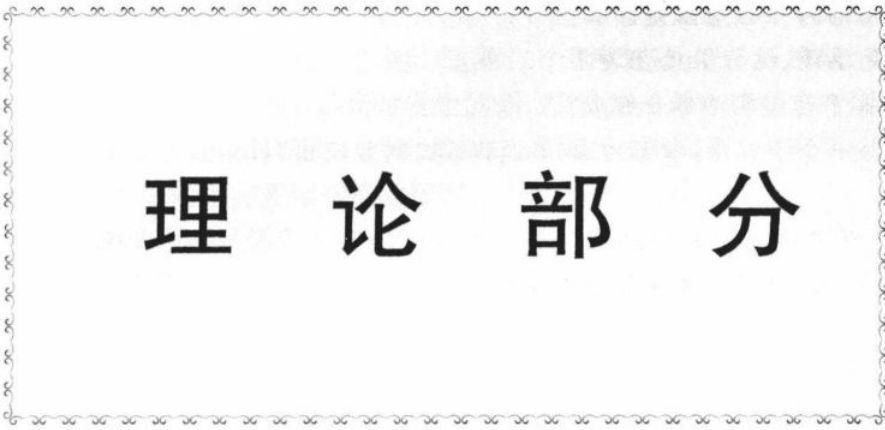
上机部分

上机 1 Java 语言简介	246
第 1 阶段 指导	246
第 2 阶段 练习	251
上机 2 Java 语言基础	252
第 1 阶段 指导	252
第 2 阶段 练习	260
上机 3 类和对象	261
第 1 阶段 指导	261
第 2 阶段 练习	266
上机 4 继承与多态	267
第 1 阶段 指导	267
第 2 阶段 练习	272
上机 5 Java 基础类的应用	274
第 1 阶段 指导	274
第 2 阶段 练习	282
上机 6 集合框架	283
第 1 阶段 指导	283
第 2 阶段 练习	291
上机 7 异常处理	292
第 1 阶段 指导	292
第 2 阶段 练习	295
上机 8 文件输入与输出	296
第 1 阶段 指导	296
第 2 阶段 练习	300
上机 9 Java 的高级特性	301
第 1 阶段 指导	301

第 2 阶段 练习	304
上机 10 JDBC 基本应用	305
第 1 阶段 指导	305
第 2 阶段 练习	311

附录部分

附录 1 多线程	314
1.1 线程简介	314
1.2 实现多线程	316
1.3 线程的状态	319
1.4 线程操作的一些方法	320
1.5 多线程的同步以及通信	324
1.6 线程的声明周期	330
1.7 线程的优先级	330
附录 2 Java 网络编程	333
2.1 网络基础	333
2.2 网络协议	335
2.3 java.net 包	336
2.4 建立 Socket 网络通信	337
2.5 UDP 协议网络程序	340
附录 3 Java 图形化编程	346
3.1 组件和容器	346
3.2 程序界面布局	354
3.3 事件处理	357

A decorative rectangular border with a repeating scroll-like pattern surrounds the central text.

理 论 部 分

第 1 章 Java 语言简介

预习任务

1. Java 语言发布日期?
2. 如何在电脑上安装 Java 运行环境?
3. Java 语言有哪些特点?
4. Java 程序运行原理是什么?
5. 如何编译、运行 Java 程序?
6. Java 平台由哪两部分组成?

7. 编写一个小程序,命名为 Demo.java,控制台输出“Hello,这是我的第一个 Java 应用程序”。

本章目标

- 了解 Java 语言的发展和历史背景
- 了解 Java 语言的特点
- 了解 Java 运行机制和环境
- 尝试编写第一个 Java 程序

对于计算机语言的发展史,业界一般认为:B 语言导致了 C 语言的诞生,C 语言演变出了 C++ 语言,而 C++ 语言将让位于 Java 语言。要想更好地了解 Java 语言,就必须了解它产生的原因,推动它发展的动力,以及它对其他语言的继承。像以前其他成功的计算机语言一样,Java 继承了其他语言的先进原理,同时又因其独特的环境要求而提出了一些创新性的概念。本书将从实用的角度,对 Java 语言、库及应用程序进行包括语法在内的详细介绍。在本章里,我们将介绍 Java 语言产生的背景、发展过程,以及使它变得如此重要的原因。

