

疯狂 Java讲义

新东方IT培训广州中心 李刚 编著
软件教学总监

疯狂源自梦想

技术成就辉煌

疯狂源自梦想

技术成就辉煌





疯狂

Java讲义

新东方IT培训广州中心 李刚 编著
软件教学总监

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

从 2000 年至今, Java 语言一直是应用最广的开发语言, 并拥有最广泛的开发人群。如今, Java 已经不再简单地是一门语言, 它更像一个完整的体系, 一个系统的开发平台。更甚至, 它被延伸成一种开源精神。

本书深入介绍了 Java 编程的相关方面, 全书内容覆盖了 Java 的基本语法结构、Java 的面向对象特征、Java 集合框架体系、Java 泛型、异常处理、Java GUI 编程、JDBC 数据库编程、Java 注释、Java 的 IO 流体系、Java 多线程编程、Java 网络通信编程和 Java 反射机制。共覆盖了 java.awt、java.lang、java.io 和 java.nio、java.sql、java.text、java.util、javax.swing 包下绝大部分类和接口。

本书并不单纯从知识角度来讲解 Java, 而是从解决问题的角度来介绍 Java 语言, 所以本书中介绍了大量实用案例开发: 五子棋游戏、梭哈游戏、仿 QQ 的游戏大厅、MySQL 企业管理器、仿 EditPlus 的文本编辑器、多线程、断点下载工具、Spring 框架的 IoC 容器……这些案例既能让读者巩固每章的知识, 又可以让读者学以致用、激发编程自豪感, 进而引爆内心的编程激情。本书光盘里包含书中所有示例的代码, 如果读者希望获得课后练习的程序代码, 请自行登录 <http://www.lcegang.org> 站点下载。

本书为所有打算深入掌握 Java 编程的读者而编写, 适合各种层次的 Java 学习者和工作者阅读, 也适合作为大学教育、培训机构的 Java 教材。但如果只是想简单涉猎 Java, 则本书过于庞大, 不适合阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

疯狂 Java 讲义 / 李刚编著. —北京: 电子工业出版社, 2008.10
ISBN 978-7-121-06646-7

I. 疯… II. 李… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 064472 号

责任编辑: 朱沐红

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

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: 850×1168 1/16 印张: 56.75 字数: 1720 千字 彩插: 2

印 次: 2008 年 10 月第 1 次印刷

印 数: 4000 册 定价: 99.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



如何学习 Java

——谨以此文献给打算以编程为职业、并愿意为之疯狂的人

经常看到有些学生、求职者捧着一本类似 JBuilder 入门、Eclipse 指南之类的图书学习 Java，当他们学会了在这些工具中拖出窗体、安装按钮之后，就觉得自己掌握、甚至精通了 Java；又或是找来一本类似 JSP 动态网站编程之类的图书，学会使用 JSP 脚本编写一些页面后，就自我感觉掌握了 Java 开发。

还有一些学生、求职者听说 J2EE、Spring 或 EJB 很有前途，于是立即跑到书店、或图书馆找来一本相关图书。希望立即学会它们，然后进入软件开发业、大显身手。

还有一些学生、求职者非常希望找到一本既速成、又大而全的图书，比如突击 J2EE 开发、一本书精通 J2EE 之类的图书（包括笔者曾出版的《轻量级 J2EE 企业应用实战》一书，据说销量不错），希望这样一本图书就可以打通自己的“任督二脉”，一跃成为 J2EE 开发高手。

也有些学生、求职者非常喜欢 J2EE 项目实战、项目大全之类的图书，他们的想法很单纯：我按照书上介绍，按图索骥、依葫芦画瓢，应该很快就可学会 J2EE，很快就能成为一个受人羡慕的 J2EE 程序员了。

.....

凡此种种，不一而足。但最后的结果往往是失败，因为这种学习没有积累、没有根基，学习过程中困难重重，每天都被一些相同、类似的问题所困扰，起初热情十足，经常上论坛询问，按别人的说法解决问题之后很高兴，既不知道为什么错？也不知道为什么对？只是盲目地抄袭别人的说法。最后的结果有两种：

- ① 久而久之，热情丧失，最后放弃学习。
- ② 大部分常见问题都问遍了，最后也可以从事一些重复性开发，但一旦遇到新问题，又将束手无策。

第二种情形在普通程序员中占了极大的比例，笔者多次听到、看到（在网络上）有些程序员抱怨：我做了 2 年多 Java 程序员了，工资还是 3000 多点。偶尔笔者会与他们聊聊工作相关内容，他们会告诉笔者：我也用 Spring 了啊，我也用 EJB 了啊……他们感到非常不平衡，为什么我的工资这么低？其实笔者很想告诉他们：你们太浮躁了！你们确实是用了 Spring、Hibernate 又或是 EJB，但你们未想过为什么要用这些技术？用这些技术有什么好处？如果不用这些技术行不行？

很多时候，我们的程序员把 Java 当成一种脚本，而不是一门面向对象的语言。他们习惯了在 JSP 脚本中使用 Java，但从不去想 JSP 如何运行，Web 服务器里的网络通信、多线程机制，为何一个 JSP 页面能同时向多个请求者提供服务？更不会想如何开发 Web 服务器；他们像代码机器一样编写 Spring Bean 代码，但从不去理解 Spring 容器的作用，更不会想如何开发 Spring 容器。

有时候，笔者的学生在编写五子棋、梭哈等作业感到困难时，会向他们的大学师兄、朋友求救，这些程序员告诉他：不用写了，网上有下载的！听到这样回答，笔者不禁感到哑然：网上还有 Windows

下载呢！网上下载和自己编写是两码事。偶尔，笔者会怀念以前黑色屏幕、绿荧荧字符时代，那时候程序员很单纯：当我们想偷懒时，习惯思维是写一个小工具；现在程序员很聪明：当他们想偷懒时，习惯思维是从网上下一个小工具。但是，谁更幸福？

