

疯狂

Java讲义

精粹

李刚 编著

疯狂源自梦想

技术成就辉煌

疯狂源自梦想

技术成就辉煌



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



疯狂

Java#X



李刚 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是《疯狂 Java 讲义》的精粹版，本书同样保留了《疯狂 Java 讲义》的特性：本书基于 Java 7 完成，全面介绍了 Java 7 的新特性。

本书删除了《疯狂 Java 讲义》中关于图形界面编程、数据库编程的内容，保留了 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 注释、Java 的 IO 流体系、Java 多线程编程、Java 网络通信编程和 Java 反射机制的内容。本书的这种调整一方面是为了对《疯狂 Java 讲义》进行精简，另一方面也是应广大高校老师的反馈：高校课程前期已经讲授过数据库相关内容，因此本书删除了相关内容。本书的写作是基于 Java 7 完成的，全面介绍了 Java 7 的二进制整数、菱形语法、增强 switch 语句、多异常捕获、自动关闭资源的 try 语句、JDBC 4.1 新特性、NIO.2、AIO 等新特性。

本书光盘里包含书中所有示例的代码和《疯狂 Java 实战演义》的所有项目代码。但由于本书并未包含 Java 图形界面编程的相关知识，因此读者在阅读带图形界面的项目代码时可能有一定困难，此时读者可以参考《疯狂 Java 讲义》的相关内容，也可以登录 <http://www.crazyit.org> 站点与笔者及本书庞大的读者群相互交流。

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。本书专门针对高校课程进行过调整，尤其适合作为大学教育、培训机构的 Java 教材。本书不适合只想简单涉猎 Java 的读者阅读。

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 讲义精粹 / 李刚编著. —北京：电子工业出版社，2012.1
ISBN 978-7-121-15579-6

I. ①疯… II. ①李… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 270650 号

策划编辑：张月萍

责任编辑：葛 娜

印 刷：北京中新伟业印刷有限公司

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：787×1092 1/16 印张：37.25 字数：923 千字 彩插：4

印 次：2012 年 1 月第 1 次印刷

印 数：4000 册 定价：79.00 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、多层机制，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？

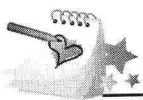
如果真的打算将编程当成职业，那就不应该如此浮躁，而是应该扎扎实实先学好 Java 语言，然后按 Java 本身的学习规律，踏踏实实一步一个脚印地学习，把基本功练扎实了才可获得更大的成功。

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。




2011 年 12 月 17 日



光盘说明

一、光盘内容

 本光盘是《疯狂 Java 讲义精粹》一书的配书光盘，书中的代码按章、按节存放，即第 3 章第 1 节所使用的代码放在 codes 文件夹的 03\3.1 文件夹下，依此类推。

另：书中每份源代码也给出与光盘源文件的对应关系，方便读者查找。

本光盘 codes 目录下有 14 个文件夹，其内容和含义说明如下：


(1) 01~13 文件夹名对应于《疯狂 Java 讲义精粹》中的章名，即第 3 章所使用的代码放在 codes 文件夹的 03 文件夹下，依此类推。

(2) 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。

(3) 光盘根目录下提供了一个“Java 设计模式（疯狂 Java 联盟版）.chm”文件，这是一份关于设计模式的电子教材，由疯狂 Java 联盟的杨恩雄亲自编写、制作，他同意广大读者阅读、传播这份开源文档。

(4) 因为有些读者提出关于《疯狂 Java 讲义》课后习题标准答案的问题，因此本书光盘根目录下包含一个 project_codes 文件夹，该文件夹里包含了疯狂 Java 联盟的杨恩雄编写的《疯狂 Java 实战演义》一书的光盘内容，该光盘中包含了大量实战性很强的项目，这些项目基本覆盖了《疯狂 Java 讲义精粹》课后习题的要求，读者可以参考相关案例来完成《疯狂 Java 讲义精粹》的课后习题。

二、运行环境

 本书中的程序在以下环境调试通过：

(1) 安装 jdk-7-windows-i586.exe，安装完成后，添加 CLASSPATH 环境变量，该环境变量的值为.;%JAVA_HOME%/lib/tools.jar;%JAVA_HOME%/lib/dt.jar。如果为了可以编译和运行 Java 程序，还应该在 PATH 环境变量中增加%JAVA_HOME%/bin。其中 JAVA_HOME 代表 JDK（不是 JRE）的安装路径。如何安装上面工具，请参考本书的第 1 章。