1.1 Java 的由来

1.1.1 Java 的产生

Java 是由 James Gosling, Patrick Naughton, Chris Warth, Ed Frank 和 Mike Sheridan 于 1991 年在 Sun Microsystems 公司设计出来的。开发第一个版本花了 18 个月。该语言开始名叫“Oak”,于 1995 年更名为“Java”。从 1992 的秋天 Oak 问世到 1995 的春天公开发布 Java 语言,许多人对 Java 的设计和改进做出了贡献。Bill Joy, Arthur

van Hoff, Jonathan Payne, Frank Yellin 和 Tim Lindholm 是主要的贡献者,正是他们的贡献使最初原型得以成熟。

说起来多少有些令人吃惊,Java 的最初推动力并不是因特网,而是源于对独立于平台(也就是体系结构中立)语言的需要,这种语言可创建能够嵌入微波炉、遥控器等各种家用电器设备的软件。用作控制器的 CPU 芯片是多种多样的,但 C 和 C++ 以及其他绝大多数语言的缺点是只能对特定目标进行编译。尽管为任何类型的 CPU 芯片编译 C++ 程序是可能的,但这样做需要一个完整的以该 CPU 为目标的 C++ 编译器,而创建编译器是一项既耗资巨大又耗时较长的工作。因此需要一种简单且经济的解决方案。为了找到这样一种方案,Gosling 和其他人开始一起致力于开发一种可移植、跨平台的语言,该语言能够生成运行于不同环境、不同 CPU 芯片上的代码。他们的努力最终促成了 Java 的诞生。

在 Java 的一些细节被设计出来的同时,第二个并且也是最重要的因素出现了,该因素将对 Java 的未来起着至关重要的作用。这第二个因素当然就是万维网(WWW)。如果万维网的成型和 Java 的实现不是同时发生的话,那么 Java 可能保持它有用但默默无闻的用于电子消费品编程语言的状态。然而,随着万维网的出现,Java 被推到计算机语言设计的最前沿,因为万维网也需要可移植的程序。

绝大多数程序员在涉足编程领域时就知道可移植的程序像他们的理想一样难以捉摸。尽管人们对高效的、可移植的(独立于平台)编程方式的追寻几乎和编程历史一样久远,但它总是让位于其他的更为紧迫的问题。此外,因为计算机业被 Intel, Macintosh 和 UNIX 这 3 个竞争对手垄断,大多数程序员都在其中的某个领域内长期工作,所以对可移植语言的需求就不是那么迫切。但是,随着因特网和 Web 的出现,关于可移植性语言的旧问题又被提了出来。毕竟,因特网由不同的、分布式的系统组成,其中包括各种类型的计算机、操作系统和 CPU。尽管许多类型的平台都可以与因特网连接,但用户仍希望他们能够运行同样的程序。曾经一个令人烦恼却无需优先考虑的问题现在变成了急需解决的问题。

1993 年,Java 设计小组的成员发现他们在编制嵌入式控制器代码时经常遇到的可移植性问题,在编制因特网代码的过程中也出现了。事实上,开始被设计为解决小范围问题的 Java 语言同样可以被用在大范围的因特网上。这个认识使他们将 Java 的重心由电子消费品转移到 Internet 编程。因此,中立体系结构编程语言的需要是促使 Java 诞生的原动力,而 Internet 却最终导致了 Java 的成功。

正如前面提到的,Java 的大部分特性是从 C 和 C++ 中继承的。Java 设计人员之所以故意这么做,主要是因为他们觉得,在新语言中使用熟悉的 C 语法及模仿 C++ 面向对象的特性,将使他们的语言对经验丰富的 C/C++ 程序员有更大的吸引力。除了表面类似外,其他一些促使 C 和 C++ 成功的因素也帮了 Java 的忙。首先,Java 的设计、测试、精炼由真正从事编程工作的人员完成,它根植于设计它的人员的需要和经验,因而也是一个程序员自己的语言。其次,Java 是紧密结合的且逻辑上是协调一致的。最后,除了那些 Internet 环境强加的约束以外,Java 给了编程人员完全的控制权。如果你程序编得好,你编写的程序就能反映出这一点。相反,如果你的编程手法拙劣,也能在你的程序中反映出

来。换一种说法,Java 并不是训练新手的语言,而是供专业编程人员使用的语言。

Java 和 C++ 之间的相似性容易使人将 Java 简单地想象为 C++ 的高级版本,但其实这是一种误解。Java 在实践和理论上都与 C++ 有重要的不同点。尽管 Java 受到 C++ 的影响,但它并不是 C++ 的增强版。例如,Java 与 C++ 既不向上兼容,也不向下兼容。当然,Java 与 C++ 的相似之处也是很多的,如果你是一个 C++ 程序员,你会感觉到对 Java 非常熟悉。另外一点是:Java 并不是用来取代 C++ 的,设计 Java 是为了解决某些特定的问题,而设计 C++ 是为了解决另外一类完全不同的问题。两者将长时间共存。

