



通过阅读本书，你可以借助J2SE 5.0这个最新、最出色的Java平台来学习Java编程。J2SE 5.0拥有强大的新特性和新功能，使Java性能产生了巨大的飞跃。

新的Java平台所带来的丰富内涵使本书相比于过去的版本似乎稍厚了些，但是并不说明新的内容更复杂。当然，要想成为熟练的Java程序员，所需的基础知识会略有增多，但都不困难。只要有进取心和一点点对编程的领悟力，你就可以得到足够多的知识，成为一个高效的Java程序员。

本书提供了规范化的学习方法，可帮助你事半功倍地完成学习。你不仅会学到一项相当流行的技术，而且做完一件值得做的事情将使人心旷神怡。

尝试读本书吧，你肯定会喜欢的！

Ivor Horton's
Beginning Java 2, JDK 5 Edition

Java 2入门经典

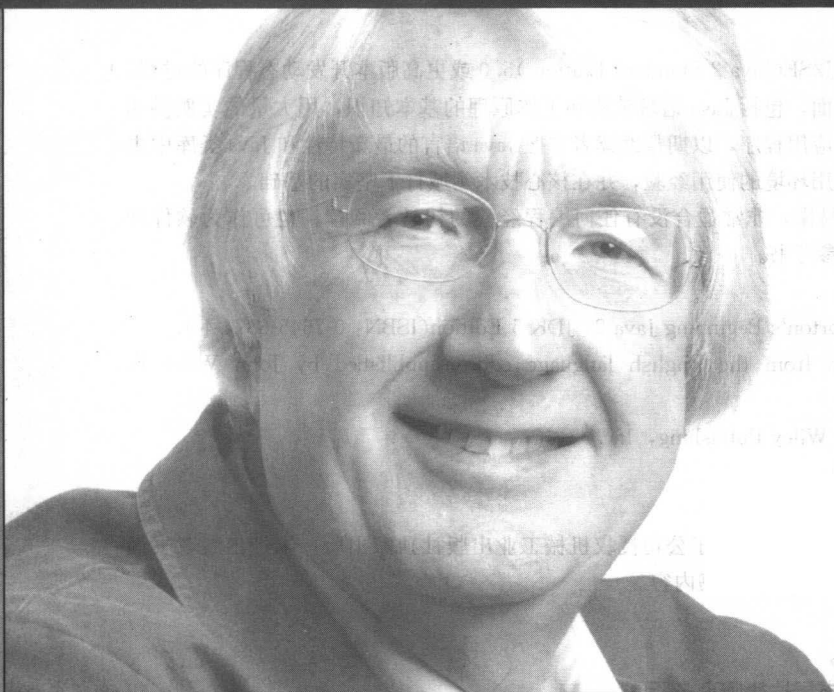
JDK 5

(美) Ivor Horton 著

潘晓雷 于浚泊 王丹 等译



机械工业出版社
China Machine Press



TP312
1807

Ivor Horton's
Beginning Java 2, JDK 5 Edition

Java 2入门经典

JDK 5

(美) Ivor Horton 著
于浚泊 王丹 等译



机械工业出版社
China Machine Press

本书综合介绍了使用 J2SE(Java 2 Standard Edition)5.0 或更高版本开发动态程序的过程,阐述了 Java 语言的各个方面,包括 Java 语言结构和工作原理的基本知识,用大量的实例讲述如何开发功能齐全的 Java 应用程序,以期帮助读者掌握 Java 语言的最新特性和 Java 类库中主要的功能包,积累 Java 应用环境的使用经验,并在核心技术领域打下坚实的基础。

本书系统全面、浅显易懂,非常适合没有任何编程经验的初学者阅读,也可作为软件开发人员和高校师生的必备参考书。

Ivor Horton: Ivor Horton's Beginning Java 2, JDK 5 Edition(ISBN: 0-7645-6874-4).

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

Copyright © 2005 by Wiley Publishing, Inc.

All rights reserved.

本书中文简体字版由约翰·威利父子公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

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:01-2005-2763

图书在版编目(CIP)数据

Java 2 入门经典:JDK 5/(美)霍顿(Horton I.)著;潘晓雷等译.一北京:机械工业出版社,2006.1

书名原文:Ivor Horton's Beginning Java 2: JDK 5 Edition

ISBN 7-111-18089-5

I.J… II.①霍… ②潘… III.JAVA 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2005)第 148253 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:姜淑欣 孙笑竹

北京京北制版厂印刷·新华书店北京发行所发行

2006 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·67.75 印张

定价:118.00 元

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

译者序

也许读者是一位 C++ 或其他语言的程序员，也许读者从未接触过任何编程语言。无论如何，学习 Java 语言都是非常好的选择。这个理由有些是技术上的，有些是非技术上的。从技术上讲，Java 语言具有如下优点：