当笔者的学生把他们完成的小作业放上互联网之后，然后就有许多人称他们为“高手”！这个称呼却让他们万分惭愧；惭愧之余，他们也感到万分欣喜，非常有成就感，这就是编程的快乐。编程的过程，与寻宝的过程完全一样：历经辛苦，终于找到心中的梦想，这是何等的快乐？

如果真的打算将编程当成职业，那就不应该如此浮躁，而是应该扎扎实实先学好 Java 语言，然后按 Java 本身的学习规律，踏踏实实一步一个脚印地学习，把基本功练扎实了才可获得更大的成功。

实际情况是，有多少程序员真正掌握了 Java 的面向对象？真正掌握了 Java 的多线程、网络通信、反射等内容？有多少 Java 程序员真正理解了类初始化时内存运行过程？又有多少程序员理解 Java 对象从创建到消失的全部细节？有几个程序员真正独立地编写过五子棋、梭哈、桌面弹球这种小游戏？又有几个 Java 程序员敢说：我可以开发 Struts？我可以开发 Spring？我可以开发 Tomcat？很多人又会说：这些都是许多人开发出来的！实际情况是：许多开源框架的核心最初完全是由一个人开发的。现在这些优秀程序已经出来了！你，是否深入研究过它们，是否深入掌握了它们？

如果要真正掌握 Java，包括后期的 Java EE 相关技术（例如 Struts、Spring、Hibernate 和 EJB 等），一定要记住笔者的话：绝不要从 IDE（如 JBuilder、Eclipse 和 NetBeans）工具开始学习！IDE 工具的功能很强大，初学者学起来也很容易上手，但也非常危险：因为 IDE 工具已经为我们做了许多事情，而软件开发者要全部了解软件开发的全部步骤。

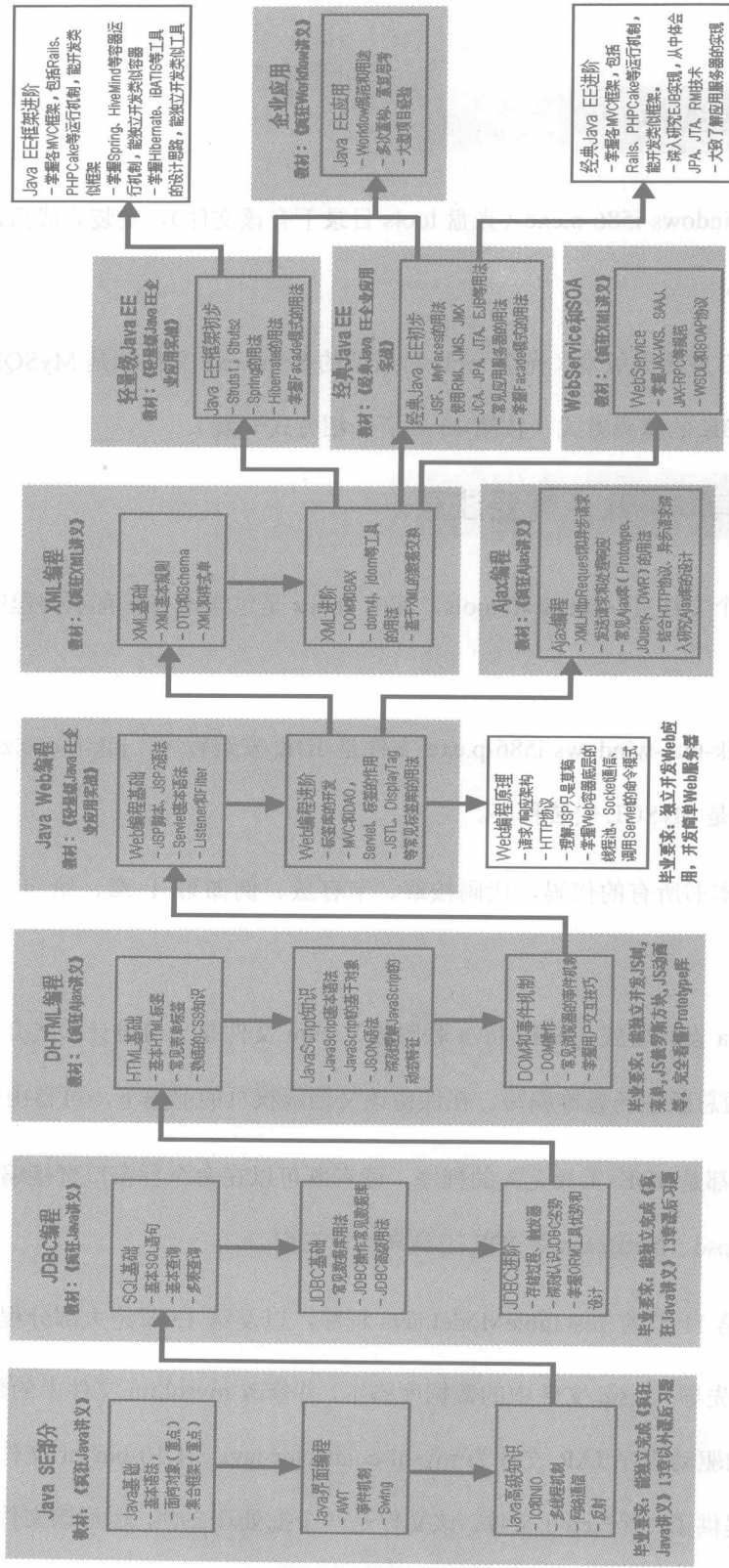


2008 年 3 月 17 日



疯狂 Java 学习路线图

笔者就自己对 Java EE 体系的理解，对 Java EE 学习者给出一个粗略线路图：





备注：

1. 没有灰色覆盖的区域稍有难度，请谨慎尝试。
2. 本学习线路图不涉及设计模式、软件方法学等概念，但希望大家能从开发中悟道。
3. 本人并不认为 Spring、Hibernate 很复杂，只要基础扎实，掌握框架是水到渠成的。


光盘说明


代码运行的前提条件:


