

世界著名计算机教材精选

PEARSON
Prentice
Hall

Java 面向对象 程序设计 (第2版)

David M. Arnow 著
Scott Dexter
Gerald Weiss

郑莉 张长富 等 译



INTRODUCTION TO PROGRAMMING USING JAVA

Second Edition



清华大学出版社

世界著名计算机教材精选

Java 面向对象程序设计

(第 2 版)

[美] David M. Arnow 著

Scott Dexter

Gerald Weiss

郑莉 张长富 等译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2006 by PEARSON EDUCATION ASIA LIMITED and TSING-HUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Introduction to Programming Using Java: An Object-Oriented Approach, Second Edition by David M. Arnow, Scott Dexter, Gerald Weiss, Copyright © 2004

EISBN: 0-321-20006-3

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall PTR.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education(培生教育出版集团)授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字 01-2004-6476 号

版权所有,翻印必究。举报电话: 010-62782989 13501256678 13801310933

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。

图书在版编目(CIP)数据

Java 面向对象程序设计(第2版)/(美)阿诺(Arnow, D. M.), (美)德克斯特(Dexter, S.), (美)韦斯(Weiss, G.)著; 郑莉, 张长富等译. —北京: 清华大学出版社, 2006. 9

(世界著名计算机教材精选)

书名原文: Introduction to Programming Using Java, 2e

ISBN 7-302-13510-X

I. J… II. ①阿… ②德… ③韦… ④郑… ⑤张… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 085022 号

出版者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-62770175

地址: 北京清华大学学研大厦

邮编: 100084

客户服务: 010-62776969

印刷者: 清华大学印刷厂

装订者: 北京市密云县京文制本装订厂

发行者: 新华书店总店北京发行所

开本: 185×260 印张: 37 字数: 921 千字

版次: 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书号: ISBN 7-302-13510-X/TP. 8478

印数: 1~3000

定价: 69.00 元

前 言

本书旨在作为编程入门课程的基本教材。它不需要任何编程背景。本书涵盖的材料足以适应一或两个学期,其后讲述传统的 CS2(数据结构)课程的情况。

给学生的说明

本书是应用 Java 介绍计算机编程技巧的入门图书。之所以选用这个流行语言,是出于几个理由:

- Java 是一个面向对象的语言。面向对象已经成为软件开发领域的一种基本方法。在本教材中,我们将解释什么东西构成了语言的面向对象。
- Java 是一个相对简单的面向对象语言,至少与其他一些面向对象语言(比如 C++) 相比是这样。尽管 C++ 更多复杂性的介绍超越了本书的范围,但还是要指出,它存在一些让刚刚开始学习的学生误入歧途的缺陷。其中的很多陷阱都不会在 Java 中出现。
- Java 从其他流行语言中借用了许多特性,特别是从 C 和 C++ 中。这些熟悉的特性也使得 Java 对那些语言的用户产生了吸引力。
- Java 甚至让初级程序员也能够编写出包含相当复杂用户界面——按钮、列表框、滚动条等——的程序。
- Java 能够运行在多种类型的机器上——PC 机、苹果机、Sun 工作站等。
- Java 提供了一些相当复杂的工具,包括相对容易地访问网络和互联网,这使得它对许多编程领域都有吸引力。
- 使用 Java 编程有乐趣。正像前面已经提到的那样,即使是一个经验较少的新手也能够利用 Java 提供的工具编写出外观漂亮、行为复杂的程序。

尽管有那么多好处和乐趣,尽管你将逐步学习使用 Java,但我们还有一个十分特别的目的:让你开始像程序员那样思考。这就是说,学习分析问题,将问题划分为部件块,并设计一个解决方案。它还意味着大量的练习。编程并不能够简单地从书本上学习——必须编写大量的计算机代码。在学完本书的时候你不会成为一个专家,但如果你留心并做了各种编程练习后,你就踏上了征途。

给老师的说明

在这本教材中,我们使用 Java 编程语言向学生介绍编程。我们的核心焦点集中在对

问题开发软件解决方案的过程上。这一过程不能够抽象得到,而需要描述大量的 Java 语言以及它的一些类库,并且讨论一些编程技巧和算法。本前言详细说明我们来如何达到这些目标。