(2) 安装 MySQL 5.5 或更高版本，安装 MySQL 时选择 GBK 编码方式。

三、注意事项

(1) 代码中有大量代码需要连接数据库，读者应修改数据库 URL 以及用户名、密码，让这些代码与读者运行环境一致。如果项目下有 SQL 脚本，则导入 SQL 脚本即可；如果没

有 SQL 脚本，系统将在运行时自动建表，读者只需创建对应的数据库即可。

(2) 在使用本光盘的程序时，请将程序拷贝到硬盘上，并去除文件的只读属性。

四、技术支持

如果您使用本光盘中遇到不懂的技术问题，您可以登录如下网站与作者联系：

<http://www.crazyit.org>



前 言

2011年7月，Oracle如约发布了Java 7正式版。接下来笔者对《疯狂Java讲义》进行了升级：使用Java 7改写了全书所有程序，全面介绍了Java 7的各种新特性。

《疯狂Java讲义》面市的近3年时间内，得到了广大读者的一致好评。笔者也收到了一些关于《疯狂Java讲义》的建议：有人觉得这本书实在太太、太厚了；也有读者建议把《疯狂Java讲义》拆分成上、下册来出版；针对这些建议，笔者慎重地斟酌了很久：Java经过10多年的发展，已经成为世界上应用最广泛的编程语言，而且Java语言的强大是有目共睹的。功能如此丰富、强大，历史如此悠久的一门编程语言，如果希望通过一本三、四百页的书能讲透，笔者觉得不太现实。虽然笔者也很希望对《疯狂Java讲义》进行精简，但实在难以取舍。如果把《疯狂Java讲义》拆分成上、下册来出版，这种做法会增加读者的经济负担（1本书变成了2本书，总价会高很多）。后来出版社的朋友启发说：可以尝试把《疯狂Java讲义》部分内容抽取成“精粹本”；又想到有高校老师反应：学校在安排Java课程之前，已经安排了数据库开发相关课程，《疯狂Java讲义》中关于SQL、JDBC编程的内容有些重复。出于上面两种需求，笔者对《疯狂Java讲义》内容进行了调整，删除了部分内容，得到这本《疯狂Java讲义精粹》，本书更适合作为高校教材使用。

在以“疯狂Java体系”图书为教材的疯狂软件教育中心（www.fkjava.org）里，经常有学生询问：为什么叫疯狂Java这个名字？也有一些读者通过网络、邮件来询问这个问题。其实这个问题答案可以在本书第1版的前言中找到。疯狂的本质是一种“享受编程”的状态。在一些不了解编程的人看来：编程的人总面对着电脑、在键盘上敲打，这种生活实在太枯燥了，但实际上是因为他们并未真正了解编程，并未真正走进编程。在外人眼中：程序员不过是在敲打键盘；但在程序员心中：程序员敲出的每个字符，都是程序的一部分。

程序是什么呢？程序是对现实世界的数字化模拟。开发一个程序，实际是创造一个或大或小的“模拟世界”。在这个过程中，程序员享受着“创造”的乐趣，程序员沉醉在他所创造“模拟世界”里：疯狂地设计、疯狂地编码实现。实现过程不断地遇到问题，然后解决它；不断地发现程序的缺陷，然后重新设计、修复它——这个过程本身就是一种享受。一旦完全沉浸到编程世界里，程序员是“物我两忘”的，眼中看到的、心中想到的，只有他正在创造的“模拟世界”。

在学会享受编程之前，编程学习者都应该采用“案例驱动”的方式，学习者需要明白程序的作用是：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？——知道一个知识点能解决什么问题，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。

仅仅只是看完这本书，你不会成为高手！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。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参考本书所附光盘中《疯狂Java实战演义》的配套代码。如果需要获得编程思路和交流，可以登录<http://www.crazyit.org>与广大读者和笔者交流。

