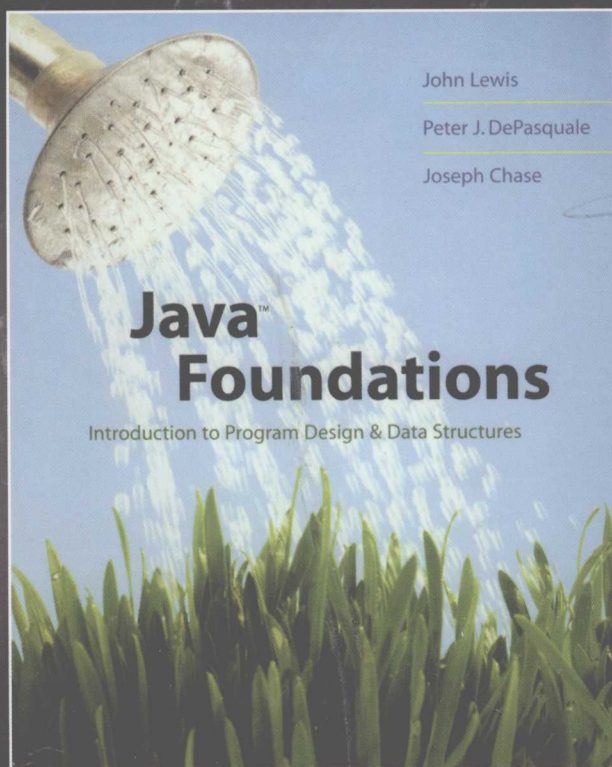


计 算 机 科 学 丛 书

Java程序设计 与数据结构导论

(美) John Lewis Peter J. DePasquale Joseph Chase 著 辛运伟 饶一梅 译
雷德福大学 新泽西学院 雷德福大学 南开大学



Java Foundations
Introduction to Program Design
and Data Structures



机械工业出版社
China Machine Press

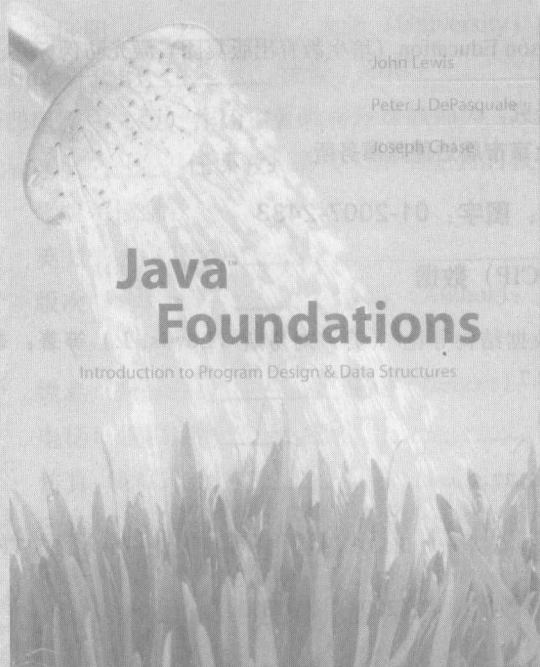
计 算 机 科

TP312/2935D

2008

Java程序设计 与数据结构导论

(美) John Lewis Peter J. DePasquale Joseph Chase 著 辛运伟 饶一梅 译
雷德福大学 新泽西学院 雷德福大学 南开大学



Java Foundations
Introduction to Program Design and Data Structures



机械工业出版社
China Machine Press

本书作者颇具匠心地将程序设计、数据结构等相关课程内容有机地包容在一本书中，解决了这两门专业基础课不能很好衔接的问题。本书从一开始就将重点放在软件开发实践方面，在解释如何编写类之前先介绍了控制结构，从而使读者在真正了解了基础程序设计语句之后再学习编写类代码。作者还使用数据结构中的内容作为问题导引，在介绍经典算法的同时，全面介绍了如何用所学知识解决实际问题。

本书内容全面，讲解透彻，可作为程序设计、数据结构等相关课程的教材。

Simplified Chinese edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Java Foundations: Introduction to Program Design & Data Structures* (ISBN 978-0-321-42972-8) by John Lewis, Peter J. DePasquale, Joseph Chase. Copyright © 2008.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2007-2433