在本书的 Web 站点 www.aw.com/cssupport 上可以找到 FAQ(常问问题)部分。它以问与答的形式涵盖了众多主题。

对第 2 版的修改

在第 2 版中,我们已经保留和加强了尽早介绍对象方法的承诺。特别地,前面的章节已经进行了大量重写和扩展。介绍使用对象的第 2 章和第 3 章加入了基本的面向对象概念的额外讨论和演示;我们也强调和探索了将阅读程序作为建模行为一部分的好处。我们已经拓展了类定义过程的处理方式(第 4 章和第 5 章),以便使用更多的示例提供更精细的介绍。类设计现在有了自己的独立一章(第 7 章),它也拥有了更多、更丰富的示例。这些新的示例部分由前面介绍的一些必要的编程概念所支持(比如基本类型、赋值以及条件等),其他必要概念(高级条件、更多的基本类型以及简单计数循环)在第 6 章中介绍。集合遍历的讨论已经使用计数循环和下标进行了重写;使用 Enumeration 接口作为替代方法的简短解释放在了附录 D 中。以这种方式,我们使这个主题的讨论与循环模式、迭代、Vector 集合的内容更加贴切了。后面的章节我们也进行了一些调整,最大的变化是删除了示例一章,该章中的部分示例现在分散在了剩余的章节中。

我们已经重写了大部分 GUI 补充材料,以便补充材料与章节正文介绍的概念联系得更密切。

最后,我们已经把所有节里的练习移动到了相应章的末尾,并增加了大量新练习。

范式

对任何编程语言的介绍都必须解决范例选择问题。我们介绍的编程平台是 Java,因此,我们选择面向对象的编程也就不令人吃惊了。然而,尽管在 CS1 课程中过程和函数编程介绍什么内容是相当清楚的,但在本级课程中,到底介绍 OOP 的什么内容则存在着各种不同的竞争版本。我们将重点集中在:

- 定义和使用类
- 行为和响应问题
- 使用复合而不是继承

通过识别问题中的主要对象,描述其行为,之后定义提供其行为的类的方式解决本文中的典型问题。在求解过程中通常会定义少数几个(经常是一个)独立的辅助类。通过在 main 方法中编写应用主要对象的短小代码来求解问题。

20 世纪 90 年代早期,在 OOP、过程编程、命令编程之间存在着一些混乱。现在,人们已经很好地理解了 OOP 与过程编程属于不同的编程范式,命令编程则在两者之间都借用了一部分。毫无疑问,一旦使用了赋值语句,那就是在做命令编程,而且,改变对象状态

的发送消息也可以看作是命令编程。因此,本教程将从头讲述 OOP 和命令编程。但是,就像在刚开始介绍过程编程的方法(在介绍过程的 CS1 课程中流行了很长时间)中在介绍诸如条件和循环这样的命令式设施之前首先介绍过程调用机制和用法那样,这里我们在介绍条件语句之前首先介绍消息发送和对象创建的机制和用法,在介绍迭代之前首先介绍类的定义。这两种情况下的道理是一样的,首先介绍范式和开发相应环境下的命令式设施更受重视一些。

在过程编程中,完成一件任务的方法是,找到一个能够完成这个过程,然后调用这个过程。如果不存在这样的过程,那么程序员就不得不编写一个这样的过程。在 OOP 中,完成一件任务的方法是,找到或创建一个其行为包含了完成任务并向对象发送消息的类的对象。如果这样的类不存在,那么程序员就必须创建一个这样的类。

过程

我们的主要精力集中在针对问题开发软件方案的过程上。为了达到这个目标,我们在 4 个方面介绍了非正式的,但是系统的方法:

- 从问题陈述中开发类规范
- 对给定的类规范实现类
- 构造循环
- 构造递归方法

前两个方法在第 4 章、第 5 章中以基本方式介绍,并在第 7 章中充实,然后在本书随后章节中一致地应用。后两个方法与迭代和递归一起介绍,它们分别在第 10 章和第 14 章中阐述。同样地,在随后章节中出现迭代或递归的地方,也一致地应用这两种方法。

这些系统方法一致地反复使用是必要的,目的是让学生意识到,这些方法并不仅仅是好听的甜言蜜语,而是可以在开发问题解决方案过程中真实可用的思想。

对过程的强调意味着引论性教材中常用的两种做法在本书中几乎看不到,这两种方法是代码的剖析和代码的增量修改(尽管后一种方式确实在本书的一些拓展示例和练习中出现过)。代码剖析要求一开始时就给出完整类实现的展示,而不给出开发过程。接着仔细地分析代码。这种方法在解释代码工作方式时是有帮助的,但它并没有解释开发代码的过程。

教材中的许多主题以及它们的介绍顺序都由介绍过程这个想法所确定。例如,在提供类规范和定义的方法之前,学生必须首先相当熟悉类是放置在一起的一组行为的概念,也要熟悉类的组合使用。为了达到这个目的,第 3 章讨论了一些 Java 预定义类(包括 BigInteger、Date、GregorianCalendar 以及一些相关的 I/O 类)。

