21世纪高等学校计算机教育实用规划教材

Java高级编程与应用

栾 颖 编著

清华大学出版社

21世纪高等学校计算机教育实用规

Java高级编程与应用

栾 颖 编著

清华大学出版社
北京

内 容 简 介

全书共分 10 章，内容包括 Java 的基础知识、Java 常量与变量、类和对象、库的设计、接口和线程、集合与迭代、Java 安全性编程、加密和解密、使用消息验证码、图形界面应用、容器嵌套类、事件与 Swing 包、Java 网络编程、Java 的多媒体编程以及 Java 应用实例。

本书内容详尽，循序渐进，可作为高等院校计算机及相关专业的 Java 语言程序设计或网络编程基础等课程的教材，也可作为应用开发人员自学参考用书。

图书在版编目(CIP)数据

Java 高级编程与应用/栾颖编著. —北京：清华大学出版社，2014
21 世纪高等学校计算机教育实用规划教材
ISBN 978-7-302-38043-6

Ⅰ. ①J…　Ⅱ. 栾…　Ⅲ. ①JAVA 语言－程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 219817 号

责任编辑：魏江江　薛　阳
封面设计：常雪影
责任校对：白　蕾
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：23　　**字　　数**：557 千字
版　　次：2014 年 12 月第 1 版　　**印　　次**：2014 年 12 月第1 次印刷
印　　数：1～2000
定　　价：39.50 元

产品编号：061035-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善，社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，合理调整和配置教育资源，在改革和改造传统学科专业的基础上，加强工程型和应用型学科专业建设，积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业，积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度，从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时，不断更新教学内容、改革课程体系，使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用，工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展，急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前，工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践，如现有的计算机教材中有不少内容陈旧（依然用传统专业计算机教材代替工程型和应用型学科专业教材），重理论、轻实践，不能满足新的教学计划、课程设置的需要；一些课程的教材可供选择的品种太少；一些基础课的教材虽然品种较多，但低水平重复严重；有些教材内容庞杂，书越编越厚；专业课教材、教学辅助教材及教学参考书短缺，等等，都不利于学生能力的提高和素质的培养。为此，在教育部相关教学指导委员会专家的指导和建议下，清华大学出版社组织出版本系列教材，以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

（1）面向工程型与应用型学科专业，强调计算机在各专业中的应用。教材内容坚持基本理论适度，反映基本理论和原理的综合应用，强调实践和应用环节。

（2）反映教学需要，促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

（3）实施精品战略，突出重点，保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本，合理配套。基础课和专业基础课教材要配套，同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材，教学参考书，文字教材与软件教材的关系，实现教材系列资源配套。

(5) 依靠专家，择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人：魏江江 weijj@tup.tsinghua.edu.cn

前　言

人类已经进入了21世纪，科学技术突飞猛进，知识经济初见端倪，特别是信息技术和网络技术的迅速发展和广泛应用，对社会的政治、经济、军事、科技和文化等领域产生越来越深刻的影响，也正在改变着人们的工作、生活、学习和交流方式。信息的获取、处理、交流和应用能力，已经成为人们最重要的能力之一。培养一大批掌握和应用现代信息技术和网络技术的人才，在全球信息化的发展中占据主动地位，不仅是经济和社会发展的需要，也是计算机和信息技术教育者的历史责任。

1995年底，Java程序设计语言闯入了Internet领域，并迅速占据了显著地位。距离Sun公司第一次发布Java已经整整19年了，19年对于计算机飞速发展的进程来说不算短，它足以淘汰掉许多技术，也足以考验真正的强者。时至今日，Java已成为Internet中最受欢迎、最具影响的编程语言之一。

由于时代的进步，网络的飞快发展，Java的身影随处可见，而要成为一个优秀的Java编程人员，一定要打下坚实的专业基础。能够建立相当复杂的用户界面虽然很好，但如果代码臃肿、很耗内存和效率低下，用户则不会满意。

本书具有概念清晰、例子丰富、内容覆盖面广、内容难度适中、实践与理论紧密相结合等特点。

全书共分10章，第1章介绍Java的综合概述，包括Java的虚拟机概述、Java常量与变量等内容；第2章为类和对象，主要包括库的设计、类创建、扩展类等内容；第3章介绍接口和线程，主要包括接口、克隆和多线程等内容；第4章介绍集合，主要包括集合与迭代、集合实现等内容；第5章介绍Java安全性编程，主要包括基于口令的加密和解密、使用消息验证码等内容；第6章介绍图形界面应用，主要包括布局管理器、容器嵌套类等内容；第7章介绍事件与Swing包，主要包括基础窗口类、组件类等内容；第8章介绍Java网络编程，主要包括网络编程的基础、Java网络编程等内容；第9章介绍Java的多媒体编程，主要包括Image类、图像映射等内容；第10章综合介绍Java应用实例。