图书在版编目（CIP）数据

Java程序设计与数据结构导论/（美）刘易斯（Lewis, J.）等著；辛运韩等译. —北京：机械工业出版社，2008.7

（计算机科学丛书）

ISBN 978-7-111-24177-5

I. J… II. ①刘… ②辛… III. JAVA语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字（2008）第073931号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：刘立卿

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2008年7月第1版第1次印刷

184mm × 260mm · 26.75印张

标准书号：ISBN 978-7-111-24177-5

ISBN 978-7-89482-677-0（光盘）

定价：56.00元（附光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作，而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章教育

华章科技图书出版中心

译者序

我们非常高兴这次有机会翻译这本教材。这本书的内容非常丰富，篇幅也较多。作者在前言里也强调，如此多的内容，适合作为两到三个学期的教材使用。这样的内容安排，是作者深思熟虑之后的结果。实际上，高等学校计算机专业的课程体系中，很多课程是紧密联系的，但在讲授中，各位教师往往不能兼顾自己的讲授内容与其他课程的关系。这是由现行的教学体制所决定的。限于各种情况，学生们也不能主动地去思考本课与其他课程的衔接关系。本书在这方面做出了很好的尝试。作者颇具匠心地将两门甚至三门相关课程的内容包容在一本书中，各部分的内容既独立又相关。教师在讲授时可以兼顾前后内容的呼应，学生们也能领会知识点之间的关系，真正做到知识的融会贯通。

具体来说，程序设计语言课程被安排为各高校计算机专业大学一年级新生的入门课程，数据结构多安排在二年级。这两门课程往往由不同的教师担任，这在一定程度上割裂了两者的联系。作为它们之间的重要关联内容——面向对象的设计思路和方法，既没能在一年级学习语言的时候深入理解，也不能等到二年级学习数据结构课程时再去介绍。一年级的学生把学习重点放在了语言的语法上，此时他们缺乏编制较大例题的能力，教师也没有机会借助有具体背景的实际例子向他们介绍面向对象的编程技巧。这样的现状导致学生们在二年级学习数据结构时不知道如何运用面向对象的思想去解决相关的实际问题。本书作者对这样的难题给出了令人满意的解答。作者先详细介绍了Java语言的语法，在学生们掌握了基本编程能力之后，重点介绍类及对象的知识，这些内容不局限于Java语言，而是一般的面向对象编程语言都具备的。这些内容的介绍是充分的，对学生们来说是非常必要的，也正是我们目前在程序设计课程所欠缺的。同时，作者使用Java语言作为实现面向对象思想的工具，使学生们自然而然地提高了驾驭Java语言的能力。之后，作者使用一个实际的例子，全面详细地向学生介绍如何编写、调试一个程序。这个环节是体现本书实践性的一个典型代表。作者还使用数据结构中的内容作为问题导引，在向学生介绍经典算法的同时，也全面介绍了如何用前面学到的技能来解决实际问题。

本书不同于国内很多教材的另一个特色是，它是一本资源性的教材。本书是关于Java语言的，在介绍语言本身的同时，也全面介绍了学习这个语言重要的相关资料，包括下载资料的网站、调试工具的使用、编写程序用到的测试知识等。特别是书后的附录内容，相信对教师和学生都非常有用。

本书还提供了大量的代码，这些代码对学生们理解课程内容非常有益。每章后面列出的各类习题可以检验学生的学习效果。

在此，译者非常感谢机械工业出版社华章分社给我们提供了这个翻译机会。我们不仅学习了作者的编写思想，更见识了作者的敬业精神。书中反映出的作者的认真态度，使我们在翻译过程中不敢有丝毫的懈怠。也非常感谢各位编辑们。正是他们的认真工作，才让本书顺利地 and 读者见面。

前言

欢迎阅读本书。本书可作为导论性课程的教材，教学内容从最基本的程序设计概念到复杂数据结构的设计和实现，这种统一的教学方法能让这些重要的内容自成体系，并且更易让学生理解。

我们借鉴了业界领先的教材《Java Software Solutions》中的最精华内容作为介绍资料，重新组织后作为本书大纲的有效补充。例如，我们没有将图形用户界面的内容分散在好几章中，而是将这些内容很好地组织在一章内。

在后面的几章中，有关集合和数据结构的介绍基本上参照了《Java Software Structures》一书，但重新组织了相关资料，使得条理更清晰。这样，程序设计概念的介绍范围更全面、更集中，相互的衔接也更自然。

