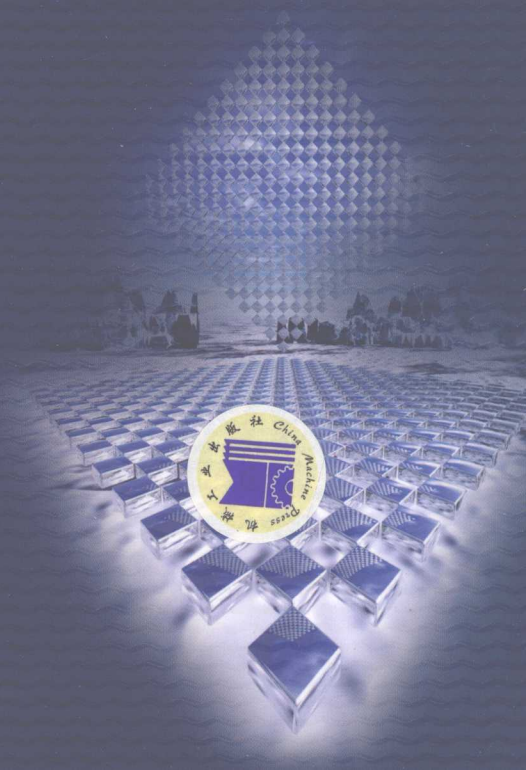
免费提供电子教案

高等院校规划教材
计算机科学与技术系列

Java技术应用基础
——对象·模式·虚拟机

任 哲 等编著



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

Java 技术应用基础

——对象·模式·虚拟机

任 哲 等编著

机械工业出版社

本书是普通高等学校在学生学习了微机原理、C/C++程序设计、操作系统和编译原理之后的 Java 技术课程教学用书。本书以介绍面向对象程序设计思想和方法为目标，以设计模式为线索，重点介绍面向抽象编程技术及 Java 技术的主要特点。主要内容为：Java 语言，虚拟机对 Java 性能的支持，设计模式在 Java 中的应用，Java Beans 及其事件处理机制。

本书从程序设计的角度比较全面地介绍了 Java 的核心技术和核心思想，并尽可能地涵盖当今先进程序设计理念。本书的特点是：说理性强，文字简练、通俗，配有适当数量的例题及源代码（可从 www.cmpedu.com 下载）。

本书适合作为普通高等学校程序设计的总结课教材，也可以作为 IT 企业的 Java 技术初级培训教材，以及工程技术人员的参考书。

图书在版编目(CIP)数据

Java 技术应用基础—对象·模式·虚拟机/任哲等编著. —北京：机械工业出版社，2009.2
（高等院校规划教材·计算机科学与技术系列）
ISBN 978-7-111-26208-4

Ⅰ.J… Ⅱ.任… Ⅲ.JAVA 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2009）第 014653 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
责任编辑：陈 皓
责任印制：李 妍

北京蓝海印刷有限公司印刷

2009 年 3 月第 1 版·第 1 次印刷
184mm×260mm ·25.75 印张·638 千字
0001— 3000 册
标准书号：ISBN 978-7-111-26208-4
定价：42.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：(010) 68326294 68993821
购书热线电话：(010) 88379639 88379641 88379643
编辑热线电话：(010) 88379753 88379739
封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21 世纪高等院校计算机教材系列”、“21 世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21 世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术，以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

20世纪90年代初，Java一出现就以它的Applet吸引了众人的眼球，从而在IT界刮起了一阵Java旋风。在随后的年代里，它以惊人的速度发展成为一门计算机领域不可或缺的技术。它以完全面向对象、动态、稳定、多线程，以及跨平台应用等优势成为了实现企业应用、中间件、分布式计算的有力工具。可以毫不夸张地说，现在整个IT业几乎无处不在使用Java。

顺应时代要求，近些年来各高等学校也把Java作为计算机专业的重要课程之一，并投入了大量的人力和物力，在一定程度上满足了我国软件企业的人才需求。但遗憾的是，迄今为止，企业对学校培养的学生仍然不甚满意。究其原因，作者认为主要有以下几个方面。