语言

随着详细阐述程序开发的过程,我们介绍了 Java 语言的特性。为了防止这些特性的细节讨论离开发代码的过程的讲述太远,我们经常会把这些讨论推迟到称之为“Java 小

插曲”的特殊小节中讲述。在这些小节中,我们给出了代码开发课程中引入特性的细节。我们还使用这些小节来介绍其他地方没有出现,但 CS1 的学生应该熟悉的语言特性。尽管如此,Java 语言的一些特性还是没有被覆盖到,比如位运算、并发、同步以及内部类。

GUI 编程

Java 对图形用户界面的支持也是它出现在 CS 教学中的原因之一——这既是对指导老师也是对学生讲的。我们把这个主题的讨论放在了一系列的 GUI 补充材料中,也就是说,出现在每章末尾的一个特殊节中。每一章的主要内容完全独立于这些补充材料。每一个补充材料都介绍了一组新的 GUI 工具/技巧,目的是对相应章中介绍的主要内容提供补充。

这种组织方式的优点是允许指导老师完全跳过 GUI 编程的介绍,或者在任何时刻都可以介绍它。并且,这种组织方式也进一步强化了面向对象编程这个课文主题的核心地位,而不是把精力放在语言专用特色上。进一步地,它也把课文的主题从类库的变化中隔离出来,主要是从类库的抽象窗口工具箱(AWT, Abstract Windows Toolkit)中隔离出来。

出于下面的几个理由,我们专门选择使用小应用程序(applet)而不是 GUI 补充材料中的应用程序:

- 可以很容易把小应用程序转换为应用程序;反过来的转换则更加困难,而且在某些情况下转换是不可能的。
- 小应用程序的执行环境更加复杂,因此课文中更值得讨论。
- 学生喜欢创建“酷毙”的 Web 页面,并在这些页面中显示它们的小应用程序。
- 广泛地涵盖 AWT 超出了本教材的范围。因此,我们的方法是回应下述几个关键问题:
- 小应用程序基础(第 1 章和第 2 章)
- 布局——部件的放置(第 4 章)
- 事件处理(第 6 章和第 8 章)
- 文本和图形中的精度(第 5 章、第 11 章和第 12 章)
- 线程(第 11 章和第 14 章)

沿着这样的途径,将会遇到众多有用的 AWT 类和方法。

类似地,我们专门选择了使用 AWT 而没有选择 Swing(尽管在最后的 GUI 补充材料中我们给出了 Swing 的简短介绍)。我们的理由包括下述几个方面:

- 与从 JDK 1.0 的事件模型到 AWT 的事件模型(JDK 1.1)的变化不同,在 AWT 和 Swing 之间的差别主要在审美标准上,至少在初学者这个层次是这样的。
- Swing 引入了一些对于初学者来说并不必要的复杂性(例如,组件放置中内容池的引入)。
- 只有引入了继承、接口、多态后更为有趣的改进才变成可理解的内容(例如,所有 JComponents 都继承自 AWT Container 这个事实让我们可以把一个 JComponents

(比如图标)替换为另一个 JComponents(比如按钮))。

灵活性

每一个计算机科学系都有自己的文化和 CS1 课程的目标。即使是在 CS1 课程 OOP 中完全分享我们方法的指导老师或许还会想要重新组织,甚至遗漏本教材中的某些主体。因此,我们已经尽最大努力来使多个主题的介绍相互独立。与此同时,我们也确实希望本教材中的主题相互之间建立联系。因此,值得介绍一下解决这些冲突的几个途径。

继承

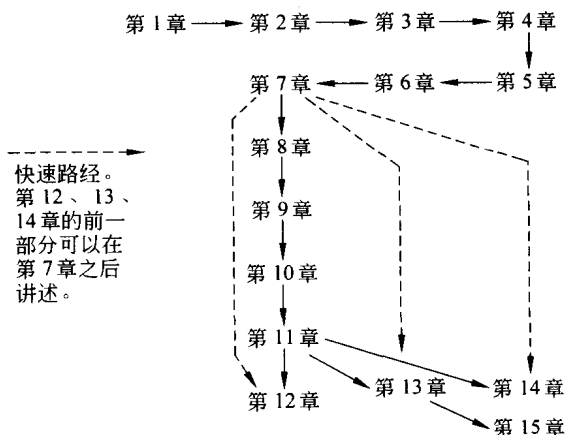
继承是第 12 章的课程。但是,希望在本课程更早一些时候介绍这个主题的指导老师可以直接从第 7 章转到第 12 章,并讲述该章的前三分之二的內容。该章剩余的三分之一内容阐述多态和接口,它使用了第 10 章到第 11 章的材料。