本书主要由栾颖编写，此外参加编写的还有刘超、邓俊辉、梁朗星、李旭波、张棣华、刘泳、邓耀隆、何正风和周品。

本书编写过程中由于时间仓促，加之作者水平有限，错误和疏漏之处在所难免，在此诚恳地期望得到各领域的专家和广大读者的批评指正。

编　者

2014年10月

目　录

第1章 Java的综合概述

根据Sun Microsystems公司的定义，Java是“一个简单、健全、面向对象、平台独立、多线程、动态并且通用的编程环境”。如此简单的定义却将Java拓展到许多全新的领域，某些领域最初甚至看不到与Java有丝毫联系。如今，有微处理器的地方，几乎就有Java。无论是大型企业或是微型器件，无论是廉价手机还是巨型机，都用到了Java。为了让Java支持如此多变的复杂环境，由此而开发了大量不同版本的API（Application Programming Interface，应用程序编程接口），这些API都植根在一组共同的核心类上。

尽管如此，要成为一名优秀的Java程序员，仍要先打好基础。高度复杂的用户界面制作固然有用，但如果代码臃肿、既耗内存效率又低，是无法让用户满意的。Java开发人员可能会用到成百上千种开发方法。本书将着重介绍Java的核心语言特性，例如多线程和内存管理，这些特性能使一个Java应用程序焕然一新。

Java广为适用的核心，也就是其普及的原因，在于其平台独立性，Java的WORA（Write Once，Run Anywhere）特性源自其自身的运行方式，尤其是利用抽象执行环境来分享Java代码和底层操作系统。

1.1 面向对象初步

Java是一种面向对象的编程语言，要想真正掌握Java，首先必须明确的就是什么是面向对象以及面向对象的核心思想。最近几年，面向对象编程在软件开发领域掀起了一阵狂热的浪潮，得到了迅猛发展，受到越来越多的关注，也有越来越多的人加入到Java的开发行列。那么，究竟是什么原因使如此多的人热衷于面向对象编程呢？

面向对象编程（Object Oriented Programming，OOP）具有多方面的吸引力，对于生产管理人员来说，它实现了一次性投入多次使用，使开发成本更加低廉。对于设计分析人员来说，利用UML建模更加直观、方便，完成的程序更加易于维护。对于程序员来说，更加理解并领会设计人员的意图，使开发过程不再变得枯燥无味。

UML（unified modeling language）是一种统一建模语言，是在1997年11月由国际对象管理组织批准认定的面向对象建模语言，它的伟大之处就是统一了软件的分析、设计和编写的规范。就像一张施工设计图，只要是按照统一模式设计的图纸，每一个人都可以看懂。

任何事情都是双方面的，在利用面向对象的好处时，我们必须为掌握它而付出努力，也就是思考对象的过程，需要采用抽象思维，而不是程序化的思维方式，这种思想的形式需要长时间的努力和实践。所以说面向对象不仅仅是一种编程语言的实现，更重要的是抽象思

想的形成。这种思想就是把一切事物都抽象化为对象，并给对象赋予一定的特征，简单地说，面向对象就是抽象与具体的过程，抽象的过程是得到对象，具体的过程是对象的实例化。也许有人会问那以前软件开发的方式是怎样的呢？

这个问题可以这样回答，在 Java 诞生以前开发软件的方式基本上都是面向过程的。接下来也许大家更想知道面向过程与面向对象到底有什么样的区别呢？可用一个比较形象的比喻来简单说明一下：如果把程序实现的过程形象地比喻为盖房子，那么面向过程就是将水、木、土等原材料按照房子的建筑顺序一直盖上去，一旦房子过时需要重新盖的话，所有的原材料都将没有用处了。而面向对象则是在动手盖房子之前，先烧砖，将原材料烧制成一块块用于不同目的的砖（这些砖并不像日常生活中所见的砖都是一样的形状，这里强调的是用于不同目的的砖，大家可以理解为不同的形状。比如说有的地方需要圆形的砖，有的地方需要方形的砖，那么就制成圆形或方形的砖），再根据房子的需要将这些砖按照一定的顺序组织起来。很明显，一旦房子过时了，砖还可以搬到别处用（如果用专业一点的话说就是代码的复用性），这就是面向对象的精髓。当然砖烧制得好坏，这才是面向对象编程最具挑战性的一面，也就是说能烧制出可以不断重复使用的砖，对于程序开发人员来说，正是它的魅力所在。

由于砖的制成，盖新房子时，可以利用部分旧砖，而不需要重新去烧制同样的砖。这对于房子的主人来说，在一定程度上降低了成本，对于施工者来说，使盖房子这种工作也变得轻松了。

相信大家都见过积木，用一块块不同形状的木块，可以搭建不同形状的房子，这些不同形状的木块就是所要烧制的砖。当然，积木是已经给做好的不同形状的木块，现在编程中需要的是程序员去做这些不同形状的木块。这是不是很有挑战性呀？