- 简单。Java 最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。其语法风格类似于 C++，但摒弃了 C++ 中容易引发程序错误的地方，同时提供了丰富的类库。
- 面向对象。Java 语言的设计完全是面向对象的。除了基本数据类型外，Java 的所有数据都是用对象表示的。
- 分布式。Java 同时支持数据分布和操作分布。对于数据分布，Java 提供了一个叫做 URL 的对象，利用这个对象，可以打开并访问相同 URL 地址上的对象，访问方式与访问本地文件系统相同。对于操作分布，Java 的 applet 小程序可以从服务器下载到客户端，即部分计算在客户端进行，提高系统执行效率。
- 可靠性和安全性高。因为 Java 最初的设计致力于电子类消费产品，所以要求较高的可靠性。首先，Java 是强类型的语言，要求显式的方法声明，这保证了编译器可以发现方法调用错误，保证程序更加可靠；其次，Java 不支持指针，这杜绝了内存的非法访问；第三，Java 的自动单元收集避免了内存丢失等动态内存分配导致的问题；第四，Java 解释器运行时实施检查，可以发现数组和字符串访问的越界。最后，Java 提供了异常处理机制，程序员可以把一组错误代码放在一个地方，这样可以简化错误处理任务，便于恢复。
- 可移植性。这一直是 Java 程序开发人员的精神指标，也是 Java 之所以能够受到他们喜爱的原因之一。大多数编译器产生的目标代码只能运行在一种 CPU 上，即使那些能支持多种 CPU 的编译器也不能同时产生适合多种 CPU 的目标代码。但 Java 编译器不同，它产生的目标代码(J-Code)是针对 Java 虚拟机(Java Virtual Machine)的，而不是实际存在的 CPU。Java 虚拟机能掩盖不同 CPU 之间的差别，使 J-Code 能运行于任何具有 Java 虚拟机的机器上。
- 多线程。Java 在两方面支持多线程：一方面，Java 环境本身就是多线程的。若干个系统线程运行负责必要的无用单元回收、系统维护等系统级操作；另一方面，Java 语言内置多线程控制，可以大大简化多线程应用程序的开发。
- 动态。Java 的动态特性是其面向对象设计方法的发展，它允许程序动态地装入运行过程中所需要的类。
- 支持 Unicode 码。Java 使用 Unicode 作为它的标准字符，这项特性使得 Java 的程序能在不同语言的平台上编写和执行。Java 也是目前所有计算机语言当中，惟一“天生”使用 Unicode 的语言。

从非技术的角度谈学习 Java 的益处是 Java 的应用非常广泛，如果读者靠编写程序为生，那么迟早都会接触到 Java。Java 已经渐渐地渗透到各个领域，可以用它写出 Java Servlet，将其挂在 apache 或其他

Web 服务器上；还可以写出 Java applet，在网页浏览器上执行；甚至可以写出数据库的存储过程，然后安装到 Oracle 上。另外，Java 是免费的，Sun 公司提供 Java 相关的工具，允许用户开发并执行 Java 程序。用户无须支付授权费，也不用花钱买 Java 开发工具；而且可以从网站上免费获得 Java 的信息文件。

基于上述原因，学习 Java 语言着实是件不错的事。本书是学习 Java 语言的理想教材，同前几版相比较，本书增添了许多对 Java 2 中新引进技术的介绍。从 Java 产生到现在，Java 2 是第一次真正意义上对语言实现进行了大刀阔斧的改造。这包括泛型、增强的 for 循环、自动装箱/拆箱、枚举类型、可变参数、静态导入、格式化的输入/输出、元数据等，这些新特性很大程度上简化了开发。

本书作者 Ivor Horton 著有许多优秀的 Java 教程，他独到的编写风格赢得了全球数十万的读者，多部作品被评为最受欢迎的 Java 编程流行书籍。本书继承了他一贯的风格，并具有如下特点：

- 本书的目标是为读者成为优秀的程序员打下坚实的基础，因此本书着重帮助读者掌握 Java 语言各种元素在软件开发中的使用。
- 本书在介绍新的特性和概念之后都会精心安排一些示例，并详细地讲解示例程序的工作原理。同时各个章节后面都设有相应的练习，帮助读者从更深的层次上理解 Java 的编程思想。
- 本书的结构设计得十分到位，对于初次接触编程的读者，只需追随作者的思路即可；对于已经熟悉前几版 Java 特性的程序员，可以快速地了解有关 Java 的新特性。
- 本书介绍 Java 在特定领域(例如数据库和 XML)的应用时，不仅讲解 Java 语言本身对应用的支持，而且生动、详尽地介绍了相关的知识。

本书讲解深入浅出、结构合理，并提供了非常丰富的实例，非常适合作为编程初学者的自学教材；同时由于其涉及领域广泛，也可以作为具有一定编程经验的软件开发人员的参考手册。

本书由潘晓雷组织翻译并负责全书译稿的统稿和审校，黄晓博负责初稿的文字校对。本书翻译具体分工如下：潘晓雷翻译前言、序、第 1 章、第 13 章、第 14 章、附录 A、附录 B；谷云翻译第 2 章、第 3 章；聂振辉翻译第 4 章、第 8 章、第 15 章；王丹翻译第 5 章～第 7 章；禹金璐翻译第 9 章～第 12 章、第 16 章；侯文翻译第 17 章、第 18 章；赵晓岩翻译第 19 章、第 20 章；赵毅翻译第 21 章、第 24 章、第 25 章；于浚泊翻译第 22 章、第 23 章。另外，海利顺翻译部为本书的翻译工作提供了一定的帮助。

译著文字力求信、达、雅，但由于译者水平和时间有限，译作中难免有疏漏之处，恳请专家和读者批评指正。

潘晓雷

中国科学院青年公寓

序

在阅读这篇序言之前，读者的脑海里可能有些疑问。首先，这是不是一本适合我的书，书中的材料是不是最新的，还有书的内容是不是反映了最终版 API？其次，面对这么厚的一本书，我该学习哪些东西？从哪里开始学习？