递归

递归是第 14 章的课程。但是,希望在本课程更早一些时候介绍递归的指导老师可以直接从第 7 章转到第 14 章,并讲述该章的前三分之一的内容,这里没有包括数组和向量的内容。另一方面,希望不讲递归主题的指导老师也能够这样做。

异常

异常是第 13 章的课程。但是,该章的主要内容可以在讲述第 7 章之后直接讲述。当然,GUI 补充材料为指导老师在图形和事件编程方面提供了极大的灵活性。依赖图概要展示了这些关系。



输入/输出

我们认为,在学生着手编写类定义(从第4章开始)之前,让学生得到充足的类、对象、构造器、复合、级联以及作为模型的类概念方面的完整经验是十分重要的。第3章利用各种示例类提供了这些关键主题的丰富和复杂的讨论,其中包括 java.io 包中的一些类。

第3章是一系列展示的集大成者,但为了安全起见,对于用别的方法为学生准备第4章内容,从而希望跳过第3章的指导老师来说,我们在附录C中提供了一个十分简单的 I/O 包 AWIO,它可以用于代替 3.9~3.12 节的内容。

字样

现在,给出代码示例中出现的四种元素的字样十分有用。

首先是代码本身:

```
Class PrefaceExample {  
    public static void main(String[] arg) {  
        System.out.println("just an example");  
    }  
}
```

我们经常在代码中添加注释,使用以//开始的记号或者使用/*和*/。这些都是实际作为代码一部分的记号。下述代码的第二行就是一个注释示例:

```
Class PrefaceExample {  
    //在显示上输出一个简短的示例性说明  
    public static void main(String[] arg) {  
        System.out.println("just an example");  
    }  
}
```

在我们开发计算机代码的过程中,经常会用一些占据位置的标记,表示还要编写的代码。下面代码中的第五行就是这样的占位符(伪代码)的一个示例:

```
Class PrefaceExample {  
    //在显示上输出一个简短的示例性说明  
    public static void main(String[] arg) {  
        System.out.println("just an example");  
        更多的输出语句放在这里  
    }  
}
```

最后,通常不是代码组成部分但可以帮助说明用法的解释性标记放在代码周围的阴影框中,通常带有箭头指示:

```
Class PrefaceExample {
```

```
    //在显示上输出一个简短的示例性说明
```

```
    public static void main(String[] arg) {
```

```
        System.out.println("just an example");
```

```
        更多的输出语句放在这里
```

```
    }
```

```
}
```

注释

一个带有 main 方法的类

各章概要——非 GUI 部分

第 1 章: 进入 Java 这里我们引入了编程作为创建情景模型的概念。我们的主要目标是向读者介绍类、对象的概念以及它们在 Java 中实现的消息传递机制。此外,我们还展示和解释了一个程序示例,并讨论了编写和运行 Java 程序的机制。

第 2 章: 发送消息和执行功能 在这一章中,焦点集中在通过向对象发送消息使用对象以及相关的所有细节的机制上。我们详细说明了第 1 章中介绍的三个关键概念——类、对象和消息传递——并介绍了其他 OOP 概念和必要的基础知识:方法、参数、返回值、签名、原型、重载、引用变量、声明、赋值、int 类型和算术运算以及简单条件。String 和 PrintStream 类用于演示这些概念。

第 3 章: 使用对象和基本类型 这一章在开始处展示了如何使用类的构造器创建对象。我们继续强调类是行为存储池这个主题。这个观念通过考察一些 Java 预定义类的方式得到加强;我们也探索了如何使用级联和复合的方式将行为结合起来。与此同时,我们进一步引入了一些必需的编程概念(例如条件和 boolean 值),以便增加示例的丰富性。

第 4 章: 定义类 此时,学生已经相当清楚 OOP 的首要原则:如果有什么事情需要完成的话,那么就找到一个能够完成它的对象并向它发送一条消息。现在我们向读者展示如何定义新类。这个定义要求一些机制方面的方法,我们在这一章开始介绍这些内容:类定义结构、方法定义结构、声明、范围、参数的使用、局部变量、实例变量以及返回语句。在所有这些必要的语言细节中,我们试图将焦点保持在行为、接口和状态的概念上。按照这个思路,我们在类定义上引入了限制版本的系统方法,该类定义将在下一章中介绍。

尽管仅仅说明了方法的初步特点,但是,一旦学完了本课程的这一章,我们就会把悬着的心放下来。此时,我们能够给出涉及新类定义中的任务了。现在就可以进行 OOP 开发了。

