大学本科计算机专业应用型规划教材

丛书主编：高林

面向对象的程序设计与Java

娄不夜 王利 编著

清华大学出版社

大学本科计算机专业应用型规划教材

丛书主编：高林

面向对象的程序设计与Java

娄不夜 王利 编著

清华大学出版社
北京

内容简介

本书以“零”为起点，从类和对象的概念入手，着重介绍 Java 面向对象的程序设计，旨在帮助读者建立面向对象的思想，掌握面向对象编程的基本技能。同时，本书对 Java 语言的各种基本特性及相关的编程技术也做了详细的介绍。

全书共分 12 章，内容包括 Java 程序初步、数据与数据运算、Java 语句、Java 类、继承与接口、数组与字符串、例外处理、多线程编程、输入输出与文件处理、容器布局、事件处理、小应用程序编程等。

本书立足基本理论和方法，注重实践与应用。从应用的角度来介绍基础理论知识，通过例子来说明编程的方法和过程。本书每一章的最后两部分都是小结和精选习题，便于读者复习、总结、巩固、练习与提高。

本书语言流畅，内容翔实，逻辑严谨，分析透彻，适合作为普通高等院校计算机及相关专业的教材，也可作为读者自学 Java 语言和面向对象编程技术的用书。

图书在版编目(CIP)数据

面向对象的程序设计与 Java/娄不夜，王利编著. —北京：清华大学出版社，2004.8
(大学本科计算机专业应用型规划教材/高林主编)
ISBN 7-302-08835-7

Ⅰ. 面…　Ⅱ. ①娄… ②王…　Ⅲ. Java 语言—程序设计—高等学校—教材　Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2004)第 056833 号

出 版 者：清华大学出版社　　地　　址：北京清华大学学研大厦
　　http://www.tup.com.cn　　邮　　编：100084
　　社 总 机：010-62770175　　客户服务：010-62776969
组稿编辑：谢　琛
文稿编辑：汪汉友
版式设计：肖　米
印 刷 者：北京市清华园胶印厂
装 订 者：三河市金元装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　印张：20.75　字数：474 千字
版　　次：2004 年 8 月第 1 版　2004 年 8 月第1 次印刷
书　　号：ISBN 7-302-08835-7/TP・6267
印　　数：1～5000
定　　价：26.00 元

丛书序

大学本科计算机专业应用型规划教材

为适应我国“以信息业带动工业化，发挥后发优势，实现社会生产力的跨越式发展”以及大力发展制造业和优化产业结构的要求，应用型人才培养已成为高等学校人才培养的重要任务。

以微电子技术为基础、计算机技术为主体的信息技术，是当前人类社会中发展最快、渗透性最强、应用面最广的先导技术。信息技术的广泛应用推动着以信息产品制造业、软件业、信息系统集成业和信息咨询服务业为主体的信息产业的发展。在新的世纪里，信息已成为重要的生产要素和战略资源，信息技术成为先进生产力的代表，信息产业将发展成为现代产业的带头产业，人类即将跨越工业时代进入信息时代。因此，信息化成为当今世界经济和社会的发展趋势，大力推进社会和国民经济信息化是推进我国社会主义现代化建设的重要任务。计算机和信息技术的发展不仅需要大批专业技术人才，而且还产生了一批新的职业岗位，毋庸置疑，信息及其相关职业将成为未来最紧缺的职业。

计算机和信息技术与应用的人才需求将呈多元化、多层次趋势，表现在科学、技术、产业、应用、服务诸多方面。不仅需要从事科学、技术研发的人才，而且更需要把研发成果转变为现实产品的技术和管理人才，不仅要有能从事计算机和信息科学、技术工作的人才，而且更需要能从事计算机和信息产业、应用、服务工作的人才，以及在各类人才中的精英人才、领军人物。这实际是对我国计算机和信息类高等教育改革提出了新的要求和新的课题，要求我国高等教育进行结构调整，满足人才培养的多元化，大力培养具有计算机和信息技术专长的应用型人才——他们是这些领域的技术专家和管理专家，可以在相应的行业、企业担任各种技术工作。

