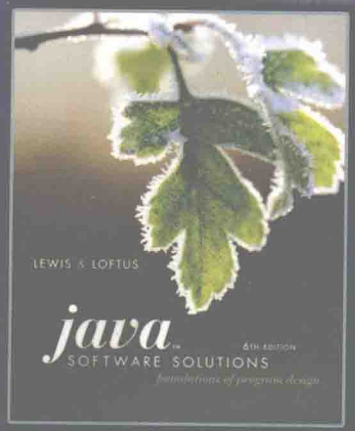
国外计算机科学教材系列

PEARSON

Java程序设计教程

（第六版）

Java Software Solutions
Foundations of Program Design, Sixth Edition



英文版

[美] John Lewis
William Loftus 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

国外计算机科学教材系列

Java 程序设计教程

（第六版）

（英文版）

Java Software Solutions

Foundations of Program Design

Sixth Edition

[美] John Lewis
William Loftus 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书对面向对象的思想和机制进行了准确而透彻的剖析，为读者深入学习Java语言程序设计提供了全面、详细的指导。全书覆盖了面向对象设计的广泛内容，介绍了Java语言的基本数据类型、流程控制、类和对象等。在深入分析面向对象设计方法的基础上，介绍了封装机制、继承机制和多态性的实现与应用、异常的捕捉和处理、集合类的定义方法和泛型类的概念，书中提供了大量具有可实践性的程序实例、自测题及答案、练习题和编程项目、生动的“软件失误案例”和“视频讲解”学习辅导短片，并且在每章中都补充了针对本章主题的图形用户界面（GUI）设计知识和实例，这部分内容可形成完整的Java GUI设计知识体系。此外本书还有内容丰富的附录。

本书适合作为高等院校软件专业与计算机应用专业的教材，同样对工程技术人员也有很高的参考价值。

版权贸易合同登记号　图字：01-2009-1390

图书在版编目（CIP）数据

Java程序设计教程：第6版：英文/（美）刘易斯（Lewis, J.），（美）洛夫特斯（Loftus, W.）著.
北京：电子工业出版社，2009.6
（国外计算机科学教材系列）
书名原文：Java Software Solutions: Foundations of Program Design, Sixth Edition
ISBN 978-7-121-08808-7

Ⅰ.J…　Ⅱ.①刘…　②洛…　Ⅲ.JAVA语言-程序设计-教材-英文　Ⅳ.TP312

中国版本图书馆CIP数据核字（2009）第073262号

责任编辑：冯小贝
印　　刷：北京市天竺颖华印刷厂
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
开　　本：787 × 980　1/16　　印张：52　　字数：1165千字
印　　次：2009年6月第1次印刷
定　　价：85.00元

凡所购买电子工业出版社的图书有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳—希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

前　言

欢迎使用本书第六版，多年来本书能够满足广大师生的需求并为教学服务，这使作者深感欣慰。第六版进一步加强了计算技术导论方面的教学，特别是增强了对教师的支持力度。

与以前的版本相比，本书的整个组织结构没有显著修改，教师与学生的反馈意见表明这种结构正符合他们的需求。本书仍然注重强调基础的核心概念，并在每一章的“图形设计之路”(Graphics Track)中分别讨论图形和图形用户界面 (GUI)，为图形设计的教学提供了极大的灵活性。此外，本书轻松随意的文风和颇有趣味的程序实例至今仍然对读者保持着一定的吸引力。

本书内容的一些增强之处是提供了有助于学生获得更好的程序设计体验的学习资料。各章的自测题已移到该章的末尾，并增加了更多的题目，使学生能随着学习过程进行自测。本书还增加了贯穿全书的新内容，称为“软件失误案例”。在每个“软件失误案例”中都讲述了一个典型的软件问题引发的事故，进而强调软件设计在软件系统开发中所扮演的重要角色。

本书还新增了易激发读者兴趣的“视频讲解”短片，这些“视频讲解”深入结合于每一章，由作者开发和主讲，目的是通过一些从设计到编码的关键主题的实例，展示如何完整地解决一些软件开发问题，为读者提供更多的帮助。这些“视频讲解”学习指导短片可以从与本书相关的网站或CD中获取。