第 5 章: 高级类定义 在这一章中我们讲述了类定义的更多机制:构造器、静态方法、final 值以及关键字 this。

第 6 章: 方法内部: 命令式程序设计 现在我们已经讨论了类定义的主要课题,要转到编写强大方法的主题上了。我们继续给出类定义的进一步的示例,但是在这里,我们的焦点集中在命令式编程概念上(基本数据类型、广泛选择的条件语句以及计数循环的介绍),我们使用这些概念来提供方法所需要的行为。

第 7 章: 类设计 此刻,我们介绍余下课文中将持续使用的、类定义的系统方法。这

个方法通过识别问题陈述中的名词开始。以此为起点,标识出主要对象。之后实现该对象的类,同时实现问题中相关的附加类。类的实现包括定义其行为和状态。这种直接的“瀑布”方法对于 CS1 学生遇到的绝大多数设计问题都是足够使用的。

第8章:验证对象的行为 本章是测试需求和技巧的概要介绍。它引入了测试驱动和模块测试的概念。除此之外,它为读者提供了测试案例选择和构造的起点。

第9章:多个对象的处理 这里我们介绍了迭代和集合记号的额外限制形式。引入了 while 循环并解释了它的工作机制,但是,进一步的讨论放在了下一章中,其用途局限于读取/处理循环模式中:

```
读取
While (noe eof) {
    处理
    读取
}
```

我们也介绍了新的 for 循环模式:

```
for (i = 0; i < 集合元素数量; i++)
    处理第 i 个元素
```

我们使用的集合是 Vector 类。它提供了几个优点:

- 索引的复杂性被推迟到学生能够处理多个对象的时候介绍。
 - 不要求新的语法。向量仅仅是另一个对象,可以通过向它发送消息来进行管理。
- 在本章的小结中我们介绍了数组,并演示了数组和向量之间的一些相似点和不同点。尽管存在一些限制,但是,当我们学完本课程的这一章之后,可以长出第二口气了。

至此,我们可以给出更值得回忆的“真实”应用程序了,它与实用类存在着差异。

第10章:设计迭代 这是深入介绍最重要的 CS1 命令式编程课题的一章:循环的构造。一项 CS1 教材和课程的调查揭示了三种方法:

- 没有方法——仅仅模仿了图书中的代码;
- 提供了一组循环模式;
- 某种类型的系统开发技巧,通常是打了折扣的形式化方法。

本章中,我们结合了后两种方法。我们提供了一个系统化的技巧,并把它应用到了几个典型问题上,识别出了我们后面能够引用的循环模式作为结果。

第11章:维护对象集合 本章的焦点是搜索和排序算法。我们已经超越了通常仅仅介绍集合对象和数组的做法,而是考虑了排序外部文件。在此,我们利用此机会介绍了 Java 线程。

第12章:扩展类的行为 我们介绍继承的方法是强调通过子类扩展超类的行为。这是一种最常见的继承使用方法,特别是刚刚开始学习的程序员——它利用现有类,并通过增加状态和行为来产生一个功能更强的类。我们在第3到第7章介绍的类就采用了这种方法,包括预定义类和程序员定义的类,其理由如下:

- 学生已经熟悉扩展的类。

- 如果需要的话,指导老师可以在本课程的更早一些时候介绍继承,例如,在第 7 章之后马上介绍。

状态的扩展、保护型实例变量、覆盖、多态都可以在这个环境中很容易地引导出来。

讲述 GUI 补充材料的老师可能希望至少介绍一下第 12 章的开始部分,以便尽早让学生体验到继承是如何让即使初学的人也能够实现复杂窗口应用程序的感觉。

我们还提供了继承的另一种用法的说明:从几个逻辑相关的类中提取出公共的行为/状态,构造出一个超类和类层次。我们再次强调了行为的概念以及层次模型中各层建立的模型与对象抽象的不同。

最后,给出了接口,它是强制类遵从指定协议的一种手段。

第 13 章:异常 贯穿于教材的另一个主题是责任驱动的编程。类应该尽可能地对其行为负责。问题的另一面是类不应该对逻辑上不属于它的行为负责。我们以这种方式提供异常处理——方法向方法的调用者发出异常信号,它不一定表示出错。尽管完全理解异常的层次结构需要一些必要的继承知识,但本章的前面部分可以在早些时候讲述(在第 7 章之后),以便清楚地阐明许多方法签名中出现的抛出子句。