关于对象

“对象”概念的引入时机，一直是由计算机教师自由掌握的，事实上导论性课程的教学不可能是相关知识的简单叠加。这里我们有机会说明我们的方法。

首先，本书是**纯面向对象的**，所以采用了循序渐进的自然方式来介绍。在过程式程序设计方法中也存在的一些概念，如方法和它们的调用，我们使用的是面向对象的术语来讨论的。所以，没有由多个方法组成单个类的例子。事实上，在我们的例子中，包含main方法的类不再包含其他的方法。

我们从一开始就使用**对象**这个术语，并且始终使用面向对象的术语来讨论所有的内容。在第1章中给出了面向对象概念的概述，并在本书中始终强化并充实这些概念。之后立即介绍了Java标准类库中的类，实例化了这些类中的对象，并使用了它们提供的各种服务。在前4章中，学生们学习编写的程序只由单个的类组成，并且其中只含有单一的main方法，不过这些程序在介绍如表达式和条件等这些程序设计的基本概念之外，都有效地使用了标准库中预定义的类和对象。

在编写例子时**完全没有使用第三方的类**。那些方法的使用方式与标准类库中（总是可用）的类是有区别的，所以容易给学生带来疑惑，因此本书的作者没有选用这些内容。凡是例子中使用的非类库中的每个类都在本书中给出了全面介绍，不留下任何不可思议的隐晦的地方。

要解决的问题仍然是：是在详细介绍如何编写类之前介绍控制结构的内容，还是把次序倒过来。事实是，哪种方法都有好处，教师可以使用任一种方式。如果先介绍类，就会先揭示对象的本质及它们的使用方法。但是，缺乏基本的控制结构的驾驭能力，此时的例子常常是无趣且没有实用意义的。本书**先介绍控制结构，然后再介绍类的概念**。第4章使用小的、单个的方法示例来介绍条件和循环的细节，为第5章多个类的示例打下坚实基础。

各章安排

第1章（绪论）介绍了Java程序设计语言和基本的程序开发过程。介绍了面向对象的开发方法，包括相关的概念和术语。这一章还介绍了帮助学生熟悉开发环境所适用的主要参考

资料。

第2章（数据和表达式）探讨了Java程序中使用的数据基本类型及执行计算时表达式的使用。讨论了数据类型之间的转换，及如何借助于Scanner类交互地从用户读入输入。

第3章（类及对象的使用）探讨了预定义的类及由它们而创建的对象的使用。类和对象用来操作字符串、产生随机数、执行复杂的计算，及格式化的输出。讨论了包、枚举类型和包装类。

第4章（条件和循环）介绍了用于判定的布尔表达式的使用。讨论了与条件及循环相关的所有语句，包括for循环的增强版本。为了分析并读入文本文件中的重复输入，再次讨论了Scanner类。

第5章（编写类）探讨了与编写类及方法相关的基本问题。内容包括实例数据、可见性、作用域、方法参数及返回值类型。也涉及了构造方法、方法设计、静态数据和方法重载等内容。

第6章（图形用户界面）全面研究了Java GUI处理，重点是组件、事件和监听器。使用许多GUI例子讨论了不同类的组件和事件。另外，也介绍了布局管理器、包含层次、边界、工具提示和助记符等内容。

第7章（数组）涉及数组和数组的处理。内容包括边界检查、初值表、命令行参数、变长参数表和多维数组等。

第8章（继承）讨论了类的派生及相关的概念，包括类层次、重写及可见性等。重点是强调在软件设计中继承的正确使用及它的作用是什么。

第9章（多态）探讨了绑定的概念，及它与多态的关系。之后研究了如何使用继承或接口完成多态引用。与多态相关的设计问题也在这一章内介绍。

第10章（异常）介绍了异常处理及不捕获异常时的后果。探讨了try-catch语句，讨论了异常的传播。本章还介绍了处理输入输出时的异常使用，给出了一个写文本文件的例子。

第11章（创建、测试和调试）详细介绍了Java包机制，介绍了辅助开发中、大规模程序的创建文件的使用。另外，本章讨论了通过JUnit进行测试、测试用例开发和单元测试的概念。也介绍了程序调试方面的内容。本章的例子创建了一个定制的邮件客户端，学生们可以对此有所扩展。