当然，通过这几段文字不可能完全涵盖面向对象的思想，只是想让大家对面向对象有一个初步的认识，并在以后的学习过程中尝试着用面向对象的思维方式去思考问题、分析问题。

任何事件都可以看成是一对象，对象随处可见。可以把对象理解为现实生活中存在的实实在在的物品，如灯、桌子等；也可以理解为抽象中的每一件事件、逻辑等。每个物品、每件独立的事物都可以作为一个对象；还可以把一类东西作为一个对象，如砖和积木等，把不同形状的砖或积木作为一个对象；还能把多个相同形状的砖或积木看作一个对象；也可以把所有的砖或积木作为一个对象。

如图 1-1 所示，有两个不同颜色、形状和品种的梨，每一个梨可以作为一个对象（因为它们确实有不同之处），而所有的梨也可以作为一个对象（因为它们确实有相同之处）。通过这幅图可以看出，两个不同颜色、形状、品种的梨可以看成一个统一的对象，那就是梨，可以说不同的对象可以构建成新的对象，结合上面的描述，可以得出下面的结论：

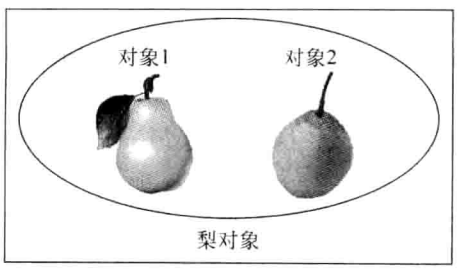

图 1-1　对象

（1）所有的事物都可以是对象。

（2）对象可以是一个容器，有自己的存储空间，可以容纳不同的对象。如图 1-1 所示，对象梨

可以容纳两个或多个不同形状、颜色、品种的梨。也就是说，我们可以根据已经存在的对象构建新的对象。

(3) 对象是程序构建的基本元素，程序是一大堆对象的组合。通过各个对象间的消息传递，每个对象都知道自己要做些什么。

(4) 同一类的所有对象都能接收相同的信息。根据我们的图示，梨都有品种这个属性，可以向对象 1 传递品种的请求，也可以向对象 2 传递品种请求。

1.2 Java 的虚拟机概述

Java 语言的引擎是 JVM(一种抽象计算系统用于解释编译过的 Java 程序)，其他编程语言(如 C 或 C++)中的编译器会根据特定的处理器，通常还根据特定的操作系统将源代码编译为可执行程序。该可执行程序无须外部支持就能在计算机上运行。

这种模式的缺点是不具有可移植性；在一个操作系统下编译的代码无法在另一个操作系统上运行，而必须在程序需要运行的各个操作系统上重新编译。另外，由于每个处理器供应商对编译器特性要求不同，因此在特定操作系统中为特定处理器系列(如 Intel x86、SPARC 或 Alpha 系列)编译的代码，在相同操作系统但不同类型的处理器上可能也无法运行。例如，同样是运行 Windows NT 的 Alpah 工作站和 Intel PC，前一平台上所编译的代码就无法在后一平台上工作。

当人们开始为 Internet 编写应用程序时，该问题就变得尤为严重。这些应用程序需要兼容各种不同的操作系统、各种不同的平台以及各种不同的浏览器。解决问题的唯一办法是开发一种平台独立的语言。

20 世纪 90 年代初期，Sun Microsystems 的开发人员提出了用于消息类电子产品的平台独立的语言，遗憾的是，平台独立的概念在当时因太过超前而搁置。不过，随着 Internet 的出现，开发人员意识到这种语言的巨大潜力，Java 由此诞生。

Java 语言可移植性的关键在于 Java 编译器输出的并非标准可执行代码。Java 编译器会生成一组已优化的指令，称为字节码(bytecode)程序。字节码是按规定格式排列的一系列字节。字节码程序由运行时系统(即 JVM)进行解释。因此在一个平台上生成字节码程序后，其他安装了 JVM 的平台就能运行该程序。

虽然不同平台上的 JVM 各不相同，但上文做法是可行的。换句话说，在 UNIX 工作站上编译的 Java 程序能够在 PC 或 Mac 上运行。只要按 Java 语言的标准形式编写源代码，然后编译为字节码程序，每个 JVM 都能将字节码解释为所在平台对应的本地调用(也就是解释为指定处理器能够识别的语言)。正是通过这种抽象形式，不同平台上的各种操作系统能够以相同的方式完成一些操作，如打印、文件访问以及硬件操作。

字节码的一个特点(也有人认为是一个缺点)就是字节码在计算机上运行时并不是由处理器直接执行的。字节码程序是通过 JVM 运行的，由 JVM 解释字节码，因此 Java 是一种解释型语言(Interpreted Language)。Java 作为解释型语言而具有平台独立性，但相比标准的可执行类型代码，Java 的性能也稍差。不过，自 JSDK(Java Software Development Kit，Java 软件开发包)1.3 版发布后，Java 程序与其他语言程序之间的速度差距已经基本消除。

表 1-1 中列出了编译型语言与解释型语言的对比。