本书有什么特点



本书是《疯狂 Java 讲义》的精粹版，本书并不是一本简单的 Java 入门教材。本书的内容完全取自《疯狂 Java 讲义》，只是删减了原书中关于 SQL、JDBC 编程，以及 Java 图形界面编程的相关内容。

因此，本书具有如下三个特点。

1. 阐述透彻、原理清晰

本书并不是简单地罗列 Java 语法规则，而是尽量从语法设计者的角度向读者解释每个语法规则的作用、缘由；本书力求从运行机制来解释代码的运行过程，从内存分配的细节上剖析程序的运行细节。阅读本书不仅要求读者知道怎么做，而且要求读者能理解“为什么这么做”。

2. 再现李刚老师课堂氛围

本书的内容是笔者 6 年多授课经历的总结，知识体系取自疯狂 Java 实战的课程体系。

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，这些正是上千个 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。



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。







2011-12-17

目 录 CONTENTS

第 1 章 Java 语言概述	1	1.6.3 初学者容易犯的错误	19
1.1 Java 语言的发展简史	2	1.7 垃圾回收机制	21
1.2 Java 的竞争对手及各自优势	5	1.8 何时开始使用 IDE 工具	23
1.2.1 C#简介和优势	5	 我想学习 Java 编程,到底是	
1.2.2 Ruby 简介和优势	5	学习 Eclipse 好呢,还是学	
1.2.3 Python 简介和优势	6	习 NetBeans 好呢?	23
1.3 Java 程序运行机制	7	1.9 本章小结	24
1.3.1 高级语言的运行机制	7	本章练习	24
1.3.2 Java 程序的运行机制和			
JVM	7	第 2 章 数据类型和运算符	25
1.4 开发 Java 的准备	9	2.1 注释	26
1.4.1 下载和安装 Java 7 的 JDK	9	2.1.1 单行注释和多行注释	27
 不是说 JVM 是运行 Java 程		2.1.2 文档注释	27
序的虚拟机吗? 那 JRE 和		 API 文档是什么?	27
JVM 的关系是怎样的呢?	9	 为什么要掌握查看 API 文档	
 为什么不安装公共 JRE		的方法?	30
呢?	10	2.2 标识符和关键字	34
1.4.2 设置 PATH 环境变量	11	2.2.1 分隔符	34
 为什么选择用户变量? 用		2.2.2 标识符规则	36
户变量与系统变量有什么		2.2.3 Java 关键字	36
区别?	13	2.3 数据类型分类	37
1.5 第一个 Java 程序	13	2.4 基本数据类型	37
1.5.1 编辑 Java 源代码	13	2.4.1 整型	38
1.5.2 编译 Java 程序	14	2.4.2 Java 7 新增的二进制整数	39
 当我们编译 C 程序时,不仅		2.4.3 字符型	40
需要指定存放目标文件的		2.4.4 浮点型	41
位置,也需要指定目标文件		2.4.5 Java 7 新增的数值中使用	
的文件名,这里使用 javac		下划线分隔	43
编译 Java 程序时怎么不需		2.4.6 布尔型	43
要指定目标文件的文件名		2.5 基本类型的类型转换	44
呢?	14	2.5.1 自动类型转换	44
1.5.3 运行 Java 程序	15	2.5.2 强制类型转换	45
1.5.4 根据 CLASSPATH 环境变量		2.5.3 表达式类型的自动提升	47
定位类	16	2.6 直接量	48
1.6 Java 程序的基本规则	17	2.6.1 直接量的类型	48
1.6.1 Java 程序的组织形式	17	2.6.2 直接量的赋值	48
1.6.2 Java 源文件的命名规则	18	2.7 运算符	49