本书的基石

本书是基于下述的基本思路和观点编写的，相信这有助于使其成为一本优良的教材：

- **真正的面向对象。**一本真正介绍完全面向对象方法的教材必须使用对象语言，即所有问题都应该用面向对象的术语来讨论。但这并不意味着学生看到的第一个程序就必须讨论多个类和方法的写法，学生应当在学会编写对象之前先学会使用对象。本书采用一种自然的学习进程，使学生逐步达到具有设计实际的面向对象解决方案的能力。
- **良好的程序设计实践。**重要的不是教学生如何编程，而是教学生如何写出好软件，编程与编写软件是有差别的。编写软件并不是按照说明完成一套动作，一个好程序远非仅仅是一组语句的集合。本书汇集了一些可作为良好程序设计技巧基础的实践经验，不仅应用于本书所有的实例，还要在讨论中进一步强化，学生将学会如何解决问题及如何实现设计方案。全书贯穿了软件工程基本技术的介绍，并使得所介绍的技术形成完整的体系结构。新的“软件失误案例”通过生动的实例展示不遵守合理可靠的程序设计实践原则所带来的危险，进一步复习了软件工程基本技术课程。

- **充足的程序实例**。学生将通过程序实例来学习。本书提供了大量已完整实现的实例来描述特定的概念。书中穿插列举了易理解的小例子和实用的大例子以相辅相成，并对图形和非图形实例的采用进行了合理的选择。新的“视频讲解”以生动的表现形式提供了更多的程序设计实例。
- **图形与图形用户界面（GUI）**。图形能够激发学生极大的学习兴趣，而且图形设计可以用做卓越的面向对象应用实例。因此，本书各章都安排了一组精心设计的章节，称为“图形设计之路”，包含了事件处理和图形用户界面的使用。随着图形设计主题的进展，学生将很自然地学会以适当的方式建立GUI。对于不需要涉及图形设计的学生，则可以完全忽略“图形设计之路”。

各章概述

第1章（计算机系统概述）一般性地介绍计算机系统，包括计算机基本结构和硬件、网络、程序设计及语言翻译。同时对Java也进行了介绍，并讨论了通用程序开发的基础和面向对象程序设计。本章广泛地涵盖了学生所要熟悉的开发环境介绍。

第2章（数据与表达式）探讨Java程序中使用的一些基本数据类型和计算表达式的用法，讨论了数据类型间的转换及如何借助标准的Scanner类与用户进行交互式输入。

第3章（使用类和对象）探讨预定义类的使用及其对象的创建，这些类和对象用于处理字符串、产生随机数、执行复杂的计算和格式化输出。此外还讨论了枚举数据类型。

第4章（编写类）探讨与编写类和方法相关的问题，包括实例数据、可见性、引用范围、方法参数和返回类型。此外还讨论了封装和构造方法。与编写类相关的其他问题或者更深入的内容将在第6章讨论。

第5章（条件和循环语句）讨论用于判断的布尔表达式的用法，介绍所有与条件判断和循环相关的语句，包括已增强功能的新loop语句。此外进一步讨论用Scanner类重复接收和解析输入数据及读取文本文件的方法。

第6章（面向对象设计）进一步深入和扩展类设计问题的讨论，包括解决一个问题所需要的类和对象的识别技术及类间的关联关系，同时还讨论了静态类成员、接口、枚举型类的设计。对于方法设计和方法重载也进行了讨论。

第7章（数组）介绍数组和数组处理的扩展内容，包括命令行参数、可变长度参数列表和多维数组。对ArrayList类及其作为通用类型的用法也进行了探讨。

第8章（继承）介绍了类的派生及其相关的概念，例如类层次结构、优先性和可见性。本章着重讨论合理地使用继承性，以及在软件设计中使用继承的原则。

第9章（多态性）探讨了绑定的概念及绑定与多态性的关系，然后讨论了如何使用继承或接口实现多态性，并利用排序实例示范多态性机制，最后讨论了有关多态性的程序设计问题。

第10章（异常）探讨了Java标准库中用于定义异常的类层次结构，以及定义用户异常对象的方法。同时还讨论了处理输入/输出时的异常使用方法，并分析了一个编写文本文件的程序示例。