我看过很多序言，其中大部分都是给读者讲一段轶事或者故事，然后稍稍介绍一下作者，但最后都不能为上述问题给出更好的答案。为了避免落入这种俗套，首先我要开门见山地回答上面第二个问题：这是一本值得从第一页一口气读到最后一页的书。如果读者从来没有接触过 Ivon Horton 的著作，那一定会感到惊喜。Ivon Horton 撰写的著作通俗易懂，非常好读。这一点我是了解的，因为我已经把整本书从头到尾读了一遍。

本书的本版是基于 J2SE 5.0 平台的。J2SE 5.0 历经 3 年开发，是多年来 Java 平台最重大的更新之一。此版本的开发过程中有世界范围内 160 多名专家参与，他们都参与 Java Community Process，致力于为所有开发者提供一款更好的开发平台。我本人也以 JSR 176 的 Specification Lead 身份参与其中，而 JSR 176 就定义了 J2SE 5.0 的内容。因此，我最关心的地方在于确保本书的精确性以及和最终版 API 集相一致。我甚至把本书每个代码实例编译运行了两次，发现了大量非常棒的代码实例，这一点读者不久就会发现。

那么从本书可以学到什么呢？首先，作者会阐述基本的编程部件，帮助读者完成自己的第一个 Java 程序。然后，一步一步介绍 Java 语言，包括新的泛型、功能更强的 for 循环、枚举类型等等。在后面的章节里，读者还会接触到 Java 语言的最新特性，甚至学到其他一些非语言的特性，譬如 XML、DOM3 的更新等。所以，不论是初学者，还是已经具备一些 Java 编程经验的人，都能从本书中学到使用最新版 Java 所必需的技术。

最后，建议读者好好阅读这本书，享受 JDK 5.0 带来的一切，最终发现原来使用 J2SE 5.0 是这么简单！

Calvin Austin

Sun 公司 J2SE 5.0 规范制订者之一

前言

欢迎选用本书

欢迎选用本书。本书可指导读者学会 Java 语言和 Java 平台应用程序接口 (Application Program Interface, API)，理解和掌握使用 J2SE 5.0 或者更高版本开发程序的基本技巧。

本书不仅向读者教授 Java 语言，还介绍了作为 Java 程序员应该了解的各种相关知识。本书合理安排内容、结构，以便读者精心安排进度，循序渐进地学习，其中每一阶段学习内容均以前面的已学内容为基础。

本书的读者对象

Java 程序设计是一个宽泛的领域，而且随着 Java 语言功能的改善和相关类库的扩充，它还在不断发展。Java 是一种为互联网编程、跨平台开发和教学目的而设计的面向对象语言，自发布以来便发展迅猛。这主要由于三点：Java 与生俱来的平台无关性；Java 语言形式简单却功能强大；以及对相当多的任务而言 Java 高效易用。很多大型应用软件的开发者都选择使用 Java 语言，因为与其他语言相比，Java 更加易于开发和维护，并且无须修改代码即可在各种结构的计算机和操作系统上运行。使用 Java 语言，程序员的工作将变得效率更高、更轻松。

本书的目标是帮助读者全面地理解 Java 语言，并通过对 Java 编程环境的体验在其每个核心领域打下坚实的基础。本书使用丰富的代码示例阐明了 Java 语言的各个方面，读者可以而且应该自己动手建立并且运行这些代码示例。每一章末尾均有练习题，以便读者学会运用所学知识。学完本书各章内容后，读者将能够独立编写功能丰富且富有效率的 Java 程序。

本书书名中的“入门”(Beginning)一词更多地是指讲述风格而非技能水平，因此本书也可叫做“Java 直通车”(Straight into Java)。这是因为不论读者是具有其他语言编程经验的程序员还是初学者，都能从本书的内容构架中找到适合自己层次的内容。

本书假定读者懂得最基本的一些编程知识，即至少了解程序工作原理的基本概念。但是先前没有一定编程经验的读者也能顺利地使用本书。本书的内容安排相当紧凑，但是并不缺少对 Java 程序的工作原理的必要解释和说明。

本书的内容

本书按照如下顺序向读者讲授 Java 语言的编程知识：

- 首先，介绍一些 Java 编程的基础术语和概念，以及后续内容的概要与框架，为日后的深入学习打好基础。
- 其次，清楚地解释 Java 语言的各项特性——包括基本数据类型、处理数据的控制结构、面向对

象、运行错误的处理方式以及线程等等，此外还解释了泛型的概念和使用方法。本书不但阐述了这些语言元素的含义，还告诉读者如何在实践中运用它们。

- 第三，全面介绍了 Java 类库中关键的包(package)，尤其是介绍并用大量实例说明了 java.math、java.io、java.nio、java.util、java.awt、java.awt.event、java.applet、javax.swing、javax.xml 和 javax.sql 包。这些包涵盖了文件处理、打印、集成类、帮助对象、图形用户界面(Graphical User Interface, GUI)、applet、XML 以及关系数据库等诸多方面的功能。
- 第四，通过介绍一个名为“Sketcher”的应用程序的开发全过程，指导读者在真实环境中学习运用 Java 语言功能和类库。在开发工程中，读者既要学会利用菜单、工具栏和状态栏等“组装”出程序的用户界面，也要实现绘制处理大量图形元素、编辑文本以及打印和保存文本处理结果(包括使用 XML 实现导入与导出)等功能。通过这种把功能函数组合成为应用程序的过程，读者能更好地理解如何在自己的实际项目中运用 Java 语言，这些经验是无法通过那些琐碎的示例得到的。
- 最后，介绍了 SQL 基础，并讲授了如何使用 JDK 自带的 JDBC 功能操作关系数据库。