 安装 jdk-6u3-windows-i586-p.exe (光盘 tools 目录下有该文件), 安装完成后, 按第一章所介绍方式添加环境变量。


 安装 MySQL5.0 或更高版本 (光盘 tools 目录下的 Setup.exe 文件就是 MySQL-5.0.45), 安装 MySQL 时候选择 GBK 的编码方式 (按第 13 章所介绍方式安装)。


除此之外, 还有如下几点需要注意:


 本书光盘分两个文件夹: codes 和 tools, 其中 codes 是包含本书所有示例程序。tools 下包含本书所使用的工具。

 tools 目录下 jdk-6u3-windows-i586-p.exe 文件是 JDK 安装程序, jdk-6-doc.zip 是 JDK1.6 的 API 文档, Setup.exe 是 MySQL 安装程序。

 codes 下包含本书所有的代码, 代码按章、节存放。例如第 1 章、第 4 节的代码存放在 codes/01/1-4 路径下。

 codes 下的 Java 设计模式 (疯狂 Java 联盟版).chm 文件是一份设计模式介绍文档, 该文档主要由疯狂 Java 联盟总版主杨恩雄编写。在保留该文档版权页的前提下, 可自由传播该文档。

 本书所有代码都是 IDE 工具无关的程序, 读者既可以在命令行窗口直接编译、运行这些代码, 也可以导入 Eclipse、NetBeans 等 IDE 工具来运行它们。

 本书第 12 章第 11 节的 TestTableModel.java 程序, 以及第 13 章绝大部分程序都需要连接数据库, 所以读者需要先导入 *.sql 文件中的数据库脚本, 并修改 mysql.ini 文件中的数据库连接信息。连接数据库时所用的驱动程序 JAR 文件为 mysql-connector-java-3.1.10-bin.jar 文件。这些需要连接数据库的程序里还提供了一个 *.cmd 文件, 该文件是一个批处理文件, 运行该文件可以运行响应的 Java 程序, 例如 DatabaseMetaDataTest.java 对应的 *.cmd 文件为 runDatabaseMetaDataTest.cmd。

前言

Java 语言赢得了前所未有的成功：从 2001 年到 2007 年，Java 语言一直是世界上应用最广泛的编程语言，因此现在有越来越多的工作者、学习者正努力进入 Java 领域或将要努力进入 Java 领域。

为了帮助广大工作者、学习者能真正掌握 Java 编程，感受到 Java 语言的魅力，领会到 Java 编程的快乐，笔者根据近 3 年来的 Java 授课经验，精心编写了本书。

当笔者开始写作本书之前，已经接触了非常多刚刚毕业的大学生，他们告诉笔者：之所以选择编程，仅仅是想找一份工作。笔者问他们，不喜欢编程吗？他们回答：编程有什么乐趣？还不就是写 $1+2+3+\dots+100=?$ ……这样的程序。于是笔者知道了：他们误解了程序。于是笔者告诉他们：如果我来上 C 语言，至少讲到可以编写出一个简单的游戏外挂，如果再深入一点，可以结合操作系统理论，编写一个操作系统的雏形——这才叫写程序。

程序的作用是：解决问题——如果你的程序不能解决你自己的问题，如何期望你的程序去解决别人的问题呢？那你的程序的价值何在？所以笔者认为：最好的学习方法就是“案例驱动”——知道为什么要用这个知识点，才去学这个知识点，而不是盲目学习！因此本书强调编程实战，强调以项目激发编程兴趣。

在笔者的授课过程中，几乎所有学习好的学生都有过通宵的经历，没有人逼迫、要求他们，只因为他们无法抗拒编程的诱惑——这也是笔者后来创立疯狂 Java 实训营的直接缘由。其实每个能在编程领域里小有所得的人，都曾有过通宵达旦的经历，否则很难成为一个优秀的程序员。有人可能会觉得，通宵达旦太辛苦了——但烈日下挥汗如雨地打球苦不苦？废寝忘食地玩游戏的人苦不苦？他们自然乐在其中！同理，对于疯狂的程序员而言，通宵达旦地编程、调试并不是苦，而是一种享受。如果读者早一天经历这种境界，那就会早一天打开编程的心窍。

笔者不想告诉你：看完这本书你会成为高手！在编程领域里，没有所谓的“武林秘笈”，再好的书一定要配合大量练习，否则书里的知识依然属于作者，而读者则仿佛身入宝山而一无所获的笨汉。所以本书配合了大量高强度的练习，希望读者强迫自己去完成这些项目。这些习题的答案可以登录 <http://www.leegang.org> 站点下载。

本书将一如既往地保持笔者既有的风格：简单。有一封读者来信让笔者记忆很深刻，他告诉笔者：相对有些人喜欢“深奥”的图书相比，他更喜欢“简单”的 IT 图书，“简单”的东西很清晰，明确，下一步该怎么做，为什么这样做，一切都清清楚楚，无须任何猜测、想象——正好符合计算机哲学：0 就是 0，1 就是 1，中间没有任何回旋的余地。他的来信非常符合笔者的观点：IT 图书应该是简单的、易懂的，学完之后可以带来提高，能产生效益的图书。本书正是如此，读者阅读本书时会感觉很浅显，但读完后能开发的程序又非常实用。

本书有什么特点



本书并不是一本简单的 Java 入门教材，也不是一门“闭门造车”式的 Java 读物。本书来自于笔者近 3 年的 Java 培训经历，凝结了笔者近 3000 个小时的授课经验，总结了几百个 Java 学员学习过程中的典型错误。

因此，本书具有如下三个特点：

1. 案例驱动，引爆编程激情

本书不再是知识点的铺陈，而是致力于将知识点融入实际项目的开发，所以本书中涉及了大量 Java 案例：仿 QQ 的游戏大厅、MySQL 企业管理器、仿 EditPlus 的文本编辑器、多线程、断点下载工具……希望读者通过编写这些程序找到编程的乐趣。