2.7.1	算术运算符	50
2.7.2	赋值运算符	52
2.7.3	位运算符	53
2.7.4	扩展后的赋值运算符	56
2.7.5	比较运算符	57
2.7.6	逻辑运算符	58
2.7.7	三目运算符	59
2.7.8	运算符的结合性和优先级	59
2.8	本章小结	61
	本章练习	61
第3章 流程控制与数组 62		
3.1	顺序结构	63
3.2	分支结构	63
3.2.1	if 条件语句	63
3.2.2	Java 7 的 switch 分支语句	67
3.3	循环结构	70
3.3.1	while 循环语句	70
3.3.2	do while 循环语句	71
3.3.3	for 循环	72
3.3.4	嵌套循环	75
3.4	控制循环结构	76
3.4.1	使用 break 结束循环	76
3.4.2	使用 continue 结束本次循环	78
3.4.3	使用 return 结束方法	79
3.5	数组类型	80
3.5.1	理解数组：数组也是一种类型	80
	int[] 是一种类型吗？怎么用这种类型呢？	80
3.5.2	定义数组	80
3.5.3	数组的初始化	81
	能不能只分配内存空间，不赋初始值呢？	81
3.5.4	使用数组	83
	为什么要我记住这些异常信息？	84
3.5.5	foreach 循环	84
3.6	深入数组	86
3.6.1	内存中的数组	86


	为什么有栈内存和堆内存之分？	87
3.6.2	基本类型数组的初始化	89
3.6.3	引用类型数组的初始化	90
3.6.4	没有多维数组	93
	我是否可以让图 3.13 中灰色覆盖的数组元素再次指向另一个数组？这样不就可以扩展成三维数组吗？甚至扩展成更多维的数组？	95
3.6.5	操作数组的工具类	96
3.6.6	数组的应用举例	97
3.7	本章小结	101
	本章练习	101
第4章 面向对象（上） 102		
4.1	类和对象	103
4.1.1	定义类	103
	构造器不是没有返回值吗？为什么不能用 void 修饰呢？	106
4.1.2	对象的产生和使用	106
4.1.3	对象、引用和指针	107
4.1.4	对象的 this 引用	108
4.2	方法详解	113
4.2.1	方法的所属性	113
4.2.2	方法的参数传递机制	114
4.2.3	形参个数可变的方法	117
4.2.4	递归方法	119
4.2.5	方法重载	120
	为什么方法的返回值类型不能用于区分重载的方法？	121
4.3	成员变量和局部变量	122
4.3.1	成员变量和局部变量	122
4.3.2	成员变量的初始化和内存中的运行机制	126
4.3.3	局部变量的初始化和内存中的运行机制	128
4.3.4	变量的使用规则	128

4.4	隐藏和封装	130	本章练习	167
4.4.1	理解封装	130	第 5 章 面向对象 (下)	168
4.4.2	使用访问控制符	130	5.1 Java 7 增强的包装类	169
4.4.3	package、import 和 import static	134	 Java 为什么要对这些数据进行缓存呢?	173
4.4.4	Java 的常用包	139	5.2 处理对象	173
4.5	深入构造器	140	5.2.1 打印对象和 toString 方法	174
4.5.1	使用构造器执行初始化	140	5.2.2 == 和 equals 方法	175
	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	 判断 obj 是否为 Person 类的 实例时, 为何不用 obj instanceof Person 来判断 呢?	179
4.5.2	构造器重载	141	5.3 类成员	180
	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	5.3.1 理解类成员	180
4.6	类的继承	143	5.3.2 单例 (Singleton) 类	181
4.6.1	继承的特点	144	5.4 final 修饰符	182
4.6.2	重写父类的方法	145	5.4.1 final 成员变量	182
4.6.3	super 限定	146	5.4.2 final 局部变量	184
4.6.4	调用父类构造器	149	5.4.3 final 修饰基本类型变量和 引用类型变量的区别	185
	为什么我创建 Java 对象时 从未感觉到 java.lang.Object 类的构造器被调用过?	152	5.4.4 可执行“宏替换”的 final 变量	186
4.7	多态	152	5.4.5 final 方法	188
4.7.1	多态性	152	5.4.6 final 类	189
4.7.2	引用变量的强制类型转换	154	5.4.7 不可变类	189
4.7.3	instanceof 运算符	155	5.4.8 缓存实例的不可变类	193
4.8	继承与组合	156	5.5 抽象类	195
4.8.1	使用继承的注意点	156	5.5.1 抽象方法和抽象类	195
4.8.2	利用组合实现复用	158	5.5.2 抽象类的作用	198
	使用组合关系来实现复用 时, 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?	161	5.6 更彻底的抽象: 接口	200
4.9	初始化块	161	5.6.1 接口的概念	200
4.9.1	使用初始化块	161	5.6.2 接口的定义	201
4.9.2	初始化块和构造器	163	5.6.3 接口的继承	202
4.9.3	静态初始化块	164	5.6.4 使用接口	203
4.10	本章小结	167	5.6.5 接口和抽象类	205
			5.6.6 面向接口编程	206
			5.7 内部类	210
			5.7.1 非静态内部类	210