环境的变化促使 Java 这种独立于平台的语言注定成为 Internet 上的分布式编程语言。同时,Java 也改变了人们的编程方式,特别是 Java 对 C++ 使用的面向对象范例进行的增强和完善。所以,Java 不是孤立存在的一种语言,而是计算机语言多年来的演变结果。仅这个事实就足以证明 Java 在计算机语言历史上的地位。Java 对 Internet 编程的影响就如同 C 语言对系统编程的影响一样:革命的力量将改变世界。

1.1.2 Java 语言的特点

不介绍 Java 常用语,对 Java 的总体介绍就是不完整的。尽管促使 Java 诞生的原动力是可移植性和安全性,但在 Java 语言最终成型的过程中,其他一些因素也起了重要的作用。Java 设计开发小组的成员总结了这些关键因素,称其为 Java 的专门用语,包括:简单(simple)、安全(secure)、可移植(portable)、面向对象(object-oriented)、健壮(robust)、多线程(multithreaded)、体系结构中立(architecture-neutral)、解释执行(interpreted)、高性能(high performance)、分布式(distributed)、动态(dynamic)。

下面让我们看看这些特性的含义。

1. 简单

Java 的设计目的是让专业程序员觉得既易学又好用。如果你有编程经历,你将不觉得 Java 难掌握。如果你已经理解面向对象编程的基本概念,学习 Java 将更容易。如果你是一个经验丰富的 C++ 程序员,那就最好了,学习 Java 简直不费吹灰之力。因为 Java 继承了 C/C++ 语法和许多 C++ 面向对象的特性,大多数程序员在学习 Java 时都不会觉得太难。而且它省略了 C++ 语言中所有的难以理解、容易混淆的特性,如头文件、指针、结构、单元、运算符重载和虚拟基础类等,因此更加严谨、简洁。

2. 安全

正如你知道的那样,每次当你下载一个“正常”的程序时,你都要冒着被病毒感染的危险。在 Java 出现以前,大多数用户并不经常下载可执行的程序文件,即使下载了程序,在运行它们以前也都要进行病毒检查。尽管如此,大多数用户还是担心他们的系统可能被病毒感染。除了病毒,另一种恶意的程序也必须警惕。这种恶意的程序可通过搜索你计算机本地文件系统的内容来收集你的私人信息,例如信用卡号码、银行账户结算口令。Java 在网络应用程序和你的计算机之间提供了一道防火墙(firewall),消除了你的这些顾虑。

当使用一个兼容 Java 的 Web 浏览器时,你可以安全地下载 Java 小应用程序,不必担

心病毒的感染或恶意的企图。Java 实现这种保护功能的方式是,将 Java 程序限制在 Java 运行环境中,不允许它访问计算机的其他部分,后面将介绍这个过程是如何实现的。下载小应用程序并能确保它对客户机的安全性不会造成危害是 Java 的一个最重要的方面。

3. 可移植

许多类型的计算机和操作系统都连接到 Internet 上。要使连接到 Internet 上的各种各样的平台都能动态下载同一个程序,就需要有能够生成可移植性执行代码的方法。很快你将会看到,有助于保证安全性的机制同样也有助于建立可移植性。实际上,Java 对这两个问题的解决方案是完美的也是高效的。

4. 面向对象

尽管受到其前辈的影响,但 Java 没被设计成兼容其他语言源代码的程序。这允许 Java 开发组自由地从零开始。这样做的一个结果是,Java 语言可以更直接、更易用、更实际地接近对象。通过对近几十年面向对象软件优点的借鉴,Java 在纯进化论者的“任何事物都是一个对象”和实用主义者的“不讨论对象不对象”的论点之间找到了平衡。Java 的对象模型既简单又容易扩展,对于简单数据类型,例如整数,它保持了高性能,但简单数据类型不是对象。

5. 健壮

万维网上多平台的环境使得它对程序有特别的要求,因为程序必须在许多系统上可靠地执行。这样,在设计 Java 时,创建健壮的程序被放到了高度优先考虑的地位。为了获得可靠性,Java 在一些关键的地方限制你,强迫你在程序开发过程中及早发现错误。同时,Java 使你不必担心许多常见的引起编程错误的问题。因为 Java 是一种严格的类型语言,它不但在编译时检查代码,而且在运行时也检查代码。事实上,在运行时经常碰到的难以重现的、难以跟踪的错误在 Java 中几乎是不可能产生的。要知道,使程序在不同的运行环境中以可预见的方式运行是 Java 的关键特性。