2. 再现李刚老师课堂氛围

本书的内容是笔者近 3 年授课经历的总结，知识体系取自李刚疯狂 Java 实战课程体系。本书力求再现笔者的课堂氛围：以浅显比喻代替乏味的讲解，以疯狂实战代替空洞的理论。书中包含了大量“注意”、“学生提问”部分，这些正是几百个 Java 学员所犯错误的汇总。

3. 注释详细，轻松上手

为了降低读者阅读的难度，书中代码的注释非常详细，几乎每两行代码就有一行注释。不仅如此，本书甚至还把一些简单理论作为注释穿插到代码中，力求让读者能轻松上手。

本书所有程序中关键代码以粗体字标出，也是为了帮助读者能迅速找到这些程序的关键点。

衷心感谢



一直以来，很想感谢我的妻子郭秋霞女士，她给了笔者生活上、精神上无微不至的照顾，并承担几乎所有日常家务，使得笔者能心无旁骛地专注于程序设计思考和研究。

感谢高洪霞、朱沐红编辑长期的信任和辛勤的工作，她们给予了笔者充分的创作自由，并在笔者写作过程中提出了许多宝贵的意见。

感谢所有参加疯狂 Java 实训的学生，他们是本书的“试验品”，他们所付出的努力证明了本书的价值，他们的反馈让本书更实用。

本书写给谁看







如果你仅仅想对 Java 有所涉猎，那么本书并不适合你；如果你想全面掌握 Java 语言，并使用 Java 来解决问题、开发项目，或者希望以 Java 编程作为你的职业，那么本书将非常适合你。希望本书能引爆你内心潜在的编程激情，如果本书能让你产生废寝忘食的感觉，那笔者就非常欣慰了。

2008 年 3 月 17 日

目 录 CONTENTS

第 1 章 Java 概述..... 1	2.1.2 程序的三种基本结构..... 25
1.1 Java 语言的发展简史..... 2	2.1.3 面向对象程序设计简介..... 27
1.2 Java 的竞争对手及各自优势..... 4	2.1.4 面向对象的基本特征..... 28
1.2.1 C#简介和优势..... 4	2.2 UML (统一建模语言) 介绍..... 29
1.2.2 Ruby 简介和优势..... 4	2.2.1 用例图..... 30
1.2.3 Python 的简介和优势..... 5	2.2.2 类图..... 31
1.3 Java 程序运行机制..... 6	2.2.3 组件图..... 33
1.3.1 高级语言的运行机制..... 6	2.2.4 部署图..... 33
1.3.2 Java 程序的运行机制和 JVM..... 6	2.2.5 顺序图..... 34
1.4 开发 Java 的准备..... 7	2.2.6 活动图..... 34
1.4.1 安装 JDK..... 8	2.2.7 状态机图..... 36
学生提问 不是说 JVM 是运行 Java 程序的虚拟机吗? 那 JRE 和 JVM 的关系是怎样的呢?..... 8	2.3 Java 的面向对象特征..... 36
学生提问 为什么不安装公共 JRE 系统呢?..... 9	2.3.1 一切都是对象..... 37
1.4.2 设置 PATH 环境变量..... 10	2.3.2 类和对象..... 37
学生提问 为什么选择设置用户变量, 用户变量和系统变量有什么区别呢?..... 11	2.4 本章小结..... 37
1.5 第一个 Java 程序..... 12	第 3 章 数据类型和运算符..... 38
1.5.1 编辑 Java 源代码..... 12	3.1 注释..... 39
1.5.2 编译 Java 程序..... 13	3.1.1 单行注释和多行注释..... 39
学生提问 当我们使用编译 C 程序时, 不仅需要指定存放目标文件的位置, 也需要指定目标文件的文件名, 这里使用 javac 编译 Java 程序时怎么不需要指定目标文件的文件名呢?..... 13	3.1.2 文档注释..... 40
1.5.3 运行 Java 程序..... 14	学生提问 API 文档是什么?..... 40
1.5.4 根据 CLASSPATH 环境变量定位类..... 15	学生提问 为什么要掌握查看 API 文档的方法?..... 42
1.6 Java 程序的基本规则..... 16	3.2 标识符和关键字..... 45
1.6.1 Java 程序的组织形式..... 16	3.2.1 分隔符..... 45
1.6.2 Java 源文件的命名规则..... 17	3.2.2 标识符规则..... 47
1.6.3 初学者容易犯的错误..... 18	3.2.3 Java 关键字..... 47
1.7 垃圾回收机制..... 20	3.3 数据类型分类..... 48
1.8 何时开始使用 IDE 工具..... 21	3.4 基本数据类型..... 48
学生提问 老师, 我想学习 Java 编程, 到底是学习 Eclipse 好呢, 还是学习 JBuilder 好呢?..... 21	3.4.1 整型..... 49
1.9 本章小结..... 22	3.4.2 字符型..... 49
本章练习..... 22	3.4.3 浮点型..... 51
第 2 章 理解面向对象..... 23	3.4.4 布尔型..... 52
2.1 面向对象..... 24	3.5 基本类型的类型转换..... 53
2.1.1 结构化程序设计简介..... 24	3.5.1 自动类型转换..... 53
	3.5.2 强制类型转化..... 54
	3.5.3 表达式类型的自动提升..... 56
	3.6 直接量..... 57
	3.6.1 直接量的类型..... 57
	3.6.2 直接量的赋值..... 58
	3.7 运算符..... 58
	3.7.1 算术运算符..... 58