	学生提问 非静态内部类对象和外部类对象的关系是怎样的? 214			
5.7.2	静态内部类 215			
	学生提问 为什么静态内部类的实例方法也不能访问外部类的实例属性呢? 216			
	学生提问 接口里是否能定义内部接口? 217			
5.7.3	使用内部类 217			
	学生提问 既然内部类是外部类的成员,那么是否可以对外部类定义子类,在子类中再定义一个内部类来重写其父类中的内部类? 220			
5.7.4	局部内部类 220			
5.7.5	匿名内部类 221			
5.7.6	闭包 (Closure) 和回调 224			
5.8	枚举类 227			
5.8.1	手动实现枚举类 227			
5.8.2	枚举类入门 229			
5.8.3	枚举类的 Field、方法和构造器 230			
5.8.4	实现接口的枚举类 233			
	学生提问 枚举类不是用 final 修饰了吗?怎么还能派生子类呢? 234			
5.8.5	包含抽象方法的枚举类 234			
5.9	对象与垃圾回收 236			
5.9.1	对象在内存中的状态 236			
5.9.2	强制垃圾回收 237			
5.9.3	finalize 方法 238			
5.9.4	对象的软、弱和虚引用 240			
5.10	修饰符的适用范围 244			
5.11	使用 JAR 文件 245			
5.11.1	jar 命令详解 246			
5.11.2	创建可执行的 JAR 包 248			
5.11.3	关于 JAR 包的技巧 249			
5.12	本章小结 250			
	本章练习 250			
第 6 章	Java 集合 251			
6.1	Java 集合概述 252			
6.2	Collection 和 Iterator 接口 254			
6.2.1	使用 Iterator 接口遍历集合元素 256			
6.2.2	使用 foreach 循环遍历集合元素 258			
6.3	Set 集合 258			
6.3.1	HashSet 类 259			
	学生提问 hashCode()方法对于 HashSet 是不是十分重要? 261			
6.3.2	LinkedHashSet 类 264			
6.3.3	TreeSet 类 264			
6.3.4	EnumSet 类 271			
6.3.5	各 Set 实现类的性能分析 273			
6.4	List 集合 274			
6.4.1	List 接口和 ListIterator 接口 274			
6.4.2	ArrayList 和 Vector 实现类 277			
6.4.3	固定长度的 List 278			
6.5	Queue 集合 279			
6.5.1	PriorityQueue 实现类 280			
6.5.2	Deque 接口与 ArrayDeque 实现类 280			
6.5.3	LinkedList 实现类 282			
6.5.4	各种线性表的性能分析 283			
6.6	Map 285			
6.6.1	HashMap 和 Hashtable 实现类 286			
6.6.2	LinkedHashMap 实现类 290			
6.6.3	使用 Properties 读写属性文件 290			
6.6.4	SortedMap 接口和 TreeMap 实现类 292			
6.6.5	WeakHashMap 实现类 294			
6.6.6	IdentityHashMap 实现类 295			
6.6.7	EnumMap 实现类 296			
6.6.8	各 Map 实现类的性能分析 297			
6.7	HashSet 和 HashMap 的性能选项 297			