作者在阐述上述主题的同时，还提供了一些原理性的介绍，并用具体实例予以清晰的说明。读者可以先快速浏览一遍，必要时再回顾有关基本原理。每章中的小例子主要用来阐述类及其使用方法，或者解释刚刚用到的新原理。这些小例子集中向读者展示某些特定的语言特性和工作原理。

为了获得最佳阅读效果，推荐读者在阅读的同时实践所碰到的实例。即使已下载了代码，最好还是亲手输入它们，实践一遍，这样做效果一定非同一般。这些例子为读者编写自己的程序提供了不错的代码基础和提示。实践是非常重要的，从第一次运行代码出现的错误中，读者能够收获很多。

本书所有实例源代码均可在 <http://www.wrox.com> 网站上获得。

阅读本书的前提

本书的实例源代码在 J2SE 5.0 版本下测试无误，读者需使用 JDK 5.0 或者以上版本。大部分章节的要求都不高，仅需要文本编辑器和可以运行 Java 工具的命令行窗口即可。第 1 章将给出这些要求的所有细节，并解释如何获得并安装这些指定软件。

本书约定

为了让读者获得最佳阅读效果，本书对文字编排做了一些约定。

在讨论代码时，使用 Courier 字体标识，对重要的、必须牢记的内容加上方框。例如：

诸如此类的方框提供了一些不容忘记的重要信息，这些信息与方框前后的文字有很直接的关系。

当介绍重要术语时，将使用**黑体**来突出强调。

b>

当表示组合按键时，使用 Ctrl+A 的形式。

命令行和终端输出显示为：

```
C:\> java ShowStyle
```

When the command line and terminal output are shown, it's in this style.

表现特定功能和机制的代码段将用灰色阴影予以识别：

```
if(life==aimless) {
    DoSomethingElse;
}
```


当部分改变某代码段时，被改变的部分也将用灰色阴影显示，如下所示：

```
if(life==aimless) {
    DoSomethingNew;
}
```

本书将通过“试一试”小节展示完整实例的部分代码。在“试一试”小节中，值得注意的是：代码段将被单独分离，重要的代码段加上了灰色阴影，表明程序的进展。当讲到重要的代码段时，本书会给出“程序工作原理”小节，以解释与之前所述理论相关的代码要点。

“试一试”小节中的全新代码段将添加灰色阴影，如下所示：

```
public class Motto {
    public static void main(String[] args) {
        System.out.println("He who fights and runs away");
        System.out.println("Lives to runs away another day.");
    }
}
```

由前一版本修改而来的新代码将局部添加灰色阴影，如下所示：

```
public class Saying {
    public static void main(String[] args) {
        System.out.println("I used to think I was indecisive");
        System.out.println("but now I'm not so sure.");
    }
}
```

最后，代码中出现的“程序工作原理”小节将不添加灰色阴影，如下所示：

```
System.out.println("I used to think I was indecisive");
System.out.println("but now I'm not so sure.");
```

源代码

在使用本书中的示例时，读者可以手工输入所有代码，也可直接使用本书附带的源代码文件。本书所有源代码均可从 <http://www.wrox.com> 下载。访问该网站时，只需找到本书书名(可通过搜索书名或者检索书名列表找到)，单击详细内容中的下载代码(Download Code)链接，即可得到本书所有源代码。

由于许多书的书名与本书雷同，最快捷的方法是利用 ISBN 搜索，本书的 ISBN 为 0-7645-6874-4。

下载代码后读者只需使用最习惯的工具予以解压即可。此外，读者还可以到 Wrox 主站的代码下载网页(<http://www.wrox.com/dynamic/books/download.aspx>)，可以得到本书以及其他由 Wrox 出版的书籍的相关源代码。

勘误和更新

尽管我们竭尽能力争文字和代码不出现错误，但是人非圣人，错误在所难免。如果读者在书中发现了错误(如拼写错误、代码错误等等)，希望能及时反馈给我们，从而可以提醒更多的人避免陷入同样的错误，也有利于提高本书质量。对此我们深表谢意。

若查看勘误表，请先登陆 <http://www.wrox.com>，使用搜索框或者书名列表找到本书书名，然后在详情页面中单击勘误表链接。在勘误表网页中，读者将会看到所有和本书相关的勘误信息。读者可在 www.wrox.com/misc-pages/booklist.shtml 中得到所有相关书籍的勘误表(附带有链接)。

p2p. wrox. com

若想与作者和其他读者讨论有关 Java 的技术问题，可加入 P2P 论坛(p2p. wrox. com)。这个论坛是基于 Web 的系统，读者可就 Wrox 出版的图书以及相关技术发表自己的见解，也可与图书作者和其他读者、技术人员进行交流。此论坛还提供了邮件订阅功能，读者可选择感兴趣的话题，如果论坛上新发布了相关的消息，系统会自动将这些方面的最新信息发送到读者的 E-mail 信箱。