第 14 章:递归 在递归的处理中,我们把焦点集中在它作为一种编程工具上。我们首先介绍了一些十分简单的问题,某些人可能会认为它对于介绍递归可能不那么容易。这些问题提供了开发一种构造递归解法方法的环境。之后,我们采取了一种必需的迂回路线,并讨论了如何实现递归,但我们慎重地提醒学生,不要把精力集中在如何使用递归上,而忽略了实现问题,以此结束讨论。然后,我们转到了复杂形式和递归的两个问题上——产生排列和传统的 Hanoi 塔问题——并以递归和迭代的比较结束本章。

当学完了本课程的这一章后,我们长喘了第三口气。至此,我们已经讲述了赋值、变量、表达式、数值类型、逻辑类型与字符串类型、交互和文件输入/输出、控制结构、函数(方法)、结构(类)、数组、几个算法、递归、测试与调试——所有传统的 CS1 内容。而且,无疑我们已经做了更多:类、对象、消息,还有已经包含了 GUI 补充材料的内容。在本课程最终成绩交付给教导主任时就会听到他的赞叹了。

第 15 章:客户-服务器计算 在这一章中,我们提供了使用 Java 进行网络编程的扼要介绍。我们展示了 Internet 的一些基础概念(特别地,我们描述了 TCP 连接的本质)以及使用这些概念来支持简单 HTTP 和 SMTP 客户端的开发。

附录 附录包括:

- 一份如何在 UNIX/Linux、Windows 或 MacOS X 环境下编写和运行 Java 程序的描述。
- 一组在可以使用 Java 预定义输入/输出类的场合使用的、支持输入和输出的替代类。
- 一个使用 Enumeration 接口遍历集合的简短讨论。
- 补充材料。

本书的所有读者都可以在 www.aw.com/cssupport 处得到下述补充材料:

- 源代码:出现在教材中的所有完整的类和方法。
- JavaPlace 访问。

- 错误:我们已经尽最大努力来避免错误并希望错误尽可能的少。这里你可以链接到已经发现错误的列表上。

下面的指导老师补充材料只对具备资格的指导老师有效。请与你本地的 Addison-wesley 销售代表联系,或者向 aw.csc@aw.com 发送电子邮件,以便了解和访问这些补充材料的信息。

- Instructor's Manual with Solutions;这里包含了讲授建议、示例提纲、适合于家庭作业的额外提问和问题。也包含了所有练习的完整解答。
- 按章排列的测试库,提供了 Word 文件和 Test Gen 两种格式。
- 所有图示的 PowerPoint 幻灯片。

联系我们

我们欢迎提问、评注、建议以及校正。我们的电子邮件地址是:

arnow@turingscraft.com

sdexter@brooklyn.cuny.edu

weiss@turingscraft.com

致谢

正像阅读前言致谢节的每一个人都知道的那样,教材真正是众多人们协作和支持的结果。我们从 Addison-Wesley 得到的支持是最大的。

感谢市场部的 Nathan Schultz、Lesly Hershman、Katherine Kwack;出品部的 Patty Mahtati、Jeffrey;设计部的 Joyce Cosentino Wells;Argosy 出版部的 Daniel Rausch、Edalin Michael。

我们特别感谢 Galia Shokry 和 Maite Suarez-Rivas 的耐心和鼓励。

众多的审稿人也做出了重要的贡献,他们是:南加州大学的 Michael Crowley;萨斯喀彻温大学的 Ralph Deters;俄克拉荷马大学的 Le Gruenwald;克莱姆森大学的 David P. Jacobs;加州大学的 Chung Lee;西伊利诺大学的 Mike Litman;关柏林州立大学的 Yenumula B. Reddy;罗彻斯特大学的 Nan Schaller;新墨西哥州立大学的 Esther Sterner;加州大学戴维斯校区的 Shih-Ho Wang。从第一版开始我们对下述审稿人表示感谢:加州大学达特茅斯学院的 Jan Bergandy;圣塔芭芭拉城市学院的 Robert H. Dependahl, Jr;圣路易华盛顿大学的 Eileen Kraemer;宾夕法尼亚州立大学宾州伊利分校的 Ronald L. McCarty;威斯康星大学拉克洛斯分校的 David D. Riley;卡内基梅隆大学的 Jim Robert;科尔比学院的 Dale Skrien;爱荷华大学的 Ken Slonneger。这些审稿人提出了许多有价值的建议,促使我们进一步精化和不时地重新思考我们的方法。

我们在 CS1 中使用 Java 的方法随着 CS 教育社团的成熟而成熟。本书中仅仅有极少的想法属于我们自己,但是,我们希望能够很好地表现我们的同行教育家们的优秀设想。