第12章（递归）介绍了递归的概念、递归的实现和其正确的用法。讨论中精心选择了几个示例，包括迷宫遍历和经典的汉诺塔问题。

第13章（查找与排序）探讨了线性查找和二分查找算法，还介绍了5个排序算法。本章还讨论了包括递归算法在内的算法复杂度分析技术。

第14章（集合和链表）讨论了集合的用法，确定了接口与实现分离的重要性。本章介绍了泛型，详细介绍了它们在集合类中的作用。也介绍了动态实现、静态实现和动态链表的知识。

第15章（栈和队列）介绍了两个经典的集合，讨论了它们的实现。在两个例子中，都是先从概念上介绍集合，然后把它们用作解决问题的工具，最后探讨数据结构。最后讨论了基于数组的实现和基于动态链表的实现。

第16章（树）介绍了树的术语和相关概念。讨论了不同的实现策略，详细介绍了递归的链式的实现方法。还给出了一个二叉决策树的示例。

第17章（查找树和堆）介绍了二叉查找树及其链式实现。也讨论了树的旋转算法。本章还探讨了堆和它们的实现。

第18章(图)讨论了有向图和无向图。此外,还介绍了带权图、广度优先遍历和深度优先遍历之间的差别。介绍了最小生成树,讨论了它的实现策略。

学生CD

随书附带的CD包括以下内容:

- 书中所有程序的源代码。
- Java软件开发工具包(SDK)。
- 各种Java开发环境,包括NetBeans™、Eclipse™、Dr.Java、jGRASP™、JCreator®和TextPad™。

教师资源

下列补充材料只为老师提供^①。可以咨询如何得到下列这些资源。

- 各章的PowerPoint格式的幻灯片。
- 练习及程序设计项目的全部答案。
- 功能强大的测验生成软件Test Bank——包括大量的自由回答题、多选题及判断题。
- 配合书中讲授内容的相关实验资料。

致谢

感谢世界各地的教师和学生给我们反馈意见,以让本书更加完善。欢迎您的任何意见和建议。

Addison-Wesley公司的工作团队的才干和付出总是让我们很吃惊。我们的编辑Michael Hirsch先生,总能把握出版的方向,我们非常感谢他的合作方式。他的助手Lindsey Triebel一直在帮助我们。市场部经理Michelle Brown让教师们了解本书的优点。Joyce Wells负责封面及内部的设计。Marilyn Lloyd负责出版。Addison-Wesley又受由Megan Schwenke领导的出色的Argosy Publishing所支持。我们的文稿编辑Bill McManus先生的慧眼和技术背景也让我们很吃惊。我们感谢为确保本书以最高标准面世而做出努力的所有人员。

我们想要感谢世界各地数百位教授和学生为本书提供的原始资料。这些人太多了,恕我们不能一一列出他们的姓名,但你们为《Java Software Solutions》和《Java Software Structures》所提的所有建议,都体现在本书中了。

特别要感谢新泽西大学的Jason Snyder先生的测试代码和很多其他的帮助。Jason就要从事计算机教育方面的工作了,他未来的同事与他共事是非常幸运的,不过玩牌时可不要和他打对手。

一些团体,如ACM Special Interest Group on Computer Science Education (SIGCSE)、Consortium for Computing Sciences in Colleges (CCSC) 和Computer Science Teachers Association (CSTA),也提供了很出色的资源。他们的会议和网上的活动为远方的各类学校的教师提供了机会,让人们可以共享他们的思路 and 资料。如果你是一名计算机的教师,但还没有参加这些组织的话,那你错失了很多。