1．历史原因。在Java出现之前，人们使用的开发工具大多是VB、VC、Delphi等。由于提供这些工具厂商的实力所在，这些开发工具都制作得相当精致，它们都能为用户自动生成一个完整、健壮的程序框架，软件开发者只需在这个框架中填写自己的业务代码，而不必做更多的工作。无疑，这种体贴入微的开发工具对于企业开发人员来说是件好事，但对于教学来说就不是好事了。它容易导致学校只介绍这些工具的使用方法，而不注意介绍这些工具所生成的程序框架，当然更不会去注意这些程序框架的设计思想和方法的介绍，其结果使学生得不到系统架构设计的训练，进而形成了现在这种尴尬的局面：学生学了一些基本语言，然后就使用这些基本语言在工具生成的程序框架中填空。这种现象就像一个建筑专业的学生不会盖房子，只会搞装修一样。于是，就业市场上就出现了企业高薪聘不到系统架构师，而大量计算机专业毕业生又找不到工作的景象。

2．Java本身的原因。作为应用平台，Java一直没能提供一个令人满意的应用程序编程模型，尤其在开发工具方面显得更为突出。目前，尽管Sun公司和一些大的软件供应商已经意识到这个问题，都想用开发工具来降低Java应用开发的复杂性，但和微软公司的.NET平台相比，Java的开发工具仍然显得很逊色。这些开发工具自动生成的代码像这些工具本身一样复杂。因而，人们很难把这些复杂工具组织到教学内容中，使得高校的Java教学与工程实际应用相差甚远，导致毕业生到了企业之后必须要通过较长时间的培训，否则就不能胜任工作。

3．Java教育思想的原因。要知道，Java不仅仅是一门语言，它更是一门博大精深的技术，是当今先进编程思想和技术的集大成者，而语言只是其技术思想的外在表象。可能由于某种思维惯性，相当一部分学校只是把Java当做一门新的语言来讲授，并没有借用Java平台对面向对象程序设计进行更深层次的介绍，没有使学生在面向对象程序设计方法及思想方面得到真正的训练，从而使学生参加工作后出现了大量很怪异的现象：使用的是面向对象语言Java，而编写的却是面向过程的程序，最终使其开发的软件产品在可维护性、健壮性等方面满足不了企业的要求。

4．Java教学内容的原因。20世纪80年代以来，伴随着面向对象程序设计思想的日益成熟，一些行之有效的软件工程方法也已蓬勃发展起来。但是很遗憾，这些成果并没有以合适的方式体现在高等学校的教学内容中，尽管也开设了软件工程课程，但因太偏重于理论，从而导致其内容过于空泛，况且与其他课程的配合也较差。特别是在面向对象程序设计中行之

有效的设计模式还未以合适的方式引入教学内容。正是由于这方面的缺憾，导致了毕业生空有一身理论而在工程实际问题面前却一筹莫展（从市场上大量缺乏软件架构师的现状就看出了这一点，而培养这种人才正应该是本科教育的责任）。作者认为，对于以培养应用型人才为目标的普通高等学校来说，人们在软件工程实践中总结出来的一些行之有效的设计模式往往比理论更重要。

仔细分析后，作者认为 Java 的缺憾（没能提供一个令人满意的应用程序编程模型）为计算机教育弥补自己的缺憾提供了机会。因为 Java 程序的设计者需要自己来搭建程序框架，这就促使教师必须要向学生介绍程序框架的相应知识，从而也督促教师必须从新的视角对当前的课程体系和教学内容进行审视和改革，以期进一步提升计算机教育水平，使教学更贴近生产实际。

作者通过几年的调查和研究发现，有相当一部分学校以并列的方式开设了很多门程序设计语言课，似乎程序语言学得越多，学生的软件开发能力就会越强，显然这个做法值得怀疑。作者认为，计算机语言只是一种人们用来向计算机描述问题和表述解决问题的方法的工具，学生在校期间主要学习 C/C++和 Java 等几种具有代表性的程序语言（当然，像 XML、SQL 等专用语言例外）即可，即使是讲程序语言也不应该就语言讲语言，而应该通过程序语言课程使学生掌握解决问题的思想和方法。