第11章（递归）介绍了递归的概念、递归的实现及合理使用递归的方法，并利用几个不同应用领域的程序实例，示范了递归技术如何使得某些特定问题解决得更完美。

第12章（集合）介绍集合的概念及其基本数据结构，并进一步深入探讨了抽象概念和经典的数据结构，此外还介绍了泛型。本章可作为CS2课程的入门介绍。

补充资料

学生使用的光盘[①]

该光盘包括以下内容：

- 书中所有程序的源代码
- 各种Java开发环境

可以到网站www.aw.com/cssupport上获得上述光盘所提供的大部分资源。

MyCodeMate

Addison-Wesley的MyCodeMate是一个教材专用的Web资源，可以为学生的程序设计提供指导性帮助和评估。本书的全部程序例子和选自每一章的编程项目都集成于MyCodeMate中。学生使用该工具可以通过因特网在任何一台计算机上编写程序和编译程序，并且可以在如何进展、如何定位编译器错误信息等方面接受指导并获取反馈信息。教师可以记录每个学生完成本书编程项目时的进展情况。要获取更多的信息，可访问www.mycodemate.com。

教师资源

下列的补充资源仅供教师使用，访问Addison-Wesley教师资源中心（www.aw.com/irc）或发送邮件给computing@aw.com，可以获取如何得到下述资源的信息[②]：

- 幻灯片——使用PowerPoint制作的幻灯片。
- 解答——包括习题解答和编程项目解答。
- 测试库（具有功能强大的试题生成软件）——包括大量的简答题、多项选择题和判断题。
- 实验手册——实验练习用于辅助和配合教材中的知识点教学。

致谢

非常感激遍布世界各地的教师和学生们对本书以前各个版本提出的宝贵意见与建议，我们很高兴地看到教师对学生的深切关心和学生对知识的渴求，并欢迎你们继续提出评论和问题。

Dan Joyce是完成新的自测题及其答案工作的主要承担者，并确保有足够的自测题覆盖每一个相关的主题。

① 考虑到出版需求，本书未配有光盘。全部光盘内容可在电子工业出版社华信教育资源网www.huaxin.edu.cn上下载。

② 授课教师也可与电子工业出版社联系，联系方式见书后的教辅支持页。

我们不断地惊叹于Addison-Wesley出版小组的才能和成就。本书编辑Michael Hirsch具有惊人的洞察力和责任心；他的助手Stephanie Sellinger一直给我们提供帮助和支持；市场部经理Chris Kelly使我们确信教师们能够理解和认同本书在教学法方面的优势；有才华的Beth Paquin为本书进行了封面设计和全书版面的美工设计；Jeff Holcomb为本书的出版付出了坚持不懈的努力。Addison-Wesley小组还得到了Nesbitt Graphics的一个思维敏锐的出版小组的支持，其中包括Jerilyn Bockorick、Kathy Smith、Harry Durding和Keith Lesko。我们感谢所有确保这本书达到高质量出版标准的人们。

还要特别感谢以下通过专题组、会谈和评论为本书提供了宝贵建议的团体与个人：David Atkins –University of Oregon; Thomas W.Bennet – Mississippi College; James H.Cross II – Auburn University; Wanda M.Eanes –Macon State College; Stephanie Elzer – Millersville University; Marj Feroe – Delaware County Community College, Pennsylvania; Stephen Hughes – Roanoke College; Gheorghe Muresan – Rutgers University; Bennet Setzer – Kennesaw State University; Alexander Stoytchev – Iowa State University。

下面是以前各个版本的评审者及许多的教师和朋友，他们为本书提供了有价值的反馈信息：

Elizabeth Adams	James Madison University
Lewis Barnett	University of Richmond
Tom Bennet	Mississippi College
Gian Mario Besana	DePaul University
Hans-Peter Bischof	Rochester Institute of Technology
Robert Burton	Brigham Young University
John Chandler	Oklahoma State University
Robert Cohen	University of Massachusetts, Boston
Dodi Coreson	Linn Benton Community College
James Cross	Auburn University
Eman El-Sheikh	University of West Florida
Christopher Eliot	University of Massachusetts, Amherst
Matt Evett	Eastern Michigan University
John Gauch	University of Kansas
Chris Haynes	Indiana University
James Heliotis	Rochester Institute of Technology
Laurie Hendren	McGill University
Mike Higgs	Austin College
Saroja Kanchi	Kettering University
Karen Kluge	Dartmouth College
Jason Levy	University of Hawaii
Peter MacKenzie	McGill University
Blayne Mayfield	Oklahoma State University