最后,感谢我们的家人对我们的支持和耐心。

^① 教师资源的获取请参考书末所附“教学支持说明”。

目 录

出版者的话

译者序

前言

第1章 绪论1

1.1 Java程序设计语言1

1.1.1 Java程序2

1.1.2 注释3

1.1.3 标识符和保留字4

1.1.4 空白5

1.2 程序开发6

1.2.1 程序语言等级7

1.2.2 编辑器、编译程序和解释程序8

1.2.3 开发环境9

1.2.4 语法和语义9

1.2.5 错误10

1.3 问题求解11

1.4 软件开发活动11

1.5 面向对象程序设计12

知识点摘要14

习题15

第2章 数据和表达式18

2.1 字符串18

2.1.1 print及println方法18

2.1.2 字符串连接19

2.1.3 转义字符21

2.2 变量和赋值22

2.2.1 变量22

2.2.2 赋值语句23

2.2.3 常量24

2.3 基本数据类型25

2.3.1 整型和浮点型25

2.3.2 字符类型26

2.3.3 boolean类型27

2.4 表达式27

2.4.1 数学运算符27

2.4.2 运算符优先级28

2.4.3 增量及减量运算符30

2.4.4 赋值运算符31

2.5 数据转换31

2.6 读输入数据33

知识点摘要36

习题36

第3章 类及对象的使用41

3.1 创建对象41

3.2 String类43

3.3 包45

3.4 Random类47

3.5 Math类49

3.6 格式化输出51

3.6.1 NumberFormat类51

3.6.2 DecimalFormat类52

3.6.3 printf方法53

3.7 枚举类型54

3.8 包装类55

知识点摘要56

习题57

第4章 条件和循环61

4.1 布尔表达式61

4.1.1 相等及关系运算符62

4.1.2 逻辑运算符62

4.2 if语句63

4.2.1 if-else语句65

4.2.2 使用块语句66

4.2.3 条件运算符68

4.2.4 嵌套的if语句69

4.3 数据比较70

4.3.1 浮点数比较70

4.3.2 字符比较70

4.3.3 对象比较71

4.4 switch语句72

4.5 while语句74

4.5.1 无限循环77

4.5.2 嵌套的循环	77
4.5.3 其他的循环控制	80
4.6 迭代器	80
4.7 do语句	82
4.8 for语句	84
4.8.1 迭代器和for循环	86
4.8.2 循环语句的比较	87
知识点摘要	87
习题	87
第5章 编写类	93
5.1 再谈类和对象	93
5.1.1 类和对象的确定	94
5.1.2 分配职责	95
5.2 类的剖析	95
5.2.1 实例数据	98
5.2.2 UML类图	99
5.3 封装	100
5.3.1 可见性修饰符	100
5.3.2 访问方法和修改方法	101
5.4 方法的剖析	104
5.4.1 return语句	108
5.4.2 参数	108
5.4.3 局部数据	109
5.4.4 再次讨论构造方法	110
5.5 静态类成员	110
5.5.1 静态变量	110
5.5.2 静态方法	110
5.6 类关系	112
5.6.1 依赖	112
5.6.2 同一个类中各对象之间的依赖	113
5.6.3 聚合	117
5.6.4 this引用	117
5.7 方法设计	118
5.7.1 方法分解	118
5.7.2 再次讨论方法参数	121
5.8 方法重载	124
知识点摘要	126
习题	126

第6章 图形用户界面	132
6.1 GUI元素	132
6.1.1 框架和面板	134
6.1.2 按钮和动作事件	135
6.1.3 判定事件源	136
6.2 更多的组件	138
6.2.1 文本域	138
6.2.2 复选框	140
6.2.3 单选按钮	142
6.2.4 滑动条	145
6.2.5 组合框	148
6.2.6 计时器	151
6.3 布局管理器	153
6.3.1 流布局管理器	155
6.3.2 边界布局管理器	157
6.3.3 网格布局管理器	159
6.3.4 盒子布局管理器	160
6.3.5 包含层次	162
6.4 鼠标和键盘事件	162
6.4.1 鼠标事件	162
6.4.2 键盘事件	167
6.4.3 继承适配器类	170
6.5 对话框	170
6.5.1 文件选择器	172
6.5.2 颜色选择器	173
6.6 一些重要细节	174
6.6.1 边界	174
6.6.2 工具提示和助记符	177
6.7 GUI设计	180
知识点摘要	181
习题	181
第7章 数组	186
7.1 数组元素	186
7.2 数组的声明及使用	187
7.2.1 边界检查	188
7.2.2 两种数组语法	191
7.2.3 初值表	191
7.2.4 数组作为参数	192
7.3 对象数组	193