3.7.2	赋值运算符	61	
3.7.3	位运算符	62	
3.7.4	扩展后的赋值运算符	64	
3.7.5	比较运算符	65	
学生提问	Java 为什么要对这些数据进行缓存呢?	67	
3.7.6	逻辑运算符	67	
3.7.7	三目运算符	68	
3.7.8	运算符的结合性和优先级	69	
3.8	本章小结	70	
	本章练习	70	
第 4 章	流程控制和数组	71	
4.1	顺序结构	72	
4.2	分支结构	72	
4.2.1	if 条件语句	72	
4.2.2	switch 分支语句	76	
4.3	循环结构	78	
4.3.1	while 循环语句	78	
4.3.2	do while 循环语句	79	
4.3.3	for 循环	80	
4.3.4	嵌套循环	83	
4.4	控制循环结构	84	
4.4.1	使用 break 结束循环	84	
4.4.2	使用 continue 结束本次循环	85	
4.4.3	使用 return 结束方法	87	
4.5	数组类型	87	
4.5.1	理解数组: 数组也是一种类型	87	
学生提问	int[] 是一种类型吗? 怎么使用这种类型呢?	88	
4.5.2	定义数组	88	
4.5.3	数组的初始化	88	
学生提问	能不能只分配内存空间, 不赋初始值呢?	89	
4.5.4	使用数组	90	
学生提问	为什么要我记住这些异常信息?	91	
4.5.5	JDK1.5 提供了 foreach 循环	91	
4.6	深入数组	93	
4.6.1	内存中的数组	93	
学生提问	为什么有栈内存和堆内存之分?	93	
4.6.2	基本类型数组的初始化	95	
4.6.3	引用类型数组的初始化	96	
4.6.4	没有多维数组	99	
学生提问	我是否可以图 4.13 中灰色覆盖的数组元素再次指向另一个数组? 这样不可以扩展成三维数组吗? 甚至扩展到更多维的数组?	101	
4.6.5	操作数组的工具类	102	
4.6.6	数组的应用举例	103	
4.7	本章小结	106	
	本章练习	106	
第 5 章	面向对象 (上)	107	
5.1	类和对象	108	
5.1.1	定义类	108	
学生提问	构造器不是没有返回值吗? 为什么不能用 void 修饰呢?	110	
5.1.2	对象的产生和使用	110	
5.1.3	对象、引用和指针	111	
5.1.4	对象的 this 引用	112	
5.2	方法详解	116	
5.2.1	方法的所属性	116	
5.2.2	方法的参数传递机制	116	
5.2.3	形参长度可变的方法	120	
5.2.4	递归方法	121	
5.2.5	方法重载	123	
学生提问	为什么方法的返回值类型不能用于区分重载的方法?	124	
5.3	成员变量和局部变量	124	
5.3.1	成员变量和局部变量	125	
5.3.2	成员变量的初始化和内存中的运行机制	128	
5.3.3	局部变量的初始化和内存中的运行机制	130	
5.3.4	变量的使用规则	130	
5.4	隐藏和封装	132	
5.4.1	理解封装	132	
5.4.2	使用访问控制符	132	
5.4.3	package 和 import	135	
5.4.4	Java 的常用包	140	
5.5	深入构造器	140	
5.5.1	使用构造器执行初始化	141	
学生提问	构造器是创建 Java 对象的途径, 是不是说构造器完全负责创建 Java 对象?	141	
5.5.2	构造器的重载	142	
学生提问	为什么要用 this 来调用另一个重载的构造器? 我把另一个构造器里的代码复制、粘贴到这个构造器里不就可以了吗?	143	

5.6	类的继承	144	6.6.2	接口的定义	193
5.6.1	继承的特点	144	6.6.3	接口的继承	195
5.6.2	重写父类的方法	145	6.6.4	使用接口	195
5.6.3	父类实例的 super 引用	146	6.6.5	接口和抽象类	197
	我们只是创建了一个 Ostrich 对象时，哪来的 Bird 对象？	147	6.6.6	面向接口编程	198
5.6.4	调用父类构造器	148	6.7	内部类	202
	为什么我创建 Java 对象时从未感觉到 java.lang.Object 的构造器被调用过？	150	6.7.1	非静态内部类	202
5.7	多态	151		非静态内部类对象和外部类对象的关系是怎样的？	206
5.7.1	多态性	151	6.7.2	静态内部类	207
5.7.2	引用变量的强制类型转换	152		为什么静态内部类实例方法也不能访问外部类的实例属性呢？	207
5.7.3	instanceof 运算符	154		接口里是否能定义内部接口？	208
5.8	继承与组合	154	6.7.3	使用内部类	208
5.8.1	使用继承的注意点	155		既然内部类是外部类的成员，是否可以作为外部类定义子类，在子类中再定义一个内部类来重写其父类中的内部类？	211
5.8.2	利用组合实现复用	156	6.7.4	局部内部类	211
	使用组合关系来实现复用，需要创建两个 Animal 对象，是不是意味着使用组合关系时系统开销更大？	159	6.7.5	匿名内部类	212
5.9	初始化块	159	6.7.6	闭包 (Closure) 和回调	215
5.9.1	使用初始化块	160	6.8	枚举类	217
5.9.2	初始化块和构造器	161	6.8.1	手动实现枚举类	217
5.9.3	静态初始化块	162	6.8.2	枚举类入门	219
5.10	本章小结	164	6.8.3	枚举类的属性、方法和构造器	220
	本章练习	165	6.8.4	实现接口的枚举类	223
第 6 章	面向对象 (下)	166	6.8.5	包含抽象方法的枚举类	224
6.1	基本数据类型的包装类	167	6.9	对象与垃圾回收	225
6.2	处理对象	170	6.9.1	对象在内存中的状态	226
6.2.1	打印对象和 toString 方法	170	6.9.2	强制垃圾回收	227
6.2.2	== 和 equals 比较运算符	172	6.9.3	finalize 方法	228
6.3	类成员	175	6.9.4	对象的软、弱和虚引用	230
6.3.1	理解类成员	175	6.10	修饰符的适用范围	233
6.3.2	单例 (Singleton) 类	176	6.11	使用 JAR 文件	234
6.4	final 修饰符	177	6.11.1	jar 命令详解	235
6.4.1	final 变量	177	6.11.2	创建可执行的 JAR 包	237
6.4.2	final 方法	181	6.11.3	关于 JAR 包的技巧	238
6.4.3	final 类	182	6.12	本章小结	239
6.4.4	不可变类	182		本章练习	239
6.4.5	缓存实例的不可变类	186	第 7 章	Java 集合	240
6.5	抽象类	188	7.1	Java 集合概述	241
6.5.1	抽象方法和抽象类	188	7.2	Collection 和 Iterator 接口	243
6.5.2	抽象类的作用	191	7.2.1	使用 Iterator 接口遍历集合元素	244
6.6	更彻底的抽象：接口	192	7.2.2	使用 foreach 循环遍历集合元素	246
6.6.1	接口的概念	192	7.3	Set 接口	247
			7.3.1	HashSet 类	247