目前各高校基本都首先介绍 C/C++，这无疑是正确的，但作者不赞成把 C 和 C++分为两门课程，因为 C++是 C 的超集，是 C 的发展，并不是另一门语言。把 C 和 C++分开来介绍就会使学生造成一种致命的误解：用 C 编的程序就是面向过程的程序，用 C++编的程序就是面向对象的程序。其实不然，用 C 也可以编写面向对象的程序，用 C++也可以编写面向过程的程序（现在有很多人声称用 C++编写了面向对象的程序，其实把他的程序仔细看一下就会发现，他编写的还是面向过程的程序），因为面向过程和面向对象只是两种指导程序设计的思想和方法，与使用什么语言并没有直接的联系（当然，C++、C#、Java 为面向对象编程提供了更好的支持）。

在 C/C++课程的后面最好开设微软的 MFC 技术（但不要单纯介绍 VC 工具），因为微软公司在 MFC 中比较好地体现了面向对象的设计思想和方法，如它的文档/视图结构。也就是说，可以把 MFC 作为 C/C++课程的一个大实例来看待，从而使学生真正理解 C++的用途。另外还可以顺便介绍那些 GUI 控件及其事件处理机制，可以说是一举两得。

当学生有了 C/C++和 MFC 的基础之后，学校就可以采用与 C++对比的方式快速进行 Java 语言部分的介绍，以空出学时来介绍 Java 组件和一些在软件工程实践中行之有效的设计模式（如各种工厂模式、模板模式、策略模式和 MVC 模式等），从而使学生初步掌握面向抽象编程的思想和技术，建立程序框架的初步概念，也使课程进一步贴近工程实际。因此，Java 课程应该是程序设计课程的总结课程。

基于上述思想，作者编写了本书。本书以介绍面向对象程序设计思想和方法为目标，以设计模式为线索，重点介绍了面向抽象编程技术。主要内容包括 Java 语言、虚拟机、设计模式应用、组件和事件处理机制。

在 Java 语言部分重点介绍了接口和抽象类，在程序结构部分重点介绍了 Java 常用的设计模式，在 GUI 和组件的内容中重点介绍了事件和事件处理机制，而把虚拟机的介绍则分散在部分章节中，用以支持 Java 动态性的介绍。

在编写过程中，为使本书更符合培养目标和更切合教学实际需要，作者多次请学生和教师阅读了本书的初稿，他们（特别是一些学生）在提出修改意见的同时也提出了一些问题，由于这些问题在一定程度上具有普遍性，所以作者在此一并进行回答。

1．都说 Java 很简单，是这样吗？

应该这样说：与 C++相比，Java 的基本语言确实比较简单，初学者上手很快。之所以如此，是因为这两种语言的定位不同，Java 定位于高端软件开发，而 C++则要兼容 C 并要兼顾底层系统软件的开发，所以它必须保留一些指针、虚函数、运算符重载等初级功能。除此之外，鉴于 C++发展当时面向对象思想的不成熟，因此不可避免地有一些缺憾，例如它的不符合分类学原理的多继承，接口概念不清晰等。而 Java 则利用其后发优势，把那些令程序员头痛的底层初级功能都隐藏了起来。例如，程序员无须再使用容易出问题的指针，也无须显式地销毁对象等。另外，Java 还在概念上进一步清晰了接口和抽象类，提供了反射机制等。

但要注意，Java 的整体技术并不简单。例如，为了实现代码的重用，Java 在其自身系统中大量地使用了设计模式；为了提升动态性，Java 还提供了类的反射机制。因此，全面地理解和掌握 Java 思想并开发出质量较高的 Java 程序还是一件较难的事情。