7.4 命令行参数	199	11.2.1 通过包来组织代码	258
7.5 变长参数表	199	11.2.2 创建自己的包	258
7.6 二维数组	202	11.2.3 使用CLASSPATH	259
知识点摘要	205	11.2.4 Java存档 (Jar) 文件	261
习题	205	11.2.5 使用其他开发人员的包	262
第8章 继承	209	11.3 建立Java工程	263
8.1 创建子类	209	11.3.1 IDE和CLASSPATH	263
8.1.1 protected修饰符	212	11.3.2 创建工具和Ant	263
8.1.2 super引用	213	11.4 测试	265
8.1.3 多重继承	215	11.4.1 审核	266
8.2 方法覆盖	215	11.4.2 缺陷测试	266
8.3 类层次	217	11.4.3 单元测试	267
8.3.1 Object类	218	11.4.4 集成测试	267
8.3.2 抽象类	219	11.4.5 系统测试	267
8.4 可见性	220	11.4.6 JUnit	268
8.5 设计继承	221	11.4.7 JUnit的进一步讨论	271
知识点摘要	222	11.4.8 测试驱动的开发	273
习题	223	11.4.9 关于测试及JFMail的最后注释	273
第9章 多态	226	11.5 调试	274
9.1 后绑定	226	11.5.1 使用print语句进行简单调试	274
9.2 通过继承实现多态	227	11.5.2 调试概念	274
9.3 接口	234	11.5.3 Java调试程序 (JDB)	275
9.3.1 接口层次	237	11.5.4 调试JFMail程序	277
9.3.2 Comparable接口	237	11.5.5 基于GUI的调试程序	279
9.3.3 Iterator接口	238	知识点摘要	279
9.4 通过接口实现多态	238	习题	280
知识点摘要	240	第12章 递归	282
习题	240	12.1 递归思想	282
第10章 异常	243	12.1.1 无穷递归	283
10.1 异常处理	243	12.1.2 数学中的递归	283
10.2 不捕获异常	244	12.2 递归程序设计	283
10.3 try-catch语句	244	12.2.1 递归与迭代	285
10.4 异常传播	247	12.2.2 直接递归与间接递归	285
10.5 异常类层次	249	12.3 使用递归	286
10.6 I/O异常	251	12.3.1 走迷宫	286
知识点摘要	253	12.3.2 汉诺塔	289
习题	253	知识点摘要	292
第11章 创建、测试和调试	256	习题	292
11.1 创建大型应用程序: JFMail	256	第13章 查找与排序	295
11.2 包	258	13.1 查找	295

13.1.1 线性查找	297	15.9 使用循环数组实现队列	354
13.1.2 二分查找	298	15.10 使用链表实现队列	357
13.2 排序	300	15.11 栈与队列实现的分析	359
13.2.1 选择排序	303	知识点摘要	359
13.2.2 插入排序	304	习题	360
13.2.3 起泡排序	305	第16章 树	364
13.2.4 快速排序	305	16.1 树	364
13.2.5 归并排序	306	16.2 树的遍历	365
13.3 算法分析	308	16.2.1 先序遍历	366
13.3.1 增长函数和大 $O()$ 符号	308	16.2.2 中序遍历	366
13.3.2 比较增长函数	309	16.2.3 后序遍历	367
13.3.3 分析循环的执行	310	16.2.4 层序遍历	367
13.3.4 分析递归程序	310	16.3 树的实现策略	368
13.4 分析查找及排序算法	311	16.3.1 在数组中计算链	368
13.4.1 比较查找算法	311	16.3.2 在数组中保存链	368
13.4.2 比较排序算法	312	16.3.3 链式结点	368
知识点摘要	313	16.4 二叉树实现	369
习题	314	16.5 决策树	375
第14章 集合和链表	317	知识点摘要	377
14.1 集合简介	317	习题	378
14.1.1 独立于实现的接口	317	第17章 查找树和堆	381
14.1.2 Java Collections API	318	17.1 二叉查找树	381
14.1.3 泛型	318	17.1.1 在二叉查找树中添加元素	382
14.2 包集合	319	17.1.2 从二叉查找树中删除元素	382
14.3 使用数组实现包	322	17.2 二叉查找树的实现	384
14.4 链表	327	17.3 平衡的二叉查找树	388
14.4.1 链表的管理	328	17.3.1 右旋转	389
14.4.2 没有链的元素	330	17.3.2 左旋转	389
14.5 包的链式实现	330	17.3.3 右-左旋转	389
知识点摘要	333	17.3.4 左-右旋转	390
习题	334	17.4 堆	390
第15章 栈和队列	337	17.4.1 向堆中添加元素	391
15.1 栈	337	17.4.2 从堆中删除最大元素	391
15.2 计算后缀表达式	338	17.5 堆的实现	392
15.3 使用数组实现栈	343	17.6 堆排序	396
15.4 使用链表实现栈	345	知识点摘要	397
15.5 java.util.Stack类	347	习题	397
15.6 队列	347	第18章 图	401
15.7 基数排序	349	18.1 无向图	401
15.8 使用数组实现队列	351	18.2 有向图	402