目前，我国高等教育中应用型人才培养模式相对落后，如何发展应用型教育已成为课程改革的主要任务。本套教材是以培养计算机和信息类专业本科应用型人才为目的进行的课程与教材改革尝试。在本套教材的策划过程中，清华大学出版社多次组织了由行业企业专家和有丰富教学经验的一线教师参加的研讨会，对应用型高等教育的规律和在计算机教学中的体现进行了深入的研讨。在此基础上我们力求能从整体上把握计算机和信息类应用型人才培养的特征，并体现在这套教材的编写过程中：在教材编写的指导思想上，力求在保持学科科学性的同时，体现工程和技术学科的系统性；在教材

的内容组织上，尽量采用以问题为中心的写作方法，加强案例性教学；在理论联系实际和加强能力培养方面，增加方案性设计习题和实际训练性题目，以培养学生的专门技术能力和完成实际工作任务的能力。

计算机和信息类应用型教材编写还处于改革的初步尝试阶段，希望使用这套教材的教师也能够参与到教材建设工作中来，并提出宝贵意见，以便推动课程改革并提高教材质量。

高　林

2004 年 5 月

前 言

面向对象的程序设计与

编写本书的目的是为了提供一本零起点、全面介绍面向对象程序设计的教程。本书的读者不需要了解C/C++或其他计算机语言及其程序设计的知识，但应该对程序、数据、二进制编码、文件、CPU、存储器、输入输出设备等概念有所理解。通过本书内容的学习，将为读者进一步学习各种Java技术、构建面向对象的软件系统以及网上应用系统打下坚实的基础。

Java是一种完全面向对象的程序设计语言，适合于进行面向对象的软件系统开发和开发出真正的面向对象的软件系统。在Java中，除了数值、字符、布尔型几种基本类型数据外，其他所有类型的数据都是用对象来表示的。一个对象既包含了表示其状态的数据，也包含了描述其行为的方法。而对一个对象的数据的访问，则是通过向对象传递消息、调用对象的方法进行的。这就是所谓的对象处理方式。

从作者多年学习和教学的经验来看，学习Java编程的最大困难在于建立面向对象的思维方式、熟悉面向对象的编程风格。而类和对象的概念则是其中的基础。为此，本书在第1章Java程序初步就引入类和对象的概念，并通过一个简单的例子加以具体的说明。第2章和第3章在介绍数据与数据处理、Java语句时进一步强化类和对象的概念，以及面向对象编程的思维方式。然后，以此为基础，在第4、第5章再系统地介绍Java面向对象的编程方法和技能。

字符串和数组是几乎所有的计算机语言都支持的两种基本数据结构。与大多数计算机语言不同，Java用对象来表示字符串和数组，并采用对象的方式处理。第6、第7章分别专门介绍字符串和数组的使用，同时也为了进一步巩固面向对象的编程技能。

例外处理、线程与同步和输入输出是Java编程的几项基础技术，是开发各种Java应用软件所必不可少的。本书第8、第9、第10章也用较大的篇幅分别对它们进行了介绍。在这里，例外、线程、输入输出流同样是对象，对这些编程技术的掌握也都需要建立在对象和对象处理的概念之上。

图形用户界面(GUI)设计是许多读者感兴趣的课题。从1995年发布的Java 1.0版到现在的1.4版，有关图形用户界面设计的内容已经发生了很大变化。特别是，1.1版用事件委托模型替代原先的事件传递机制来处理事件；1.2版在原先AWT组件的基础上引进了Swing组件，并建议用Swing

取代 AWT。本书第 11、第 12 章分别介绍了基于 Swing 的图形用户界面和小应用程序的设计。同样地，图形用户界面和小应用程序设计中涉及的组件、事件、事件监听器、布局管理器等也都表示为对象并采用对象的方式进行处理。

全书立足基本理论和方法，注重实践与应用。从应用的角度来介绍基础理论知识，通过例子来说明编程的方法和过程。对基本理论、原理、方法和技术的介绍力求概念明确、结构清晰、逻辑严谨。

本书每一章的最后两部分是小结和习题。小结给出了一章的内容要点，便于读者复习与总结；习题主要有选择题和编程题两种形式，用以帮助读者检查学习效果，巩固已学知识，进一步提高能力。

感谢高林教授审阅了全书，并提出了许多宝贵的意见。同时也要感谢清华大学出版社谢琛老师为本书的出版所付出的辛勤工作。另外，作者在编写本书的过程中，参阅过大量的文献和网上资料，特别是 Sun Microsystems 公司 Java 网站（http://java.sun.com/docs/books/）上的内容，在此一并表示感谢。