为更好理解 Java 是如何具有健壮性的,让我们考虑使程序失败的两个主要原因:内存管理错误和误操作引起的异常情况(也就是运行时错误)。在传统的编程环境下,内存管理是一项困难、乏味的任务。例如,在 C/C++ 中,程序员必须手动地分配并且释放所有的动态内存。这有时会导致问题,因为程序员可能忘记释放原来分配的内存,或者释放了其他部分程序正在使用的内存。Java 通过替你管理内存分配和释放,可以从根本上消除这些问题(事实上,释放内存是完全自动的,因为 Java 为闲置的对象提供内存垃圾自动收集服务)。在传统的环境下,异常情况可能经常由“被零除”或“文件未找到”这样的情况引起,而我们又必须用既繁多又难以理解的一大堆指令来对它们进行管理。Java 通过提供面向对象的异常处理机制来解决这个问题。一个写得好的 Java 程序,所有的运行时错误都可以并且应该被其进行管理。

6. 多线程

Java 语言支持多线程机制,多线程机制使得 Java 程序能够并行处理多项任务。Java 程序可以设计成具有多个线程,例如让一个线程负责数据的检索、查寻,另一个线程与用户进行交互,这样两个线程得以并行执行。多线程机制可以很容易地实现网络上的交互式操作。

7. 体系结构中立

Java 设计者考虑的一个主要问题是程序代码的持久性和可移植性。程序员面临的一个主要问题是,不能保证今天编写的程序明天能否在同一台机器上顺利运行。操作系统升级、处理器升级以及核心系统资源的变化,都可能导致程序无法继续运行。Java 设计者对这个问题做过多种尝试,Java 虚拟机(JVM)就是试图解决这个问题的。他们的目标是“只要写一次程序,在任何地方、任何时间该程序永远都能运行”。在很大程度上,Java 实现了这个目标。

8. 解释执行

通过把程序编译为 Java 字节码这样一个中间过程,Java 可以产生跨平台运行的程序。字节码可以在提供 Java 虚拟机(JVM)的任何一种系统上被解释执行。早先的许多尝试解决跨平台的方案对性能要求都很高。其他解释执行的语言系统,如 BASIC, Tcl, PERL 都有无法克服的性能缺陷。然而,Java 却可以在非常低档的 CPU 上顺利运行。前面已解释过,Java 确实是一种解释性语言,Java 的字节码经过仔细设计,因而很容易使用 JIT 编译技术将字节码直接转换成高性能的本机代码。Java 运行时系统在提供这个特性的同时仍具有平台独立性,因而“高效且跨平台”对 Java 来说不再矛盾。

9. 分布式

Java 为 Internet 的分布式环境而设计,因为它处理 TCP/IP 协议。事实上,通过 URL 地址存取资源与直接存取一个文件的差别是不太大的。Java 原来的版本(Oak)包括了内置的地址空格消息传递(intra-address-space)特性。这允许位于两台不同的计算机上的对象可以远程地执行过程。Java 还发布了叫作远程方法调用(remote method invocation, RMI)的软件包,这个特性使客户机/服务器编程达到了无与伦比的抽象级。

10. 动态

Java 程序带有多种的运行时类型信息,用于在运行时校验和解决对象访问问题。这使得在一种安全、有效的方式下动态地连接代码成为可能,对小应用程序环境的健壮性也十分重要,因为在运行系统中,字节码内的小段程序可以动态地被更新。

1.1.3 Java 的发展

Java 的最初发布本不亚于一场革命,但是它并不标志着 Java 快速革新时代的结束。与大多数其他软件系统经常进行小的改进不同,Java 继续以爆炸式的步伐向前发展。在 Java1.0 发布不久,Java 的设计者已经创造出了 Java1.1。Java1.1 新增的特性远比普通意义上的版本修订有意义,内容要丰富许多。Java1.1 增加了许多新的库元素,重新定义了小应用程序处理事件的方法,并且重新设置了 1.0 版中库的许多特性。它也放弃了原来由 Java1.0 定义的若干过时的特征。因此,Java1.1 不但增加了 Java1.0 中没有的属性,同时也抛弃了一些原有的属性。

Java 的第二个主要发布版本是 Java2。Java2 是一个分水岭,它的出现标志快速演变语言“现代时代”的开始。Java2 第一版本的版本号是 1.2。这似乎有点奇怪。原因是它参考了原来 Java 库的版本,对于整个版本来说,它本身没有多大变化。Java2 增加了很多对新特性的支持,例如 Swing 和类集框架,并且它提高了 Java 虚拟机和各种编程工具的