18.3 带权图	402	18.5 图的实现策略	408
18.4 常用的图算法	403	18.5.1 邻接表	408
18.4.1 遍历	403	18.5.2 邻接矩阵	409
18.4.2 测试连通性	406	知识点摘要	410
18.4.3 最小生成树	406	习题	410
18.4.4 判定最短路径	408	附录 [⊖]	

[⊖] 附录部分请见本书附带光盘。——编辑注

第1章 绪论

本章目标

- 介绍Java程序设计语言。
- 描述程序编译和运行涉及的步骤。
- 探讨与一般问题求解相关的内容。
- 讨论软件开发过程中涉及的活动。
- 概述面向对象原理。

本书将介绍如何编写精心设计的软件。我们将从一个非常简单的Java程序开始，以它为例研究最初的程序设计概念。然后更深入地探讨软件开发的技术基础，研究求解问题的基础、软件开发过程中涉及的活动及面向对象编程的原则。

1.1 Java程序设计语言

计算机由硬件和软件组成。计算机系统的硬件是物理的、有形的部件，能提供计算功能。它们包括芯片、机箱、电线、键盘、喇叭、硬盘、电缆、打印机等。如果没有指令告诉硬件该做什么，实际上它就没什么用处。程序就是硬件逐条执行的一系列指令。程序有时也称为应用程序。软件由程序和程序使用的数据组成。软件是与物理硬件部件相匹配的无形的东西。它们一起组成可用来解决问题的工具。

程序用具体的程序设计语言编写，这些语言使用具体的字和符号来表述问题的解决方案。程序设计语言定义了一组规则，明确地告诉程序员如何将语言的这些字和符号组成程序语句，而这些语句就是程序运行时要执行的指令。

计算机诞生之初，就出现了许多程序设计语言。本书中使用Java语言来解释编程概念和技术。虽然主要的目的是学习这些软件开发概念，与此同时也一并学习了Java程序的开发。

和其他语言相比，Java是比较新的程序设计语言。它在20世纪90年代初由Sun公司的James Gosling开发。Java于1995年正式推出，从那时起就非常流行。

Java从诞生之日起经过多次修改。最新的Java技术一般称为Java 2平台，它由以下三部分组成：

- Java 2平台，标准版 (J2SE)
- Java 2平台，企业版 (J2EE)
- Java 2平台，小型版 (J2ME)

本书主要使用标准版，顾名思义，它是语言及关联工具的主流版本。本书基于标准版的最新版本J2SE 6.0编写而成。

有些早期的Java技术已经不推荐使用了，这也就是说它们已过时，不应该再使用了。当用到时，我们会指出这些过时的元素，并讨论替代的元素。

Java是面向对象的程序设计语言。对象是组成程序的基本元素。面向对象软件开发的原则是本书的基础。本章后面将探讨面向对象程序设计的概念，这些概念一直贯穿于本书中。

Java语言有一个额外的软件库，开发程序时可以使用。称为Java标准类库的这个软件，提供了图形设计、网络上的通信、与数据库交互及其他一些功能。支持Java程序设计的标准类库很大，并且相当全面。虽然不能介绍这个库的所有内容，但本书会涉及其中的许多方面。

Java用在全世界的商业环境中。一直以来，它都是成长最快的程序设计技术之一。所以它不只是用来学习编程概念的一种好语言，未来也能让你具备实际的编程能力。

1.1.1 Java程序

我们来看一个简单但完整的Java程序。程序1.1将在屏幕上输出两个句子。这个程序会打印亚伯拉罕·林肯的一句名言。输出显示在程序的后面。

程序1.1

```
//*****
// Lincoln.java          Java Foundations
//
// Demonstrates the basic structure of a Java application.
//*****

public class Lincoln
{
```

```
    //-----
    // Prints a presidential quote.
    //-----
```

```
    public static void main (String[] args)
```

```
    {
        System.out.println ("A quote by Abraham Lincoln:");
```

```
        System.out.println ("Whatever you are, be a good one.");
    }
```

输出

A quote by Abraham Lincoln:

Whatever you are, be a good one.