2．为什么本书对 GUI 组件介绍得如此之少？

桌面系统并不是 Java 的优势，加之在先修课中学生已经学习了一定数量的 GUI 组件，因此本书安排 GUI 这章的主要目的是介绍 Java 的事件处理机制和一个常用的设计模式——观察者模式。

3．为什么在线程那章里介绍了那么多操作系统的内容？是否有些违背本书的宗旨？

确实有这个问题，这也是作者曾经犹豫过的问题。但作者思虑再三，还是这样写了。因为据作者的教学经验，虽然学生学过了操作系统课程，但普遍对进程和线程的知识掌握得不好，而多线程程序的设计又是现代应用程序设计的重要技术基础，况且 Java 的线程还引入了其他系统不多见的管程。所以作者认为，为了使学生能在一个真实的系统中再复习一下操作系统的内容还是值得的。当然，在实际教学活动中如何来处理这部分内容可以由教师灵活掌握。例如，让学生自学，或者让他们结合操作系统课撰写小论文。

4．为什么这么厚的一本书还命名为应用基础？

前面谈到了，桌面系统并不是 Java 的优势，它的优势在于网络的企业应用，即大家常说的 JavaEE，而本书介绍的 JavaSE 正是 JavaEE 的基础，所以本书也只能命名为 Java 技术应用基础。

5．什么是设计模式？它有什么用途？

毋庸置疑，对于任何工程设计来说，成熟的经验是至关重要的，特别是对于计算机这种强非线性系统。好的经验给我们以指导，并节约我们的时间；坏的经验则给我们以借鉴，可以减少失败的风险。然而，从知识层面上来讲，经验只是作为一种工作的积累而存在于个人的大脑中，很难被传授或者记录。为了解决这样的问题，人们提出了模式的概念。所谓模式，是指在一个特定背景下，反复出现的问题解决方案。通常在提到软件设计模式的时候，一般指的是在 GOF 的经典图书《Design Pattern——Elements of Reusable Object-Oriented Software》中出现的 23 个模式，它们是软件设计过程中反复出现的一些问题的解决方案。Java 发展之时，

正是模式兴盛之际，因此 Java 在其 API 中就大量地使用了设计模式，从而使得任何学习 Java 的人不在一定程度上了解模式，就不会真正地理解和掌握 Java API，也就谈不到用这些 API 来编写程序了。从这个角度来说，学习模式是学习 Java 和学习面向对象编程（不仅局限于 Java）的必经之路。

6．有人说学习 Java 要看源代码，是这样吗？

这是没有任何疑问的。只要系统的源码开放，那么它的源码就是最好的教材和老师。要知道，这些源码都是顶级高手编写的，不读不是太浪费了吗？为此，本书也在引导读者阅读源码上给了足够的注意。

7．学习了这本书，将来就能做软件系统架构师吗？

不能。因为做一个软件系统架构师需要多方面的基础和知识，绝不是学一本书就能解决问题的，但这本书确有一部分内容是以培养架构师为目标来编写的。

8．关于程序设计整体教学安排的建议。

从全局来看，普通本科学校程序设计课程开设的顺序似乎应为：

C/C++→MFC→Java（或者 C#）→软件构件

以 C/C++奠定程序设计基础，为学习操作系统、数据结构、编译原理做准备；以 MFC 了解面向对象程序设计思想和方法并了解事件驱动程序设计的特点，兼顾 GUI 及其组件（控件）；在学生学习了操作系统、数据结构、编译原理之后，以 Java（或者 C#）作为程序设计总结课，使学生掌握当代程序设计技术的综合应用（特别是设计模式），进一步使教学内容靠近实际应用。

9．关于 Java 教学的建议。

由于 Java 课程的信息量巨大，不可能完全依靠教师的课堂教学来解决问题，好在面对的都是高年级学生，他们在先修课的学习中都已形成了一定的自学能力（自学能力的培养也是高等教育的责任）。所以作者建议，在 Java 的教学中要在教师的指导之下，充分发挥学生的自学能力。鉴于此，建议在教学中采取如下策略：