Laurie Murphy	Pacific Lutheran University
Dave Musicant	Carleton College
Faye Navabi-Tadayon	Arizona State University
Lawrence Osborne	Lamar University
Barry Pollack	City College of San Francisco
B. Ravikumar	University of Rhode Island
David Riley	University of Wisconsin (La Crosse)
Jerry Ross	Lane Community College
Patricia Roth	Southeastern Polytechnic State University
Carolyn Schauble	Colorado State University
Arjit Sengupta	Georgia State University
Vijay Srinivasan	JavaSoft, Sun Microsystems, Inc.
Stuart Steiner	Eastern Washington University
Katherine St. John	Lehman College, CUNY
Ed Timmerman	University of Maryland, University College
Shengru Tu	University of New Orleans
Paul Tymann	Rochester Institute of Technology
John J. Wegis	JavaSoft, Sun Microsystems, Inc.
Linda Wilson	Dartmouth College
David Wittenberg	Brandeis University
Wang-Chan Wong	California State University (Dominguez Hills)

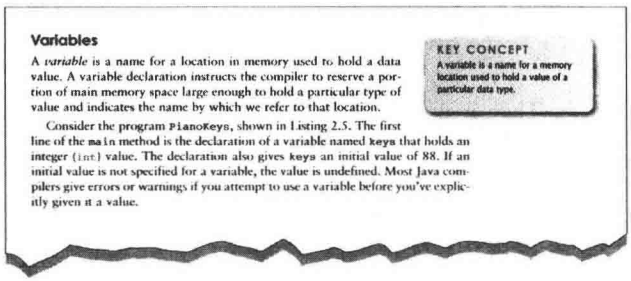
感谢我的朋友和以前在Villanova大学工作的同事，他们提供了很多的建设性意见，这些同事包括Bob Beck、Cathy Helwig、Anany Levitin、Najib Nadi、Beth Taddei及Barbara Zimmerman。

特别感谢New Jersey学院的Pete DePasquale，他设计并开发了绘图工具箱项目及提供了最初的Java类库附录。

还有许多朋友以不同的方式提供了帮助，包括Ken Arnold、Mike Czepiel、John Loftus、Sebastian Niezgoda和Sammy Perugini。在此还向所有在致谢中遗漏了名字的朋友们致以深深的歉意。

计算机科学教育（SIGCSE）的ACM特别兴趣组是一个巨大的教育信息交流资源，他们的会议给来自不同层次与不同类型的学校的教师提供了交流教育新观点、新思路和资源的机会。对于从事某一计算领域的教育工作者来说，如果还未加入SIGCSE，就会错过参与这种教育信息交流的机会。

Feature Walkthrough



Variables

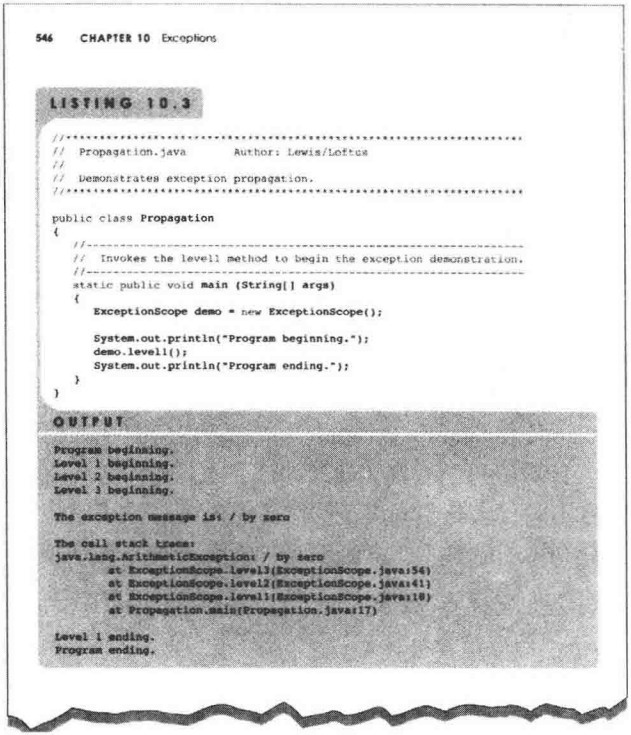
A *variable* is a name for a location in memory used to hold a data value. A variable declaration instructs the compiler to reserve a portion of main memory space large enough to hold a particular type of value and indicates the name by which we refer to that location.