对作者家庭的感谢按惯例放在末尾。我们不打算违反这个传统,但对我们家庭成员的支持、鼓励、包容和爱的感谢并不是最后才有,而是从一开始到中间,直到最后一直都充满着,原因在于他们一直伴随着我们,自始至终,每一个阶段。在本教程的编写过程中,我们的家庭容忍了缺席、熬夜、起早、迷恋于“GUI 补充材料”的抱怨、未洗的盘子、推迟的晚餐、众多的外卖、家庭计算机的独占,这还不包括多次的气馁和烦恼。尽管存在这么多的不便,但他们为我们奉献了所有的爱,支持我们的梦想,我们永远也不会忘记。谢谢你们:

Barbara

Jeanne

Fern

Kera

David, Sharon, Kathy

Yocheved

Alena

The Theoharis clan

Zvi

Joanna

Shlomo

David M. Arnow**Scott Dexter****Gerald Weiss**

目 录

第 1 章 进入 Java	1	应用程序概述	16
1.1 计算机与程序	1	第 2 章 发送消息和执行操作	22
1.2 程序与模型	1	2.1 简介	22
1.3 对象、类和消息	3	2.2 使用 PrintStream 对象	22
1.3.1 对象	3	2.3 Java 小插曲 引用、方法和消息	23
1.3.2 行为	3	2.3.1 方法的重载	24
1.3.3 信息	3	2.3.2 消息发出后	24
1.3.4 Java 程序	4	2.3.3 再论标识符	24
1.3.5 类	4	2.4 String 类	24
1.3.6 预定义的对象和类	4	2.5 方法、参数以及返回值	27
1.4 第一个对象 PrintStream 对象	5	2.6 引用变量	27
1.5 发送一个消息到 System.out 对象	5	2.7 Java 小插曲 变量和赋值	29
1.6 一个 Java 程序	7	2.7.1 变量	29
1.7 Java 小插曲 标识符、语句顺序、格 式和注释	8	2.7.2 赋值	30
1.7.1 Java 规则	8	2.7.3 赋值语句不是等式	30
1.7.2 标识符	8	2.7.4 初始化	31
1.7.3 关键字	8	2.7.5 引用变量和对象	31
1.7.4 Java 的语句顺序	9	2.7.6 微妙而重要的一点:变量 的两种角色	32
1.7.5 程序格式和注释	9	2.7.7 变量是独立的	32
1.7.6 注释	9	2.7.8 常量不是变量	33
1.7.7 包围的注释	9	2.7.9 语句	33
1.7.8 行注释	10	2.7.10 变化	33
1.8 例行步骤	10	2.8 命令式程序设计 完成任务	34
1.8.1 可访问性	10	2.8.1 整数类型和算术运算	35
1.8.2 准备执行:翻译	11	2.8.2 基本的算术运算符	36
1.8.3 执行	11	2.8.3 优先级	37
1.9 时刻	12	2.8.4 复合赋值运算符	38
1.10 本章小结	13	2.8.5 自增/自减	39
1.11 关键术语	14	2.8.6 操作的顺序	40
1.12 复习题	15	2.9 使用 String 方法	41
1.13 练习题	15	2.9.1 空串	44
1.14 GUI 补充 网页、HTML 和小		2.9.2 灵活且实用的对象	47