hashCode 方法对于 HashSet 的作用是什么?	249
7.3.2 TreeSet 类	252
7.3.3 EnumSet 类	259
7.4 List 接口	261
7.4.1 List 接口和 ListIterator 接口	261
7.4.2 ArrayList 和 Vector 实现类	264
7.4.3 固定长度的 List	266
7.5 Queue 接口	266
7.5.1 LinkedList 实现类	266
7.5.2 PriorityQueue 实现类	269
7.6 Map	270
7.6.1 HashMap 和 Hashtable 实现类	271
7.6.2 SortedMap 接口和 TreeMap 实现类	276
7.6.3 WeakHashMap 实现类	279
7.6.4 IdentityHashMap 实现类	280
7.6.5 EnumMap 实现类	281
7.7 HashSet 和 HashMap 的性能选项	282
7.8 操作集合的工具类: Collections	283
7.8.1 排序操作	283
7.8.2 查找, 替换操作	287
7.8.3 同步控制	288
7.8.4 设置不可变集合	288
7.9 烦琐的接口: Enumeration	289
7.10 本章小结	290
本章练习	290
第 8 章 泛型	291
8.1 泛型入门	292
8.1.1 编译时不检查类型的异常	292
8.1.2 手动实现编译时检查类型	293
8.1.3 使用泛型	294
8.2 深入泛型	294
8.2.1 定义泛型接口、类	295
8.2.2 从泛型类派生子类	296
8.2.3 并不存在泛型类	298
8.3 类型通配符	298
8.3.1 使用类型通配符	299
8.3.2 设定类型通配符的上限	300
8.3.3 设定类型形参的上限	302
8.4 泛型方法	303
8.4.1 定义泛型方法	303
8.4.2 泛型方法和类型通配符的区别	306
8.4.3 设定通配符的下限	307
8.4.4 泛型方法与方法重载	309
8.5 擦除和转换	309
8.6 泛型与数组	311
8.7 本章小结	313
第 9 章 与运行环境交互	314
9.1 与用户互动	315
9.1.1 运行 Java 程序的参数	315
9.1.2 使用 Scanner 获取键盘输入	316
9.1.3 使用 BufferedReader 获取键盘输入	318
9.2 系统相关	319
9.2.1 System 类	319
9.2.2 Runtime 类	321
9.3 常用类	322
9.3.1 Object 类	322
9.3.2 String、StringBuffer 和 StringBuilder 类	322
9.3.3 Math 类	327
9.3.4 Random 类	328
9.3.5 BigDecimal 类	330
9.4 处理日期的类	333
9.4.1 Date 类	333
9.4.2 Calendar 类	334
9.4.3 TimeZone 类	337
9.5 正则表达式	338
9.5.1 创建正则表达式	338
9.5.2 使用正则表达式	341
9.6 程序国际化	345
9.6.1 Java 国际化的思路	346
9.6.2 Java 支持的语言和国家	346
9.6.3 完成程序国际化	347
9.6.4 使用 MessageFormat 处理包含占位符的字符串	349
9.6.5 使用类文件代替资源文件	350
9.6.6 使用 NumberFormat 格式化数字	350
9.6.7 使用 DateFormat 格式化日期	352
9.7 本章小结	355
本章练习	355
第 10 章 异常处理	356
10.1 异常概述	357
10.2 异常处理机制	358
10.2.1 使用 try...catch 捕获异常	359
10.2.2 异常类的继承体系	360
10.2.3 访问异常信息	363
10.2.4 使用 finally 回收资源	364
10.2.5 异常处理的嵌套	367
10.3 Checked 异常和 Runtime 异常体系	367