通过 <http://p2p.wrox.com>，不仅在阅读本书时，甚至在开发自己的程序时，读者也能得到许多有益的知识。

只需以下简单操作，读者即可成为论坛的一员：

- 1) 登录 <http://p2p.wrox.com>，并单击“注册”(Register)按钮。
- 2) 阅读用户条例，选择“同意”(Agree)按钮。
- 3) 填写所有必需的个人信息，选填愿意公布的可选信息，然后单击“提交”(Submit)按钮。
- 4) 你的 E-mail 信箱中将收到一封确认函，按照其中的要求确认注册即可。

阅读 p2p 论坛中的内容无须注册，但是发布内容则必须注册。

一旦注册成功，读者即可发表自己的帖子或者对其他发帖人回帖，任意时刻均可登录 p2p 论坛阅读内容。如果想看另外一个论坛的最新消息，可通过单击“订阅”E-mail 中的论坛列表项来添加订阅。

读者若想得到更多 Wrox p2p 论坛的使用信息，请阅读 P2P FAQ，这里提供了如何使用论坛软件以及其他一些常见的问题。单击任何 p2p 页面中的“FAQ”链接即可访问。

目 录

译者序 序 前言

第 1 章 Java 概述	1	2.2 整型数据类型	24
1.1 什么是 Java	1	2.2.1 整型字面值	25
1.2 Java 语言的特性	2	2.2.2 整型变量的声明	26
1.3 学习 Java	2	2.3 浮点数据类型	27
1.3.1 Java 程序	2	2.3.1 浮点字面值	28
1.3.2 学习 Java 要走的路	3	2.3.2 浮点型变量的声明	28
1.4 Java 环境	3	2.4 固定变量的值	28
1.5 基于 Java 语言的面向对象程序设计	9	2.5 算术运算	29
1.5.1 对象是什么	10	2.5.1 整数运算	29
1.5.2 如何定义对象的类	11	2.5.2 整数除法和余数	34
1.5.3 对象的操作	13	2.5.3 自增和自减运算符	35
1.5.4 Java 程序语句	15	2.5.4 短整型数计算	36
1.5.5 封装	15	2.5.5 整数算术运算中的错误	37
1.5.6 类和数据类型	16	2.5.6 浮点运算	37
1.5.7 类和子类	16	2.6 混合数据类型的算术运算表达式	39
1.5.8 使用对象的优点	16	2.6.1 显式类型强制转换	39
1.6 Java 程序结构	16	2.6.2 赋值语句中的自动类型转换	39
1.6.1 Java 的类库	17	2.7 op=运算符	40
1.6.2 Java 应用程序	18	2.8 数学函数和常量	40
1.7 Java 和 Unicode 码	20	2.9 字符的存储	44
1.8 本章小结	20	2.9.1 字符转义序列	44
1.9 参考资料	21	2.9.2 字符的算术运算	45
第 2 章 程序、数据、变量和计算	22	2.10 按位运算	47
2.1 数据和变量	22	2.10.1 使用 AND 和 OR 运算符	48
2.1.1 变量的命名	22	2.10.2 使用异或运算符	51
2.1.2 变量名和 Unicode 码	23	2.10.3 移位运算	52
2.1.3 变量和类型	23	2.10.4 按位运算的方法	56
		2.11 有固定序列整型值的变量(枚举)	58
		2.12 布尔型变量	59
		2.13 运算符的优先级	60
		2.14 程序注释	61
		2.15 本章小结	62

2.16 练习题	63	4.3.7 提取子串	137
第3章 循环与逻辑	64	4.3.8 String 对象的修改版本	141
3.1 决策	64	4.3.9 由 String 对象创建字符数组	141
3.1.1 比较	64	4.3.10 对字符串使用 collection-based for 循环	142
3.1.2 if 语句	65	4.3.11 以字节数组的形式获取字符 串中的字符	142
3.1.3 嵌套的 if 语句	68	4.3.12 由字符数组创建 String 对象	143
3.1.4 比较枚举值	71	4.4 可变字符串	143
3.2 逻辑运算符	72	4.4.1 创建 StringBuffer 对象	144
3.3 条件运算符	76	4.4.2 StringBuffer 对象的容量	144
3.4 switch 语句	77	4.4.3 为 StringBuffer 对象改变字符 串的长度	146
3.5 变量作用域	83	4.4.4 向 StringBuffer 对象增加其他 内容	146
3.6 循环	85	4.4.5 查找子串的位置	149
3.6.1 循环的种类	86	4.4.6 替换缓冲区中的子串	149
3.6.2 嵌套循环	93	4.4.7 插入子串	149
3.6.3 continue 语句	94	4.4.8 从可变字符串中提取字符	150
3.6.4 在循环中使用 break 语句	96	4.4.9 对可变字符串的其他操作	150
3.7 断言	101	4.4.10 由 StringBuffer 对象创建 String 对象	151
3.8 本章小结	103	4.5 本章小结	153
3.9 练习题	103	4.6 练习题	153
第4章 数组和字符串	104	第5章 类的定义	155
4.1 数组	104	5.1 什么是类	155
4.1.1 数组变量	104	5.1.1 类定义中的变量	156
4.1.2 数组定义	105	5.1.2 类定义中的方法	157
4.1.3 数组的长度	106	5.1.3 变量和方法的访问	158
4.1.4 访问数组元素	106	5.2 定义类	158
4.1.5 数组变量的重用	106	5.3 方法的定义	159
4.1.6 数组初始化	107	5.3.1 由方法返回	160
4.1.7 使用数组	109	5.3.2 形式参数表	160
4.1.8 嵌套数组	112	5.3.3 类方法的定义	162
4.1.9 字符数组	117	5.3.4 在方法中访问类的数据成员	163
4.2 字符串	117	5.3.5 this 变量	163
4.2.1 字符串常量	117	5.3.6 初始化数据成员	164
4.2.2 创建 String 对象	118	5.4 构造函数	167
4.2.3 字符串数组	120	5.4.1 默认构造函数	168
4.3 字符串的运算	121	5.4.2 生成类的对象	169
4.3.1 字符串的连接	121	5.5 类的定义和使用	171
4.3.2 字符串的比较	124		
4.3.3 字符串的排序	129		
4.3.4 访问字符串中的字符	131		
4.3.5 搜索字符串中的字符	133		
4.3.6 子串搜索	134		