KEY CONCEPT
A variable is a name for a memory location used to hold a value of a particular data type.

Consider the program `PianoKeys`, shown in Listing 2.5. The first line of the `main` method is the declaration of a variable named `keys` that holds an integer (`int`) value. The declaration also gives `keys` an initial value of 88. If an initial value is not specified for a variable, the value is undefined. Most Java compilers give errors or warnings if you attempt to use a variable before you've explicitly given it a value.

Key Concepts. Throughout the text, the Key Concept boxes highlight fundamental ideas and important guidelines. These concepts are summarized at the end of each chapter.

Listings. All programming examples are presented in clearly labeled listings, followed by the program output, a sample run, or screen shot display as appropriate. The code is colored to visually distinguish comments and reserved words.



546 CHAPTER 10 Exceptions

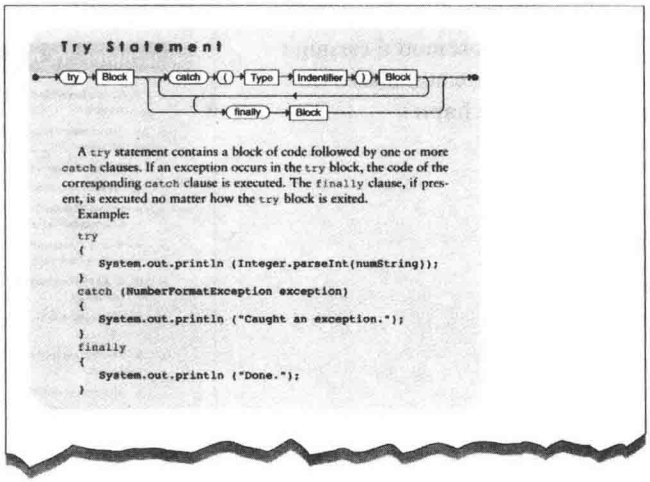
LISTING 10.3

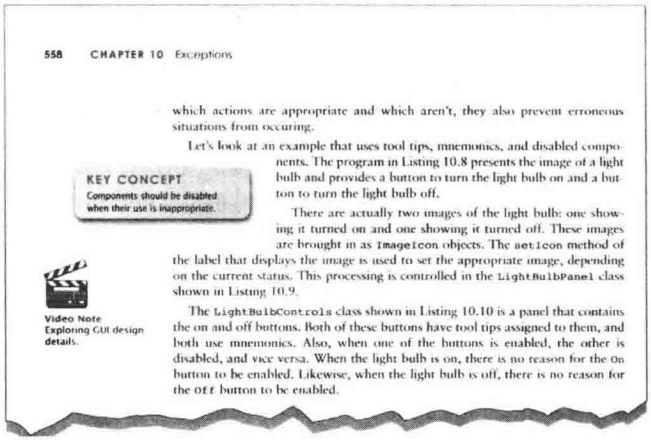
```
//********************************************************************
//  Propagation.java       Author: Lewis/Loftus
//
//  Demonstrates exception propagation.
//********************************************************************

public class Propagation
{
   //-----------------------------------------------------------------
   //  Invokes the level1 method to begin the exception demonstration.
   //-----------------------------------------------------------------
   static public void main (String[] args)
   {
      ExceptionScope demo = new ExceptionScope();

      System.out.println("Program beginning.");
      demo.level1();
      System.out.println("Program ending.");
   }
}
```

OUTPUT

```
Program beginning.
Level 1 beginning.
Level 2 beginning.
Level 3 beginning.

The exception message is: / by zero

The call stack trace:
java.lang.ArithmeticException: / by zero
        at ExceptionScope.level3(ExceptionScope.java:54)
        at ExceptionScope.level2(ExceptionScope.java:41)
        at ExceptionScope.level1(ExceptionScope.java:18)
        at Propagation.main(Propagation.java:17)

Level 1 ending.
Program ending.
```