- 语言部分。教师重点介绍接口、抽象类的概念和应用，其余部分由学生自学。
- 程序结构和模式部分。教师重点介绍策略模式、简单工厂模式、单例模式和桥梁模式，并通过这些设计模式的介绍，让学生重点领会面向抽象编程和使用关联扩充功能的思想和方法。这个部分可以说是整个 Java 教学的突破点，处理是否得当将直接影响后续教学效果。
- 虚拟机对 Java 特性的支持。由于这些内容都散落在相应的章节中，因此应选择适当的时机结合先修课（可能有些课是并行开设的，如操作系统、编译原理）进行一次梳理和总结。
- Java Beans 部分。这部分是学生将来学习 EJB 的重要基础，也是深化理解 GUI 组件的基础，所以需要慎重处理。

作者认为，处理好上述几个部分后，其他部分就会好办得多。再就是一定要引导学生阅读 Java API 源码并鼓励学生进行讨论和撰写小论文，这会大幅度提高学生分析问题和解决问题的能力。

参加本书编写的有任哲、房红征、李益民、赵谢秋，全书由任哲负责统稿。

作者在编写本书的过程中参考了大量相关文献和网上资料，并引用了其中一些例题、文

字和插图。在此，对这些文献和资料作者的辛勤劳动表示诚挚的谢意。同时也向对本书内容的编排和文字提出了很好意见和建议的老师及同学表示感谢。

在一本 60 万字左右的书中既要介绍 Java 语言，又要介绍虚拟机和设计模式，对水平有限的作者来说确实是一件难事，所以恳请读者对书中的缺点和错误提出批评和指正。

作　者

目　录

出版说明
前言
第 1 章　概述 …… 1
1.1　计算机程序的平台相关性 …… 1
1.1.1　计算机语言与计算机程序 …… 1
1.1.2　应用程序的平台相关性与 Java …… 3
1.2　Java 的故事 …… 4
1.2.1　不成功的 Oak …… 4
1.2.2　适逢其时的 Java …… 6
1.3　体验 Java …… 7
1.3.1　安装 Java 软件开发工具 SDK …… 7
1.3.2　Java 初体验 …… 10
1.3.3　Java 再体验 …… 13
1.3.4　体验 Applet …… 14
1.4　Java 运行环境 …… 17
1.4.1　Java 平台结构及功能 …… 17
1.4.2　Java 平台的 3 种实现 …… 18
1.4.3　Java 虚拟机 …… 18
1.4.4　java.exe 的作用——运行环境的引导与加载 …… 19
1.5　习题 …… 19
第 2 章　Java 语言 …… 20
2.1　Java 语言基础 …… 20
2.1.1　基本数据类型 …… 20
2.1.2　运算符 …… 20
2.1.3　表达式 …… 21
2.2　类与对象 …… 22
2.2.1　类及其对象 …… 22
2.2.2　对象引用 …… 23
2.2.3　构造方法 …… 26
2.2.4　方法重载 …… 27
2.2.5　类成员 …… 28
2.2.6　类的命名及包 …… 29
2.3　类及类成员的访问控制 …… 35
2.3.1　类的访问控制修饰字 public …… 35
2.3.2　字段和方法的访问控制 …… 35