5.6 方法重载	173	6.9.2 对象的识别	234
5.6.1 多个构造函数	173	6.10 关于枚举	235
5.6.2 使用构造函数复制对象	176	6.11 类的设计	238
5.7 对象的使用	177	6.12 使用 final 修饰符	247
5.8 递归	182	6.13 接口	248
5.9 理解包	184	6.13.1 封装程序中的常量	249
5.9.1 类的打包	184	6.13.2 声明方法的接口	252
5.9.2 将包中的类添加到程序中	187	6.13.3 接口的扩展	254
5.9.3 程序中的包和名称	188	6.13.4 接口的应用	254
5.9.4 导入静态类成员	188	6.13.5 接口类型的方法形参	261
5.9.5 标准包	189	6.13.6 接口定义中的嵌套类	261
5.10 控制对类成员的访问	191	6.13.7 接口与现实世界	262
5.10.1 使用访问属性	192	6.14 匿名类	262
5.10.2 指定访问属性	193	6.15 本章小结	262
5.10.3 选择访问属性	194	6.16 练习题	263
5.11 嵌套类	199	第 7 章 异常	264
5.11.1 静态嵌套类	200	7.1 异常的概念	264
5.11.2 使用非静态嵌套类	204	7.2 异常的类型	265
5.11.3 在顶层类之外使用嵌套类	205	7.2.1 Error 异常	265
5.11.4 局部嵌套类	206	7.2.2 RuntimeException 异常	266
5.12 finalize()方法	206	7.2.3 Exception 类的其他子类	267
5.13 本地方法	207	7.3 异常的处理	267
5.14 本章小结	208	7.3.1 指定方法可以抛出的异常	267
5.15 练习题	208	7.3.2 异常的处理	268
第 6 章 类的扩展与继承	209	7.3.3 try 代码块	268
6.1 使用现有的类	209	7.3.4 catch 代码块	268
6.2 类的继承	210	7.3.5 finally 代码块	273
6.2.1 继承数据成员	211	7.3.6 方法的组织	273
6.2.2 继承方法	212	7.3.7 执行的顺序	274
6.2.3 覆盖基类的方法	215	7.3.8 嵌套的 try 代码块	279
6.3 选择基类的访问属性	216	7.3.9 再次抛出异常	279
6.4 多态	216	7.4 异常对象	280
6.5 多级继承	221	7.4.1 Throwable 类	280
6.6 抽象类	222	7.4.2 标准异常	282
6.7 通用超类	223	7.5 定义自己的异常	282
6.7.1 toString()方法	224	7.5.1 定义异常类	282
6.7.2 确定对象的类型	224	7.5.2 抛出自己的异常	283
6.7.3 对象的复制	225	7.5.3 异常处理策略	283
6.8 接收可变实参的方法	229	7.6 本章小结	287
6.9 对象的类型强制转换	231	7.7 练习题	287
6.9.1 何时强制转换对象的类型	233	第 8 章 理解流	288
		8.1 流和新的 I/O 功能	288

8.2 流的定义	289	10.4.8 使用视图缓冲区	345
8.2.1 输入流和输出流	289	10.4.9 准备好向文件输出的缓冲区	345
8.2.2 二进制流和字符流	290	10.5 写入文件	347
8.3 输入/输出类	291	10.5.1 文件位置	348
8.3.1 基本的输入流操作	291	10.5.2 将变长字符串写入文件	354
8.3.2 基本的输出流操作	294	10.5.3 直接缓冲区和间接缓冲区	359
8.3.3 Readers 流和 Writers 流	294	10.5.4 将数值数据写入文件	359
8.4 标准流	297	10.5.5 将混合数据写入文件	363
8.4.1 从键盘获取数据	298	10.5.6 集中写操作	368
8.4.2 向命令行写	304	10.6 本章小结	371
8.4.3 printf()方法	304	10.7 练习题	372
8.4.4 将数据格式化输入到字符串	308	第 11 章 读文件	373
8.5 本章小结	309	11.1 文件读操作	373
8.6 练习题	309	11.2 文件通道读操作	375
第 9 章 访问文件和目录	310	11.3 读取文本文件	376
9.1 运用 File 对象	310	11.4 读取二进制数据	379
9.1.1 创建 File 对象	310	11.5 读取混合型数据	383
9.1.2 访问系统属性	313	11.6 文件的复制	388
9.1.3 测试和检查 File 对象	315	11.7 随机存取文件	392
9.1.4 文件列表的过滤	320	11.8 使用一个文件通道进行读写	396
9.1.5 创建、修改文件和目录	322	11.9 内存映像文件	398
9.2 创建文件输出流	324	11.9.1 锁定文件	401
9.2.1 确保文件存在	325	11.9.2 文件的部分锁定	402
9.2.2 避免覆盖文件	326	11.9.3 实际文件锁定的注意事项	402
9.2.3 FileDescriptor 对象	328	11.10 本章小结	405
9.3 本章小结	328	11.11 练习题	406
9.4 练习题	329	第 12 章 对象的序列化	407
第 10 章 写文件	330	12.1 在文件中存储对象	407
10.1 文件输入/输出基础	330	12.1.1 将对象写入文件	408
10.2 文件输入/输出	331	12.1.2 将基本数据类型写入对象流	409
10.3 通道	332	12.1.3 实现 Serializable 接口	410
10.3.1 通道操作	332	12.1.4 从文件读取对象	413
10.3.2 文件通道	334	12.1.5 使用对象序列化	417
10.4 缓冲区	334	12.1.6 自己实现类的序列化	420
10.4.1 缓冲区容量	335	12.1.7 序列化的问题和复杂情况	420
10.4.2 缓冲区位置和边界	336	12.2 本章小结	423
10.4.3 设置位置和边界	337	12.3 练习题	424
10.4.4 创建缓冲区	338	第 13 章 泛型类	425
10.4.5 标记缓冲区	343	13.1 什么是泛型	425
10.4.6 缓冲区数据传送	343	13.2 泛型类的定义	426
10.4.7 将数据传送到缓冲区	344	13.2.1 泛型的实现	427