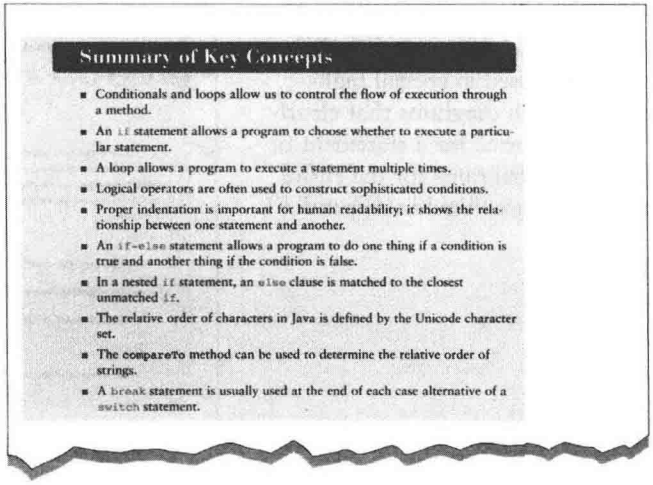
Syntax Diagrams. At appropriate points in the text, syntactic elements of the Java language are discussed in special highlighted sections with diagrams that clearly identify the valid forms for a statement or construct. Syntax diagrams for the entire Java language are presented in Appendix L.





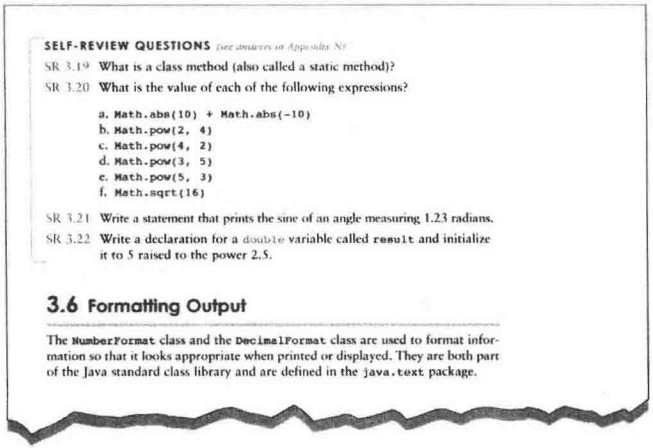
Graphics Track. All processing that involves graphics and graphical user interfaces is discussed in one or two sections at the end of each chapter that we collectively refer to as the Graphics Track. This material can be skipped without loss of continuity, or focused on specifically as desired. The material in any Graphics Track section relates to the main topics of the chapter in which it is found. Graphics Track sections are indicated by a brown border on the edge of the page.

Summary of Key Concepts. The Key Concepts presented throughout a chapter are summarized at the end of the chapter.



Summary of Key Concepts

- Conditionals and loops allow us to control the flow of execution through a method.
- An `if` statement allows a program to choose whether to execute a particular statement.
- A loop allows a program to execute a statement multiple times.
- Logical operators are often used to construct sophisticated conditions.
- Proper indentation is important for human readability; it shows the relationship between one statement and another.
- An `if-else` statement allows a program to do one thing if a condition is true and another thing if the condition is false.
- In a nested `if` statement, an `else` clause is matched to the closest unmatched `if`.
- The relative order of characters in Java is defined by the Unicode character set.
- The `compareTo` method can be used to determine the relative order of strings.
- A `break` statement is usually used at the end of each case alternative of a `switch` statement.



SELF-REVIEW QUESTIONS *(see answers in Appendix N)*

SR 3.19 What is a class method (also called a static method)?

SR 3.20 What is the value of each of the following expressions?

```
a. Math.abs(10) + Math.abs(-10)
b. Math.pow(2, 4)
c. Math.pow(4, 2)
d. Math.pow(3, 5)
e. Math.pow(5, 3)
f. Math.sqrt(16)
```

SR 3.21 Write a statement that prints the sine of an angle measuring 1.23 radians.

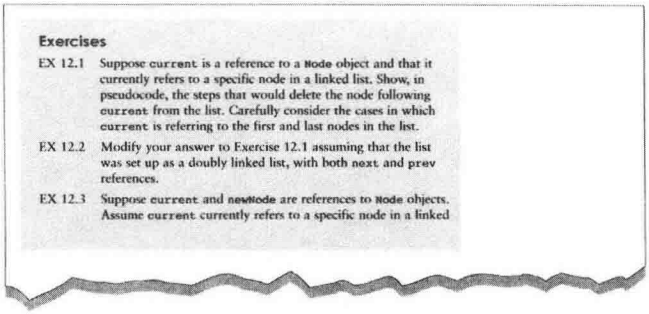
SR 3.22 Write a declaration for a `double` variable called `result` and initialize it to 5 raised to the power 2.5.