10.3.1	使用 throws 声明抛出异常	367	11.7.1	画图的实现原理	418
10.4	使用 throw 抛出异常	369	11.7.2	使用 Graphics 类	419
10.4.1	抛出异常	369	11.8	处理位图	424
10.4.2	自定义异常类	371	11.8.1	Image 抽象类和 BufferedImage 实现类	424
10.4.3	catch 和 throw 同时使用	372	11.8.2	使用 ImageIO 输入/输出位图	427
10.4.4	异常链	373	11.9	剪贴板	432
10.5	Java 的异常跟踪栈	374	11.9.1	数据传递的类和接口	432
10.6	异常处理规则	377	11.9.2	传递文本	432
10.6.1	不要过度使用异常	377	11.9.3	使用系统剪贴板传递图像	434
10.6.2	不要使用过于庞大的 try 块	378	11.9.4	使用本地剪贴板来传递对象引用	439
10.6.3	避免使用 Catch All 语句	378	11.9.5	通过系统剪贴板传递 Java 对象	442
10.6.4	不要忽略捕获到的异常	379	11.10	本章小结	445
10.7	本章小结	379	本章练习	445	
	本章练习	379	第 12 章	Swing 编程	446
第 11 章	AWT 编程	380	12.1	Swing 概述	447
11.1	GUI (图形用户界面) 和 AWT	381	12.2	Swing 基本组件的用法	448
11.2	AWT 容器	382	12.2.1	Swing 组件层次	448
11.3	布局管理器	385	12.2.2	AWT 组件的 Swing 实现	449
11.3.1	FlowLayout 布局管理器	386	学生提问	为什么单击 Swing 多行文本域时不是弹出像 AWT 多行文本域中的右键菜单?	455
11.3.2	BorderLayout 布局管理器	387	12.2.3	使用 JToolBar 创建工具条	458
学生提问	BorderLayout 最多只能放置 5 个组件吗? 那它还有什么作用?	388	12.2.4	使用 JColorChooser 和 JFileChooser	461
11.3.3	GridLayout 布局管理器	389	12.2.5	使用 JOptionPane	469
11.3.4	GridBagLayout 布局管理器	390	12.3	Swing 中的特殊容器	475
11.3.5	CardLayout 布局管理器	393	12.3.1	使用 JSplitPane	475
11.3.6	绝对定位	395	12.3.2	使用 JTabbedPane	477
11.3.7	BoxLayout 布局管理器	396	12.3.3	使用 JLayeredPane、JDesktopPane 和 JInternalFrame	481
学生提问	图 11.15 和图 11.16 显示的所有按钮都紧挨在一起, 如果希望像 FlowLayout、GridLayout 等布局管理器指定组件的间距该怎么办?	397	12.4	Swing 拖放功能	490
11.4	AWT 常用组件	398	12.4.1	拖放目标	491
11.4.1	基本组件	398	12.4.2	拖放源	494
11.4.2	对话框	400	12.4.3	简化的拖放操作	495
11.5	事件处理	403	12.5	Applet 和 JApplet	497
11.5.1	Java 事件模型的流程	403	12.5.1	Applet 简介及其安全性	498
11.5.2	事件和事件监听器	405	12.5.2	开发 Applet 类	498
11.5.3	事件适配器	409	12.5.3	使用 HTML 页面装载 Applet	500
11.5.4	事件监听器的实现形式	410	12.5.4	appletviewer 简介	502
11.6	AWT 的菜单	413	12.5.5	Applet 的生命周期和动画机制	503
11.6.1	菜单条、菜单和菜单项	413	学生提问	程序中重写 paint 方法时绘制了一个字符串, 但图 12.28 上则绘制出了如此之多的字符串, 这是为什么呢?	505
11.6.2	右键菜单	416	12.5.6	使用 Applet 创建音乐播放器	505
学生提问	为什么即使我没有给多行文本域编写右键菜单, 但当我在多行文本域上单击右键时一样会弹出右键菜单?	418			
11.7	在 AWT 中绘图	418			

12.6	使用 JProgressBar、ProgressMonitor 和 BoundedRangeModel 创建进度条	508	本章练习	595	
12.6.1	创建进度条	508	第 13 章 JDBC 编程	596	
12.6.2	创建进度对话框	512	13.1 JDBC 基础	597	
12.7	使用 JSlider 和 BoundedRangeModel 创建滑动条	513	13.1.1 JDBC 简介	597	
12.8	使用 JSpinner 和 SpinnerModel 创建微调控制器	517	13.1.2 JDBC 驱动程序类型	598	
12.9	使用 JList、JComboBox 创建列表框	520	13.2 SQL 语法介绍	599	
12.9.1	简单列表框	520	13.2.1 安装数据库	599	
12.9.2	不强制存储列表项的 ListModel 和 ComboBoxModel	524	13.2.2 关系数据库基本概念和 MySQL 基本命令	601	
12.9.3	强制存储列表项的 DefaultListModel 和 DefaultComboBoxModel	528	13.2.3 SQL 语句基础	603	
学生提问	为什么 JComboBox 提供了添加、删除列表项目的方法？而 JList 没有提供添加、删除列表项的方法呢？	530	13.2.4 DDL 语句	604	
12.9.4	使用 ListCellRenderer 改变列表项外观	530	13.2.5 DML 语句语法	618	
12.10	使用 JTree 和 TreeModel 创建树	532	13.2.6 select 语句和 SQL 函数	620	
12.10.1	创建树	533	13.2.7 分组和组函数	627	
12.10.2	拖动、编辑树节点	536	13.2.8 多表连接查询和子查询	629	
12.10.3	监听节点事件	540	13.3 JDBC 的典型用法	635	
12.10.4	使用 TreeCellRendering 改变节点外观	542	13.3.1 JDBC 常用接口和类简介	635	
12.11	使用 JTable 和 TableModel 创建表格	549	13.3.2 JDBC 编程步骤	637	
12.11.1	创建 JTable	549	学生提问	前面给出的仅是 MySQL 和 Oracle 两种数据库的驱动，我看不出驱动类字符串有什么规律啊。如果我希望使用其他数据库，我怎么找到其他数据库的驱动类呢？	637
学生提问	我们指定的表格数据、表格列标题都是 Object 类型的数组，JTable 如何显示这些 Object 对象？	550	13.4 执行 SQL 语句的方式	639	
12.11.2	TableModel 和监听器	555	13.4.1 使用 executeUpdate 执行 DDL 和 DML 语句	640	
12.11.3	TableColumnModel 和监听器	561	13.4.2 使用 execute 方法执行 SQL 语句	642	
12.11.4	实现排序	564	13.4.3 使用 PreparedStatement 执行 SQL 语句	644	
12.11.5	绘制单元格内容	567	13.4.4 使用 CallableStatement 调用存储过程	650	
12.11.6	编辑单元格内容	570	13.5 管理结果集	652	
12.12	使用 JFormattedTextField 和 JTextPane 创建格式文本	576	13.5.1 可滚动、可更新的结果集	652	
12.12.1	监听 Document 的改变	576	13.5.2 处理 Blob 类型数据	654	
12.12.2	使用 JPasswordField	578	13.5.3 使用 ResultSetMetaData 分析结果集	660	
12.12.3	使用 JFormattedTextField	579	13.6 事务处理	663	
12.12.4	使用 JEditorPane	588	13.6.1 事务的概念和 MySQL 事务支持	663	
12.12.5	使用 JTextPane	588	13.6.2 JDBC 的事务支持	665	
12.13	本章小结	595	13.6.3 批量更新	667	
			13.7 分析数据库信息	668	
			13.7.1 使用 DatabaseMetaData 分析数据库信息	668	
			13.7.2 使用系统表分析数据库信息	670	
			13.7.3 选择合适的分析方式	671	