13.2.2 泛型的实例化	428	14.7.3 创建 HashMap 容器	498
13.2.3 泛型实例的运行时类型	433	14.7.4 存储、获取以及删除对象	499
13.2.4 泛型实例之间的关系	434	14.7.5 处理映射中所有的元素	500
13.2.5 多重类型参数	434	14.8 本章小结	509
13.2.6 类型参数的作用域	435	14.9 练习题	510
13.2.7 泛型中的静态成员域	435	第 15 章 实用类的集合	511
13.2.8 类型参数限制	436	15.1 数组的工具方法	511
13.3 泛型和泛型接口	439	15.1.1 填充数组	511
13.3.1 利用 collection-based for 循环	439	15.1.2 比较数组	512
13.3.2 实现一个迭代器	441	15.1.3 对数组排序	513
13.3.3 参数化的二叉树	443	15.1.4 搜索数组	516
13.4 原生类型变量	451	15.2 Observable 对象和 Observer 对象	519
13.5 用通配符作为类型实参	452	15.2.1 定义可观察对象的类	520
13.5.1 对通配符的约束	454	15.2.2 Observable 类的方法	520
13.5.2 进一步研究 Class 类	457	15.3 产生随机数	523
13.6 数组和参数化类型	458	15.4 日期和时间	525
13.7 参数化方法	460	15.4.1 Date 类	526
13.8 参数化类型和继承	466	15.4.2 解释 Date 对象	526
13.9 本章小结	467	15.4.3 Gregorian 日历	530
13.10 练习题	467	15.5 正则表达式	535
第 14 章 集合框架	468	15.6 使用扫描器	552
14.1 理解集合框架	468	15.6.1 创建扫描器对象	552
14.2 对象的集合	469	15.6.2 获取从扫描器的输入	553
14.2.1 集	469	15.6.3 测试标记	555
14.2.2 序列	469	15.6.4 为标记定义自己的模式	556
14.2.3 映射	470	15.7 本章小结	558
14.3 迭代器	471	15.8 练习题	558
14.4 集合类	474	第 16 章 线程	559
14.5 使用向量	477	16.1 理解线程	559
14.5.1 创建向量	477	16.1.1 创建线程	561
14.5.2 把对象存入向量中	480	16.1.2 停止线程	565
14.5.3 从向量中获取对象	481	16.1.3 连接线程	567
14.5.4 从数组中删除对象	483	16.1.4 线程调度	567
14.5.5 搜索向量	484	16.1.5 实现 Runnable 接口	567
14.5.6 应用向量	485	16.2 管理线程	570
14.5.7 为集合排序	488	16.2.1 同步处理	570
14.5.8 栈式存储	490	16.2.2 死锁	585
14.6 使用链表	494	16.2.3 线程间的通信	586
14.7 使用映射	496	16.3 线程的优先级	590
14.7.1 散列处理	496	16.4 本章小结	593
14.7.2 用自己的类创建的对象作键	497	16.5 练习题	593