由于作者学识水平有限，错误和不妥之处，敬请广大读者批评和指正。如果读者有好的建议和要求，请与作者联系，电子邮件地址是 loubuye@163.com。

作　者

2004 年 2 月

目 录

面向对象的程序设计与

第1章 面向对象的程序设计与Java

Java程序初步

Java是Sun Microsystem公司(后简称Sun公司)开发的一种面向对象程序设计语言,是目前网络应用软件开发的主流工具。Java是在C++的基础上发展起来的,它继承了C++中大量的语法成分,但抛弃了C++中冗余和容易引起问题的成分。C++源自于C,而且对业已存在的C代码具有兼容性。与C++不同,Java是一个纯的面向对象的语言。

本章首先介绍Java语言的两个主要特点(面向对象和平台无关),然后通过例子进一步说明类与对象的概念、Java程序的基本结构以及Java应用程序的编译和运行过程,最后介绍Java语言的词法结构。

1.1 Java语言特点

利用Java语言可以开发面向对象的、平台无关的、安全的、多线程的、动态的、分布式的、健壮的、高性能的程序,其中面向对象和平台无关是Java两个最基本也是最主要的特点。

1.1.1 面向对象

客观世界是由客观世界的实体及实体之间的相互联系构成的,实体之间的相互联系产生实体的行为。在试图编写一个程序以解决客观世界的某个问题时,就必然要对客观世界的问题进行抽象。

传统的面向过程的程序设计侧重于实体行为的抽象,而把表示实体状态的属性置于一个被动、附属并相对分离的地位。面向对象的程序设计则把实体的属性和行为作为一个整体加以抽象。对象和类是这一抽象过程的产物,是面向对象程序设计的核心概念。

1. 对象

对象(object)是对客观世界里的任何实体的抽象。被抽象的实体可以是具体的物,也可以指某些概念,例如一部电话机、一名学生、一门课、一台计算机、一个命令按钮等。有些实体可由其他实体组成,例如一台计算机由CPU、存储器、显示器等组成。

实体有属性和行为。属性表示实体的静态特征,所有属性的组合反映实体的状态;行为表示实体的动态特征,一个行为的过程可能会影响或改变实体的状态。客观世界里的任何实体往往都有丰富的属性和复杂的行为。抽象的目的是要从这些丰富和复杂的属性

和行为中选择和提炼出为解决问题所需要的属性和方法，如图 1-1 所示。

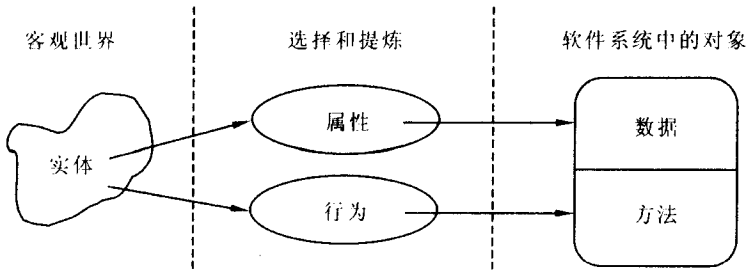

图 1-1　实体到对象的抽象

对象是客观世界实体的软件模型，由数据和方法两部分组成。数据（变量、数组）对应于属性，用于表示对象的状态；方法是对行为的实现，用于表示对象所具有的操作或所能够提供的服务。对象的数据和方法属于对象。与非面向对象程序设计语言中的过程不同，对象的方法必须通过该对象来调用。对象方法的调用意味着该对象的一个行为的发生，它可能会改变这个对象的状态，也可能需要调用另一个对象的某个方法。

对象是数据与方法的封装体。通过封装，对象可以对外界隐藏它的数据和方法的具体实现算法，而只把方法的格式信息（方法名、形参）和行为信息（功能）露在封装界面上，如图 1-2 所示。外界要使用一个对象，并不需要了解对象内部的实现细节，而只需知道对象封装界面上的信息，即该对象能够提供哪些服务。