2.4 类的继承（扩展） …… 36
2.4.1 子类的声明 …… 36
2.4.2 子类的构造方法 …… 37
2.4.3 Object 类遗传给子类的常用方法 …… 38
2.5 多态 …… 39
2.5.1 多态的概念及方法重载 …… 39
2.5.2 针对类类型的多态——方法重写 …… 40
2.6 接口 …… 43
2.6.1 接口的概念及其声明 …… 43
2.6.2 接口的实现 …… 44
2.6.3 接口的主要作用 …… 45
2.6.4 接口的扩展 …… 50
2.7 抽象类 …… 52
2.7.1 抽象类的概念及其声明 …… 52
2.7.2 抽象类的用途 …… 52
2.8 内部类和匿名类简介 …… 56
2.8.1 非静态内部类 …… 56
2.8.2 预定义引用 this …… 58
2.8.3 匿名类 …… 59
2.9 Java 提供的预定义类 …… 60
2.9.1 基本类型包装类 …… 61
2.9.2 字符串类及其对象 …… 63
2.9.3 异常类 …… 67
2.10 数组对象 …… 73
2.10.1 一维数组及其定义 …… 73
2.10.2 二维数组 …… 74
2.11 程序流程控制 …… 76
2.11.1 分支控制语句 …… 76
2.11.2 循环控制语句 …… 77
2.11.3 跳转语句 …… 77
2.12 习题 …… 78
第 3 章 虚拟机中类与对象的组织 …… 80
3.1 预备知识 …… 80
3.1.1 字节码指令 …… 80
3.1.2 UTF-8 字符编码 …… 80
3.2 字节码类文件的组织 …… 81
3.2.1 类文件总貌 …… 81
3.2.2 常量池 …… 82
3.3 类文件、Class 对象、类对象 …… 84

3.3.1 类文件与程序之间的关系 …… 84
3.3.2 类文件、类、对象之间的关系 …… 84
3.3.3 类的 Class 对象 …… 86
3.4 Class 的常用方法 …… 87
3.4.1 获取 Class 对象的方法 …… 87
3.4.2 根据类名创建对象 …… 89
3.5 Java 的反射机制 …… 90
3.5.1 Java 反射机制的基本概念及其构成 …… 90
3.5.2 获取类的方法信息 …… 91
3.6 类组织方式对 Java 语言的影响 …… 93
3.6.1 创建 String 对象方法 1 …… 93
3.6.2 创建 String 对象方法 2 …… 96
3.6.3 常量池 String 对象的直接使用 …… 98
3.7 习题 …… 99
第 4 章 Java 程序结构及设计模式 …… 100
4.1 Java 程序设计基础 …… 100
4.1.1 Java 程序结构 …… 100
4.1.2 Java 程序的设计原则 …… 104
4.2 设计模式简介 …… 106
4.2.1 模板方法（Template Method）模式 …… 106
4.2.2 策略（Strategy）模式 …… 109
4.2.3 适配器（Adapter）模式 …… 113
4.2.4 单例（Singleton）模式 …… 116
4.2.5 工厂（Factory）模式 …… 117
4.2.6 桥梁（Bridge）模式及设计模式小结 …… 121
4.3 习题 …… 125
第 5 章 Java I/O 流 …… 126
5.1 流及流类 …… 126
5.1.1 流的概念 …… 126
5.1.2 标准流对象 …… 128
5.2 Java I/O 概貌 …… 129
5.2.1 字节流 …… 130
5.2.2 字符流 …… 131
5.2.3 其他 …… 132
5.2.4 I/O 异常 …… 132
5.3 文件 I/O …… 132
5.3.1 File 类 …… 132
5.3.2 文件输入流 FileInputStream 及其应用 …… 134
5.3.3 FileOutputStream 类及其应用 …… 135