第 17 章 创建窗口	594	18.6 应用程序中的语义事件监听器	684
17.1 Java 中的图形用户界面	594	18.7 使用 Action	689
17.2 创建窗口	596	18.7.1 Action 接口	690
17.3 组件和容器	599	18.7.2 将行为应用于菜单项	691
17.3.1 窗口和框架组件	601	18.8 添加工具栏	696
17.3.2 窗口窗格	601	18.8.1 添加按钮到工具栏中	696
17.4 组件基础	602	18.8.2 添加工具提示	702
17.4.1 组件属性	602	18.8.3 禁用动作	704
17.4.2 组件的大小和位置	603	18.9 本章小结	705
17.4.3 点和矩形	606	18.10 练习题	705
17.4.4 组件的可视特性	608	第 19 章 在窗口中绘画	706
17.4.5 Swing 组件	615	19.1 使用模型/视图体系结构	706
17.5 使用容器	617	19.2 组件中的坐标系	709
17.6 容器布局管理器	619	19.3 在组件上绘图	710
17.6.1 流布局管理器	620	19.3.1 图形环境	711
17.6.2 使用边界布局管理器	624	19.3.2 绘制过程	713
17.6.3 使用卡片布局管理器	626	19.3.3 渲染操作	714
17.6.4 使用网格布局管理器	627	19.4 几何图形	714
17.6.5 使用盒式布局管理器	629	19.4.1 定义点的类	714
17.6.6 使用 GridBagLayout 管理器	634	19.4.2 直线段和矩形	716
17.6.7 使用 SpringLayout 管理器	640	19.4.3 圆弧和椭圆	720
17.7 添加菜单到窗口中	646	19.4.4 曲线	723
17.7.1 创建 JMenu 和 JMenuItem	646	19.4.5 复杂路径	729
17.7.2 创建菜单	647	19.5 填充几何图形	734
17.7.3 添加菜单项到下拉菜单中	649	19.6 管理几何图形	738
17.7.4 为菜单项添加快捷键	652	19.6.1 在文档中存储图形	739
17.8 更多关于 applet 小程序的内容	654	19.6.2 绘制图形	740
17.9 本章小结	656	19.7 使用鼠标绘制	741
17.10 练习题	657	19.7.1 处理鼠标事件	742
第 18 章 事件处理	658	19.7.2 处理鼠标按键按下事件	743
18.1 基于窗口的 Java 程序	658	19.7.3 处理鼠标拖拽事件	745
18.2 事件处理过程	659	19.7.4 处理鼠标按键释放事件	747
18.3 在图形用户界面代码中避免死锁	661	19.7.5 使用 MouseInfo 类方法确定 鼠标的位置	748
18.4 事件类	662	19.8 定义自己的图形类	748
18.4.1 低级事件类	662	19.8.1 定义线段	748
18.4.2 低级事件监听器	667	19.8.2 定义矩形	750
18.4.3 语义事件	672	19.8.3 定义圆形	752
18.4.4 语义事件监听器	673	19.8.4 绘制曲线	754
18.5 applet 小程序中的语义事件处理	674	19.9 本章小结	756
18.5.1 其他处理事件方法	682	19.10 练习题	757
18.5.2 处理低级事件和语义事件	683		

第 20 章 扩展图形用户界面	758	第 22 章 Java 与 XML	872
20.1 创建状态栏	758	22.1 XML	872
20.2 使用对话框	762	22.2 XML 文档结构	873
20.2.1 模式与非模式对话框	762	22.2.1 有效的 XML 文档	874
20.2.2 简单的模式对话框	763	22.2.2 XML 文档中的元素	875
20.2.3 即时对话框	767	22.2.3 元素属性	878
20.2.4 输入对话框	768	22.3 XML 中的数据结构	880
20.2.5 字体选择对话框	777	22.4 文档定义类型	881
20.3 弹出式菜单	790	22.4.1 声明 DTD	881
20.3.1 显示弹出式菜单	791	22.4.2 定义 DTD	882
20.3.2 实现上下文菜单	793	22.4.3 Sketcher 程序的 DTD	888
20.4 转换用户坐标系	800	22.5 构造良好文档的规则	891
20.4.1 仿射转换类	802	22.6 XML 名字空间	891
20.4.2 修改图形环境的转换	803	22.6.1 名字空间声明	892
20.4.3 创建仿射转换对象	804	22.6.2 XML 名字空间和 DTD	893
20.5 选择自定义颜色	816	22.7 XML 模式	893
20.6 本章小结	818	22.7.1 定义模式	894
20.7 练习题	818	22.7.2 定义元素	895
第 21 章 文档的整理和打印	819	22.7.3 定义复杂元素的属性	896
21.1 序列化处理草图	819	22.7.4 指定一组备选元素	898
21.2 实现可序列化接口	822	22.7.5 用于 Sketcher 程序的模式	898
21.3 支持文件菜单	827	22.7.6 使用模式的文档	903
21.3.1 使用文件选择器	827	22.8 使用 XML 文档编程	904
21.3.2 文件保存操作	828	22.8.1 SAX 处理	904
21.3.3 文件另存为操作	834	22.8.2 DOM 处理	905
21.3.4 文件打开操作	835	22.9 访问解析器	906
21.3.5 开始新建草图	837	22.10 使用 SAX	906
21.3.6 防止关闭时数据丢失	838	22.10.1 使用不同的解析器	909
21.4 在 Java 中打印	839	22.10.2 解析器的特征及特性	909
21.4.1 创建和使用 PrinterJob 对象	841	22.10.3 使用 SAX 解析文档	911
21.4.2 打印页面	843	22.10.4 实现 SAX 处理器	912
21.4.3 打印完整的草图	847	22.10.5 处理其他解析事件	920
21.4.4 横向打印	851	22.10.6 解析由模式实例化的文档	921
21.4.5 改进打印应用程序	853	22.11 本章小结	925
21.4.6 实现页面设置	854	22.12 练习题	925
21.4.7 使用 Java 的打印对话框	857	第 23 章 创建和修改 XML 文档	926
21.4.8 多页文档打印	859	23.1 文档对象模型	926
21.4.9 使用 Book 打印	866	23.1.1 设置 DOM 解析器特征	927
21.4.10 打印 Swing 组件	868	23.1.2 解析文档	928
21.5 本章小结	871	23.1.3 遍历文档对象树	929
21.6 练习题	871	23.1.4 创建 XML 文档	939