3.6 Formatting Output

The `NumberFormat` class and the `DecimalFormat` class are used to format information so that it looks appropriate when printed or displayed. They are both part of the Java standard class library and are defined in the `java.text` package.

Self-Review Questions and Answers. These short-answer questions review the fundamental ideas and terms established in the preceding section. They are designed to allow students to assess their own basic grasp of the material. The answers to these questions can be found at the end of the book in Appendix N.

Exercises. These intermediate problems require computations, the analysis or writing of code fragments, and a thorough grasp of the chapter content. While the exercises may deal with code, they generally do not require any online activity.

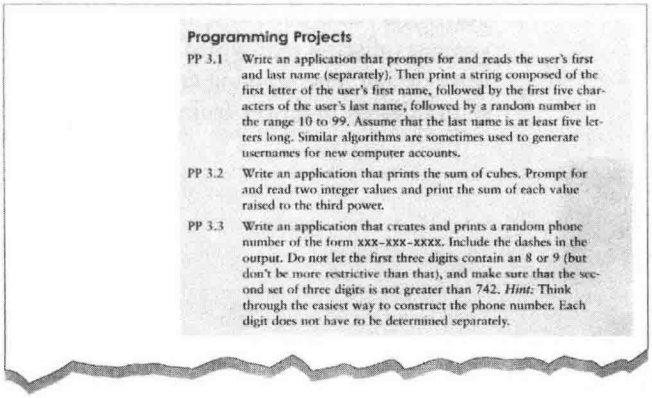


Exercises

EX 12.1 Suppose `current` is a reference to a `Node` object and that it currently refers to a specific node in a linked list. Show, in pseudocode, the steps that would delete the node following `current` from the list. Carefully consider the cases in which `current` is referring to the first and last nodes in the list.

EX 12.2 Modify your answer to Exercise 12.1 assuming that the list was set up as a doubly linked list, with both `next` and `prev` references.

EX 12.3 Suppose `current` and `newNode` are references to `Node` objects. Assume `current` currently refers to a specific node in a linked



Programming Projects

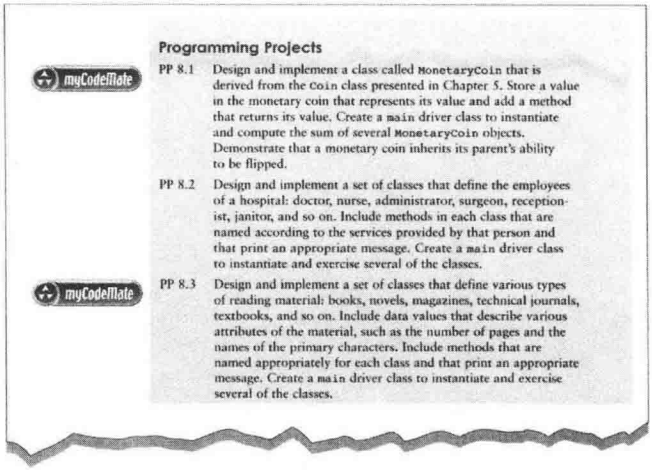
PP 3.1 Write an application that prompts for and reads the user's first and last name (separately). Then print a string composed of the first letter of the user's first name, followed by the first five characters of the user's last name, followed by a random number in the range 10 to 99. Assume that the last name is at least five letters long. Similar algorithms are sometimes used to generate usernames for new computer accounts.

PP 3.2 Write an application that prints the sum of cubes. Prompt for and read two integer values and print the sum of each value raised to the third power.

PP 3.3 Write an application that creates and prints a random phone number of the form `xxx-xxx-xxxx`. Include the dashes in the output. Do not let the first three digits contain an 8 or 9 (but don't be more restrictive than that), and make sure that the second set of three digits is not greater than 742. *Hint:* Think through the easiest way to construct the phone number. Each digit does not have to be determined separately.

Programming Projects. These problems require the design and implementation of Java programs. They vary widely in level of difficulty.