2.10 命令式程序设计 简单的条件.....	47	3.11 文件输入过程.....	77
2.10.1 if 语句.....	48	3.11.1 类 File:获取硬盘文件 信息.....	77
2.10.2 条件.....	49	3.11.2 类 FileInputStream:从磁 盘文件中读取字节.....	79
2.11 本章小结.....	51	3.11.3 类 InputStreamReader:将 byte 转化成 char.....	79
2.12 关键术语.....	52	3.11.4 类 BufferedReader:从 char 流到字符串.....	80
2.13 复习题.....	53	3.11.5 结果:为输入打开一个 文件.....	80
2.14 练习题.....	53	3.12 键盘输入.....	82
2.15 GUI 补充内容 图形的绘制和 定位.....	56	3.13 文件输出.....	84
第3章 应用对象和基本类型	64	3.14 网络计算简介.....	85
3.1 简介.....	64	3.14.1 网络的概念.....	85
3.2 创建对象.....	64	3.14.2 网络输入.....	86
3.2.1 使用构造方法创建对象.....	64	3.15 Java 小插曲 Java 文档.....	88
3.2.2 引用变量的重要作用.....	65	3.15.1 Java 应用程序接口.....	88
3.3 Java 小插曲 对象的来源.....	65	3.15.2 API 文档的内容.....	88
3.3.1 重载的和默认的构造 方法.....	65	3.15.3 文档页的层次.....	88
3.3.2 对象来自于何处.....	65	3.15.4 寻找类 String.....	88
3.3.3 不可变对象.....	66	3.16 本章小结.....	90
3.3.4 级联.....	66	3.17 关键术语.....	91
3.3.5 组合.....	67	3.18 复习题.....	91
3.4 命令式程序设计 布尔类型.....	68	3.19 练习题.....	92
3.4.1 boolean 值和变量.....	68	3.20 GUI 补充材料 初识动画.....	94
3.4.2 boolean 表达式.....	69	第4章 定义类	98
3.5 对象的应用.....	69	4.1 简介.....	98
3.6 日期和日历.....	70	4.2 类的定义 方法.....	98
3.7 命令式程序设计 其他的整数 类型.....	72	4.2.1 使用 Logger 类.....	101
3.7.1 为什么我们还要用 int.....	72	4.2.2 基本的类的定义.....	102
3.7.2 混合类型算术运算.....	72	4.3 类的定义 实例变量.....	102
3.7.3 其他的整数类型:short 和 byte.....	73	4.3.1 使用 Stopwatch 类.....	105
3.8 BigInteger.....	73	4.3.2 使用实例变量.....	106
3.9 Java 小插曲 包和 import 语句.....	75	4.4 关于方法的更多内容 返回值.....	106
3.10 输入简介.....	75	4.4.1 使用 Counter 类.....	108
3.10.1 文件输入概要:从磁盘到 程序.....	76	4.4.2 返回值与输出:再谈 Stopwatch.....	109
3.10.2 从字节到字符:Unicode 的例子.....	77	4.4.3 使用返回值.....	110
3.10.3 从 char 到 String:获得文本 的行.....	77	4.5 命令式编程 寻找最小值.....	110
		4.6 Java 小插曲 变量、声明以及 return	

语句	112	5.3.3 Name 类的实现	146
4.6.1 声明顺序	112	5.3.4 完整的 Name 类	149
4.6.2 return 语句	112	5.3.5 使用 Name 类	150
4.6.3 变量及其生存期	113	5.3.6 输出对象:修改 Name 类	150
4.7 关于方法的更多内容 形式 参数	114	5.4 提供类的行为	152
4.7.1 使用类 Minimizer	116	5.5 Java 小插曲 静态方法	155
4.7.2 使用参数	117	5.5.1 回顾我们的第一个程序	155
4.8 状态和行为	117	5.5.2 预定义的静态方法	156
4.9 类的定义 综合起来	118	5.5.3 读取基本类型值	157
4.9.1 标识方法	118	5.6 跟踪类范围内的信息 静态 变量	158
4.9.2 确定状态	119	5.7 定义常量:final	160
4.9.3 实现 test()	120	5.8 Java 小插曲 使用对象	161
4.9.4 实现其他方法	121	5.8.1 “句点”符号	161
4.9.5 使用类 TypingTutor	123	5.8.2 再谈访问控制	162
4.9.6 方法模式:get 和 set	125	5.8.3 关键字 this	162
4.10 Java 小插曲 声明、访问和 对象	125	5.8.4 将对象表示为一个字 符串	163
4.10.1 形参	125	5.8.5 比较两个对象	164
4.10.2 访问控制:public 与 private	125	5.9 重温输入方法 测试输入的 结束	167
4.10.3 对象访问自己的方法	126	5.9.1 Null 值	167
4.10.4 对象及其生存期	126	5.9.2 在 read()方法中使 用 null	167
4.11 本章小结	127	5.10 本章小结	168
4.12 关键术语	127	5.11 关键术语	168
4.13 复习题	128	5.12 复习题	168
4.14 练习题	128	5.13 练习题	169
4.15 GUI 补充材料 一个简单的 动画	130	5.14 GUI 补充材料 创建控制和 交互	169
第 5 章 高级类定义	137	第 6 章 方法内核:命令式程序设计	176
5.1 定制新对象 构造方法	137	6.1 概述	176
5.1.1 TypingTutor 类	137	6.2 一个进行米制转换的类	176
5.1.2 定义重载构造方法	139	6.2.1 模拟度量的数值	176
5.2 Java 小插曲 构造方法、初始化 以及重载	143	6.2.2 基本数据类型 float 和 double	177
5.2.1 默认构造方法	143	6.2.3 实现 convertFeetToMeters() 方法	178
5.2.2 构造方法和初始化	143	6.2.4 方法 printFeetToMeters Table():迭代	179
5.2.3 重载	144		
5.3 示例:Name 类	144		
5.3.1 确定行为	144		
5.3.2 接口	145		