5.3.4 字符流的 FileReader 和 FileWriter 类及其应用 …… 137
5.3.5 RandomAccessFile 类及其应用 …… 139
5.3.6 对象流 ObjectInputStream 和 ObjectOutputStream 的应用 …… 140
5.4 字节流采用的设计模式 …… 142
5.4.1 原始流及其设计模式 …… 142
5.4.2 过滤流及其设计模式 …… 144
5.5 原始流与过滤流的配合应用 …… 148
5.5.1 过滤流 DataInputStream 和 DataOutputStream …… 148
5.5.2 过滤流 BufferedInputStream 和 BufferedOutputStream …… 151
5.5.3 过滤流 PushbackInputStream …… 152
5.5.4 原始流 SequenceInputStream 的应用 …… 154
5.5.5 过滤流 PrintStream 的应用 …… 156
5.6 字符流的设计模式及应用 …… 157
5.6.1 字符流/字节流适配器 …… 158
5.6.2 缓冲器流 BufferedReader 和 BufferedWriter …… 160
5.7 习题 …… 161
第 6 章 Java GUI 简介 …… 163
6.1 Java GUI 概述 …… 163
6.1.1 AWT …… 163
6.1.2 Swing …… 166
6.2 Java GUI 基本概念 …… 167
6.2.1 简单 AWT GUI 程序实例 …… 167
6.2.2 Java 组件和容器 …… 168
6.2.3 布局管理器 …… 170
6.2.4 使用容器嵌套实现复杂布局 …… 174
6.3 AWT 事件及其处理 …… 175
6.3.1 事件的基本概念 …… 176
6.3.2 Java 的事件处理机制 …… 176
6.3.3 事件分类 …… 182
6.3.4 监听器 …… 183
6.3.5 Java 事件处理机制中的设计模式 …… 184
6.4 AWT GUI 程序综合实例 …… 188
6.4.1 简单计算器 …… 188
6.4.2 简单记事本 …… 192
6.5 Swing GUI 简介 …… 196
6.5.1 简单 Swing GUI 应用程序实例 …… 196
6.5.2 Swing 组件 …… 197
6.6 习题 …… 199
第 7 章 Java 与图形图像 …… 203

7.1 文本的绘制 …… 203
7.1.1 显示器屏幕的坐标系 …… 203
7.1.2 字体 …… 203
7.1.3 颜色控制 …… 206
7.2 图形 …… 207
7.2.1 基本几何图形的绘制 …… 207
7.2.2 Java2D 图形 …… 212
7.3 图像 …… 218
7.3.1 Image 类 …… 218
7.3.2 在应用程序中绘制图像 …… 218
7.4 习题 …… 221
第 8 章 Java 集合框架和泛型设计简介 …… 222
8.1 集合框架介绍 …… 222
8.1.1 什么是集合框架 …… 222
8.1.2 接口 Collection、List、Set 和 Map …… 223
8.2 集合框架的主要实现类和泛型初步 …… 224
8.2.1 列表 ArrayList …… 224
8.2.2 列表 LinkedList …… 229
8.2.3 集合 HashSet 和 LinkedHashSet …… 230
8.2.4 排序接口和集合 TreeSet …… 232
8.2.5 映射集 Map …… 240
8.2.6 集合的迭代器 Iterator …… 241
8.3 泛型设计 …… 243
8.3.1 再谈泛型的概念 …… 243
8.3.2 简单泛型程序设计 …… 244
8.4 习题 …… 246
第 9 章 类装载器与 Java 动态性 …… 248
9.1 类文件就是动态链接库 …… 248
9.2 类装载器简介 …… 253
9.2.1 类装载器的功能 …… 253
9.2.2 类装载器的组织 …… 257
9.3 显式动态程序设计 …… 259
9.3.1 Class 类的 forName()方法 …… 260
9.3.2 ClassLoader 类的 loadClass()方法 …… 265
9.4 习题 …… 266
第 10 章 Java 线程技术基础 …… 267
10.1 线程的概念 …… 267
10.2 Java 线程 …… 268
10.2.1 虚拟机对线程的支持 …… 268

10.2.2 Java 线程的状态及其转换 …… 270
10.2.3 线程调度及线程优先级 …… 271
10.2.4 线程的创建及线程类 Thread …… 273
10.2.5 线程的基本控制 …… 279
10.3 线程的互斥与同步 …… 283
10.3.1 问题的起源 …… 283
10.3.2 互斥 …… 285
10.3.3 Java 互斥的实现方法 …… 288
10.3.4 同步 …… 292
10.4 习题 …… 296
第 11 章 Java 组件 …… 297
11.1 概述 …… 297
11.1.1 软件组件 …… 297
11.1.2 JavaBeans 简介 …… 298
11.2 JavaBeans 的简单属性 …… 299
11.2.1 属性 …… 299
11.2.2 单值属性 …… 301
11.2.3 索引属性 …… 302
11.3 关联属性和限制属性 …… 303
11.3.1 属性变化事件 …… 303
11.3.2 关联属性 …… 308
11.3.3 限制属性 …… 310
11.3.4 小结 …… 315
11.4 习题 …… 315
第 12 章 Java 小程序——Applet …… 316
12.1 Java Applet …… 316
12.1.1 小程序的设计思想及 Applet 类 …… 316
12.1.2 小程序的安全性 …… 320
12.1.3 小程序的生命期及 HTML 文件 …… 320
12.2 小程序和浏览器的通信 …… 325
12.2.1 HTML 文件格式 …… 325
12.2.2 通信方法 …… 328
12.3 小程序与 Swing …… 330
12.4 习题 …… 335
第 13 章 Java 网络应用基础 …… 336
13.1 传输控制协议简介 …… 336
13.1.1 TCP 协议 …… 336
13.1.2 TCP 协议规范 …… 338
13.2 Java 的 InetAddress 类 …… 339