Addison-Wesley's MyCodeMate. Working online, students can view, compile, run, and edit select programming problems and all code listings from the textbook. Look for this MyCodeMate icon to see which Programming Projects are available with your included online subscription to MyCodeMate.

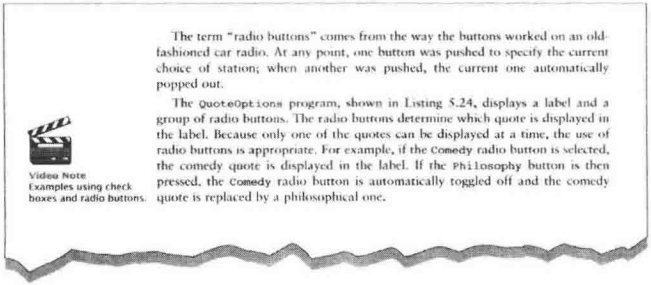


Programming Projects

myCodeMate PP 8.1 Design and implement a class called `MonetaryCoin` that is derived from the `Coin` class presented in Chapter 5. Store a value in the monetary coin that represents its value and add a method that returns its value. Create a `main` driver class to instantiate and compute the sum of several `MonetaryCoin` objects. Demonstrate that a monetary coin inherits its parent's ability to be flipped.

PP 8.2 Design and implement a set of classes that define the employees of a hospital: doctor, nurse, administrator, surgeon, receptionist, janitor, and so on. Include methods in each class that are named according to the services provided by that person and that print an appropriate message. Create a `main` driver class to instantiate and exercise several of the classes.

myCodeMate PP 8.3 Design and implement a set of classes that define various types of reading material: books, novels, magazines, technical journals, textbooks, and so on. Include data values that describe various attributes of the material, such as the number of pages and the names of the primary characters. Include methods that are named appropriately for each class and that print an appropriate message. Create a `main` driver class to instantiate and exercise several of the classes.



The term "radio buttons" comes from the way the buttons worked on an old-fashioned car radio. At any point, one button was pushed to specify the current choice of station; when another was pushed, the current one automatically popped out.

The `QuoteOptions` program, shown in Listing 5.24, displays a label and a group of radio buttons. The radio buttons determine which quote is displayed in the label. Because only one of the quotes can be displayed at a time, the use of radio buttons is appropriate. For example, if the `Comedy` radio button is selected, the comedy quote is displayed in the label. If the `Philosophy` button is then pressed, the `Comedy` radio button is automatically toggled off and the comedy quote is replaced by a philosophical one.

Video Note
Examples using check boxes and radio buttons.

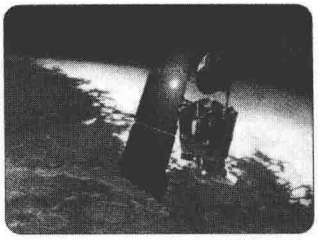
Video Notes. Presented by the author, Video Notes explain topics visually through informal videos in an easy-to-follow format, giving students the extra help they need to grasp important concepts. Look for this Video Note icon to see which in-chapter topics and end-of-chapter Programming Projects are available as Video Notes.

SOFTWARE FAILURE

NASA Mars Climate Orbiter and Polar Lander

What Happened?

As part of a series of missions exploring Mars, NASA launched the Mars Climate Orbiter in December, 1998, and the Mars Polar Lander in January, 1999. The two-spacecraft mission was designed to observe the atmospheric conditions on Mars through each of its seasons. The orbiter and the lander would have collected data about temperature, dust, water vapor, clouds, and the amount of carbon dioxide (CO_2) added and removed from the Martian pole regions.



* Artist's conception of the Mars Climate Orbiter

After its nine-month journey, the orbiter arrived at Mars in September, 1999, and fired its main engines to establish an orbit. The orbiter passed behind the planet (from Earth's perspective) five minutes later as planned, but NASA could not reestablish contact with it after expecting it to emerge. Review of the data showed that the altitude of the orbiter when it was entering orbit was only 57 kilometers, whereas the planned altitude was 140 kilometers. The minimum survivable altitude was between 85 and 100 kilometers. NASA concluded that the orbiter was destroyed by atmospheric friction.

Software Failures. These between-chapter vignettes discuss real-world flaws in software design, encouraging students to adopt sound design practices from the beginning.

Contents