13.8	使用连接池管理连接	671	15.8.5	另一种自定义序列化机制	733
13.9	本章小结	673	15.8.6	版本	735
	本章练习	673	15.9	Java 新 IO	736
第 14 章	Annotation (注释)	674	15.9.1	Java 新 IO 概述	736
14.1	基本 Annotation	675	15.9.2	使用 Buffer	737
14.1.1	限定重写父类方法: @Override	675	15.9.3	使用 Channel	740
14.1.2	标示已过时: @Deprecated	676	15.9.4	编码集和 CharSet	743
14.1.3	抑制编译器警告: @SuppressWarnings	677	15.9.5	文件锁	745
14.2	自定义 Annotation	677	15.10	本章小结	747
14.2.1	定义 Annotation	677		本章练习	747
14.2.2	提取 Annotation 的信息	679	第 16 章	多线程	748
14.2.3	使用 Annotation 的例子	680	16.1	线程概述	749
14.3	JDK 的元 Annotation	685	16.1.1	线程和进程	749
14.3.1	使用 @Retention	685	16.1.2	多线程的优势	750
14.3.2	使用 @Target	686	16.2	线程的创建和启动	750
14.3.3	使用 @Documented	686	16.2.1	继承 Thread 类创建线程类	751
14.3.4	使用 @Inherited	687	16.2.2	实现 Runnable 接口创建线程类	752
14.4	使用 APT 处理 Annotation	688	16.2.3	两种方式所创建线程的对比	754
14.5	本章小结	693	16.3	线程的生命周期	754
第 15 章	输入/输出	694	16.3.1	新建和就绪状态	754
15.1	File 类	695	16.3.2	运行和阻塞状态	755
15.1.1	访问文件和目录	695	16.3.3	线程死亡	757
15.1.2	文件过滤器	697	16.4	控制线程	758
15.2	理解 Java 的 IO 流	698	16.4.1	join 线程	758
15.2.1	流的分类	698	16.4.2	后台线程	759
15.2.2	流的概念模型	699	16.4.3	线程睡眠: sleep	760
15.3	字节流和字符流	700	16.4.4	线程让步: yield	761
15.3.1	InputStream 和 Reader	700	16.4.5	改变线程优先级	762
15.3.2	OutputStream 和 Writer	703	16.5	线程的同步	763
15.4	输入/输出流体系	705	16.5.1	线程安全问题	764
15.4.1	处理流的用法	705	16.5.2	同步代码块	766
15.4.2	输入/输出流体系	706	16.5.3	同步方法	767
15.4.3	转换流	708	16.5.4	释放同步监视器的锁定	769
	学生提问 怎么没有把字符流转换成字节流的转换流呢?	708	16.5.5	同步锁 (Lock)	770
15.4.4	推回输入流	710	16.5.6	死锁	772
15.5	重定向标准输入/输出	711	16.6	线程通信	774
15.6	Java 虚拟机读写其他进程的数据	713	16.6.1	线程的协调运行	774
15.7	RandomAccessFile	716	16.6.2	使用条件变量控制协调	778
15.8	对象序列化	720	16.6.3	使用管道流	780
15.8.1	序列化的含义和意义	720	16.7	线程组和未处理的异常	782
15.8.2	使用对象流实现序列化	720	16.8	Callable 和 Future	785
15.8.3	对象引用的序列化	723	16.9	线程池	787
15.8.4	自定义序列化	728	16.10	线程相关类	789
			16.10.1	ThreadLocal 类	789
			16.10.2	包装线程不安全的集合	791

16.10.3 线程安全的集合类.....	792	18.1.1 JVM 和类.....	851
16.11 本章小结.....	792	18.1.2 类的加载.....	852
第 17 章 网络编程.....	793	18.1.3 类的连接.....	853
17.1 网络编程的基础知识.....	794	18.1.4 类的初始化.....	853
17.1.1 网络基础知识.....	794	18.1.5 类初始化的时机.....	854
17.1.2 IP 地址和端口号.....	795	18.2 类加载器.....	855
17.2 Java 的基本网络支持.....	796	18.2.1 类加载器简介.....	856
17.2.1 使用 InetAddress.....	796	18.2.2 类加载机制.....	857
17.2.2 使用 URLDecoder 和 URLEncoder.....	797	18.2.3 创建并使用自定义的类加载器.....	858
17.2.3 使用 URL 和 URLConnection.....	798	18.2.4 URLClassLoader 类.....	862
17.3 基于 TCP 协议的网络编程.....	805	18.3 通过反射查看类信息.....	863
17.3.1 TCP 协议基础.....	806	18.3.1 获得 Class 对象.....	863
17.3.2 使用 ServletSocket 创建 TCP 服务器端.....	806	18.3.2 从 Class 中获取信息.....	864
17.3.3 使用 Socket 进行通信.....	807	18.4 使用反射生成并操作对象.....	868
17.3.4 加入多线程.....	810	18.4.1 创建对象.....	868
17.3.5 记录用户信息.....	813	18.4.2 调用方法.....	870
17.3.6 使用 NIO 实现非阻塞 Socket 通信.....	823	18.4.3 访问属性值.....	873
17.4 基于 UDP 协议的网络编程.....	829	18.4.4 操作数组.....	874
17.4.1 UDP 协议基础.....	829	18.5 使用反射生成 JDK 动态代理.....	875
17.4.2 使用 DatagramSocket 发送、接收 数据.....	829	18.5.1 使用 Proxy 和 InvocationHandler 创建动态代理.....	876
17.4.3 使用 MulticastSocket 实现多点 广播.....	834	18.5.2 动态代理和 AOP.....	878
17.5 使用代理服务器.....	845	18.6 反射和泛型.....	882
17.5.1 直接使用 Proxy 创建连接.....	846	18.6.1 泛型和 Class 类.....	882
17.5.2 使用 ProxySelector 选择代理 服务器.....	847	18.6.2 使用反射来获取泛型信息.....	884
17.6 本章小结.....	849	18.7 本章小结.....	885
本章练习.....	849	本章练习.....	886
第 18 章 类加载和反射.....	850	上善若水——兼谈我的创作理念.....	887
18.1 类的加载、连接和初始化.....	851		