13.3 Sockets 应用 …… 343
13.3.1 Socket 类及其对象的创建 …… 343
13.3.2 ServerSocket 类及其对象的创建 …… 345
13.3.3 应用示例 …… 346
13.3.4 在 Applet 中使用 Sockets 连接 …… 349
13.4 用户数据报协议及其应用 …… 353
13.4.1 UDP 协议简介 …… 354
13.4.2 UDP 协议应用实例 …… 358
13.5 统一资源定位符的应用 …… 368
13.5.1 统一资源定位符 …… 369
13.5.2 使用 URL 类对象对远程文件进行操作 …… 370
13.6 习题 …… 371
第 14 章 Java 数据库应用基础 …… 372
14.1 数据库系统概述 …… 372
14.2 JDBC 概述 …… 373
14.2.1 JDBC 的类和接口 …… 373
14.2.2 用 JDBC 访问数据库的一般步骤 …… 377
14.2.3 JDBC 的实现方式 …… 378
14.3 JDBC 的应用 …… 380
14.3.1 一个使用 JDBC/ODBC 桥的应用实例 …… 380
14.3.2 一些细节 …… 385
14.4 习题 …… 394
参考文献 …… 395

第1章 概 述

20世纪90年代初出现的Java技术，全面综合地应用了当时所有先进程序设计理念及技术，以它完全面向对象、动态、稳定、多线程及跨平台应用的优势成为了实现企业应用、中间件、分布式计算的有力工具。目前，它已发展成为一门计算机领域不可或缺的技术。

本章主要内容：

- 计算机语言和计算机程序的基本概念
- Java的产生及发展背景
- Java程序初体验

1.1 计算机程序的平台相关性

1.1.1 计算机语言与计算机程序

众所周知，人类的语言是人们在工作和生活中进行信息传递和信息交流的工具。在日常劳动和生活中，当人们向其他人表述某项工作的做法时，通常将工作分成若干个步骤，并用对方能看（听）得懂的文字（语言）表达出来。例如，某电话机的故障排除说明书如表1-1所示。

表1-1 电话机故障排除操作说明

故障现象	故障排除
无声	把插头与接线盒连接好
只能接听不能拨出	检查拨号制式是否正确
有杂音	更换接线或接线盒

如果要把上述表格用文字来表达，则为：

1）如果无声，则把插头与接线盒连接好。

2）如果只能接听不能拨出，则检查拨号制式是否正确。

3）如果有杂音，则更换接线或接线盒。

其实，上述这段文字就是一个程序，是一个用人类自然语言编写的程序，其功能是指导人们排除电话机的故障。

显然，如果希望由计算机来完成上述任务，就必须有人与计算机进行交流的语言，并用这种语言来编写程序。为此，人们发明了多种用来与计算机进行交流的语言，但由于这些语言与人类语言还存在着比较大的区别，因此这些语言都被称为计算机语言或计算机程序设计语言。

根据计算机硬件与软件的不同配置，计算机所能接受的语言也不同，以人类语言为标准，计算机语言大体上可分为机器语言、低级语言和高级语言3大类。在语法和词法上与人类语言越接近的越高级，否则就越低级。机器语言是最低级的计算机语言。