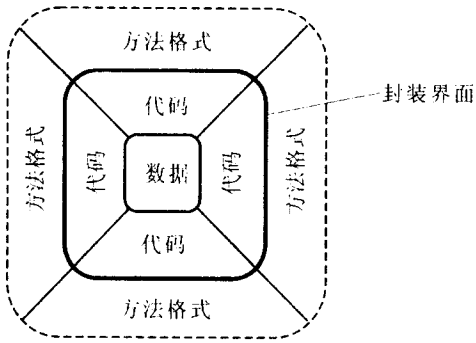

图 1-2　对象封装示意图

一般来说，外界不应该直接读取表示对象状态的数据，而应该通过调用对象的有关方法来获取对象的状态。这种用以返回对象状态的方法称为取值方法。按照惯例，取值方法一般取像 getWidth、getY 这样的名字。反过来，外界也不应该直接设置表示对象状态的数据，但可以通过调用对象的有关方法来设置对象的状态。这种用以设置对象状态的方法称为设值方法。按照惯例，设值方法一般取像 setWidth、setY 这样的名字。在对象设计中，纯粹的设值方法应尽量避免，对象状态的变化应该缘于对象的某种行为的发生，即真正的行为方法的执行。

2. 类

类(class)和对象关系密切,但并不是同一个概念。类是对一类相似对象的描述,这些对象具有相同的属性和行为、相同的变量(数据结构)和方法实现。类定义就是对这些变量和方法实现进行描述。类好比是一类对象的模板,有了类定义后,基于类就可以生成这类对象中任何一个对象。这些对象虽然采用相同的变量来表示状态,但它们在变量上的取值完全可以不同。这些对象一般有着不同的状态,且彼此间相对独立。

如图 1-3 所示,有一个矩形类 Rectangle,其中定义了两个变量 width 和 height,分别表示矩形的宽和高;定义的两个方法 getArea 和 getPerimeter,分别计算矩形的面积与周长。基于类创建的矩形对象虽然都用变量 width 和 height 来表示它们的状态,但每个矩形在变量上的取值是独立的。类中定义的方法指明了矩形对象可以向外界提供的服务。方法尽管定义在类中,但执行方法的主体是对象。同一个方法,如果由不同的对象去执行,一般会产生不同的结果。这里,我们把基于某个类生成的对象称为这个类的实例。在 Java 中,除了数组对象,任何一个对象都是某个类的一个实例。

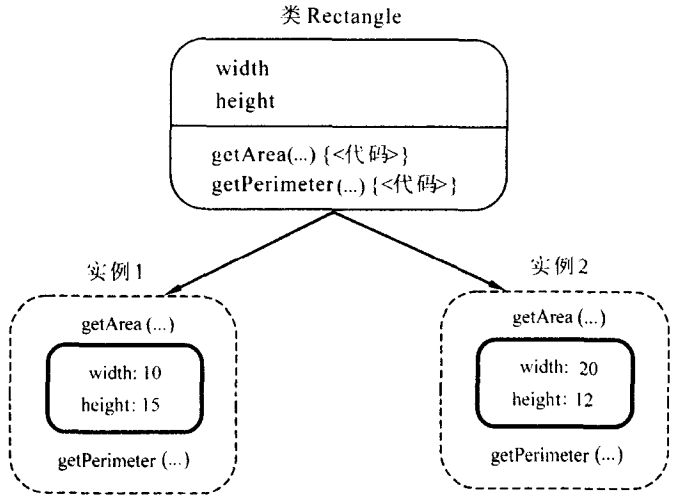

图 1-3 类与对象的关系示意图

因为类的每个实例共用相同的方法,所以在程序运行时,不管基于这个类产生了多少个实例,类中定义的每个方法的代码在内存中只需要一个副本。另一方面,类的各个实例作为对象是相互独立的,每个实例有各自的表示其状态的数据。程序运行时,每当产生类的一个实例,系统就会为该实例建立相应的一组实例变量。也就是说,类中定义的实例变量在内存中可能有多个副本,每个副本属于某个实例。

类应该提供对象封装的机制,即在类的定义中,可以指定每个对象露在封装界面上的信息是什么,隐藏在封装界面里的内容是什么。在 Java 类中,主要通过指定变量和方法的访问级别来定义对象的这种封装界面。用 public(公共的)修饰的变量和方法可以被外界访问或调用;用 private(私有的)修饰的变量和方法则不能被外界访问或调用。根据上