所有的Java应用都有类似的基本结构。虽然这个程序很小很简单，但它包含了几个重要特征。我们仔细剖析并逐项来分析一下。

程序最前面的几行是注释，它由//符号开头，一直到行尾结束。注释对程序没有影响，但写注释是为了让人能更容易地理解程序。程序员可以也应该在程序需要的地方写上注释，明确标出程序的目的，描述具体的处理过程。包括用户指南和技术参考在内的任何注释或文档都称为文档。程序中的注释称为内联文档（inline documentation）。

程序的其他部分是类定义。这个类叫Lincoln，当然我们可以用任何的名字来命名它。类定义从第一个开括号（{）开始到程序最后一行的最后一个闭括号（}）结束。所有的Java程序都用类来定义。

类定义中有描述main方法目的的另外一些注释，main方法就定义在注释的下面。方法是带有给定名字的一组程序设计语句。本例中，方法名是main，它只含有两个程序语句。与类定义一样，方法也括在括号内。

所有的Java应用都有一个main方法，它是程序处理开始的地方。main方法中的每个程序语句都要执行，一次一条，按序执行，直到到达方法的末尾。然后程序结束，或中断。Java程序中的main方法前面总有public、static和void几个字，后面将讨论这几个字。本程序中String和args的作用并没有体现。稍后再做讨论。

main方法的两行代码调用了另一个方法println。当想让某个方法执行时就调用这个方法。println方法在屏幕上打印指定的字符。要打印的字符用一个括在双引号(")中的字符串来表示。当运行程序时,系统会调用println方法打印第一个句子,再调用该方法打印第二个句子,然后,因为到达了main方法的最后一行,程序结束。

调用println方法时运行的代码并没在这个程序中定义。println方法属于Java标准类库的System.out对象。从技术上看这不属于Java语言,但在任何Java程序中都常常使用。第2章再详细介绍println方法。

1.1.2 注释

让我们再详细研究一下注释。注释只具有语言的特色,程序员可以独立于代码之外写注释,以相互交流设计思想。注释应该表达程序员最初的意图。程序常常使用很多年,随着时间的推移,也常常做许多修改。当在未来的某个时刻需要修改程序时,原来的程序员往往不会再记得程序的具体细节。而且通常不是最初的那个程序员来修改程序;这样,完全不熟悉程序的某个人必须要理解这个程序。所以需要好的文档。

具体到Java程序语言,注释的内容可以是任何的文本。计算机会忽略注释;它们不影响程序的执行。

Lincoln程序中的注释代表了Java允许的两类注释中的一种。Lincoln程序中的注释形式如下:

```
// This is a comment.
```

这类注释以双斜杠//开头,一直到行尾结束。两个斜杠中间不允许加入任何字符。计算机会忽略双斜杠后面直到行尾之间的所有文本。注释可以与代码位于同一行,用来对该行代码做说明,如下所示:

```
System.out.println ("Montyln Report"); // always use this title
```

第二种Java注释如下所示:

```
/* This is another comment. */
```

这类注释不是用行尾来表示注释的结束。从斜杠星号(/*)开始到星号斜杠(*/)结束之间的任何内容都是注释,包括不可见的表示一行结束的换行符也算在内。所以,这类注释可以扩展在多行中。斜杠和星号之间不能有空格。

如果注释的开头在/*后面有第二个星号,则可使用javadoc工具将注释的内容自动生成有关程序的外部文档。关于javadoc的详细内容请参见附录I。

可使用两类基本注释创建不同的文档风格,例如:

```
// This is a comment on a single line.
```

```
/*-----  
// Some comments such as those above methods or classes  
// deserve to be blocked off to focus special attention  
// on a particular aspect of your code. Note that each of  
// these lines is technically a separate comment.  
//-----*/
```

```
/*  
This is one comment  
that spans several lines.  
*/
```

程序员常关注于写代码而较少将注意力放在写文档上。你应该养成好的注释风格并形成习惯。注释应该要好好写,通常要用完整的句子来写。它们不应赘述显而易见的内容,而应