6.8	操作集合的工具类: Collections	298	8.2.1	使用 try...catch 捕获异常	335
6.8.1	排序操作	298	8.2.2	异常类的继承体系	336
6.8.2	查找、替换操作	302	8.2.3	Java 7 提供的多异常捕获	339
6.8.3	同步控制	303	8.2.4	访问异常信息	340
6.8.4	设置不可变集合	303	8.2.5	使用 finally 回收资源	341
6.9	烦琐的接口: Enumeration	304	8.2.6	异常处理的嵌套	344
6.10	本章小结	305	8.2.7	Java 7 的自动关闭资源的 try 语句	344
	本章练习	305	8.3	Checked 异常和 Runtime 异常 体系	345
第 7 章	泛型	306	8.3.1	使用 throws 声明抛出异常	346
7.1	泛型入门	307	8.4	使用 throw 抛出异常	348
7.1.1	编译时不检查类型的异常	307	8.4.1	抛出异常	348
7.1.2	手动实现编译时检查类型	308	8.4.2	自定义异常类	350
7.1.3	使用泛型	309	8.4.3	catch 和 throw 同时使用	350
7.1.4	Java 7 泛型的“菱形” 语法	309	8.4.4	Java 7 增强的 throw 语句	352
7.2	深入泛型	310	8.4.5	异常链	353
7.2.1	定义泛型接口、类	311	8.5	Java 的异常跟踪栈	354
7.2.2	从泛型类派生子类	312	8.6	异常处理规则	357
7.2.3	并不存在泛型类	314	8.6.1	不要过度使用异常	357
7.3	类型通配符	314	8.6.2	不要使用过于庞大的 try 块	358
7.3.1	使用类型通配符	316	8.6.3	避免使用 Catch All 语句	358
7.3.2	设定类型通配符的上限	317	8.6.4	不要忽略捕获到的异常	359
7.3.3	设定类型形参的上限	319	8.7	本章小结	359
7.4	泛型方法	320		本章练习	359
7.4.1	定义泛型方法	320	第 9 章	Annotation (注释)	360
7.4.2	泛型方法和类型通配符的 区别	323	9.1	基本 Annotation	361
7.4.3	Java 7 的“菱形”语法与 泛型构造器	324	9.1.1	限定重写父类方法: @Override	361
7.4.4	设定通配符下限	325	9.1.2	标示已过时: @Deprecated	362
7.4.5	泛型方法与方法重载	327	9.1.3	抑制编译器警告: @SuppressWarnings	363
7.5	擦除和转换	328	9.1.4	Java 7 的“堆污染”警告 与@SafeVarargs	363
7.6	泛型与数组	329	9.2	JDK 的元 Annotation	365
7.7	本章小结	331	9.2.1	使用@Retention	365
第 8 章	异常处理	332	9.2.2	使用@Target	366
8.1	异常概述	333			
8.2	异常处理机制	335			

9.2.3	使用@Documented	366	10.9.4	字符集和 Charset	429
9.2.4	使用@Inherited	367	 二进制序列与字符之间如何对应呢?	430	
9.3	自定义 Annotation	368	10.9.5	文件锁	432
9.3.1	定义 Annotation	368	10.10	Java 7 的 NIO.2	433
9.3.2	提取 Annotation 信息	370	10.10.1	Path、Paths 和 Files 核心 API	433
9.3.3	使用 Annotation 的示例	371	10.10.2	使用 FileVisitor 遍历 文件和目录	435
9.4	编译时处理 Annotation	376	10.10.3	使用 WatchService 监控文件变化	436
9.5	本章小结	380	10.10.4	访问文件属性	437
第 10 章	输入/输出	381	10.11	本章小结	439
10.1	File 类	382	本章练习	440	
10.1.1	访问文件和目录	382	第 11 章	多线程	441
10.1.2	文件过滤器	384	11.1	线程概述	442
10.2	理解 Java 的 IO 流	385	11.1.1	线程和进程	442
10.2.1	流的分类	386	11.1.2	多线程的优势	443
10.2.2	流的概念模型	387	11.2	线程的创建和启动	444
10.3	字节流和字符流	388	11.2.1	继承 Thread 类创建 线程类	444
10.3.1	InputStream 和 Reader	388	11.2.2	实现 Runnable 接口 创建线程类	446
10.3.2	OutputStream 和 Writer	391	11.2.3	使用 Callable 和 Future 创建线程	447
10.4	输入/输出流体系	392	11.2.4	创建线程的三种方式对比	449
10.4.1	处理流的用法	392	11.3	线程的生命周期	450
10.4.2	输入/输出流体系	393	11.3.1	新建和就绪状态	450
10.4.3	转换流	396	11.3.2	运行和阻塞状态	452
 怎么没有把字符流转换成字节流的转换流呢?	396	11.3.3	线程死亡	453	
10.4.4	推回输入流	397	11.4	控制线程	454
10.5	重定向标准输入/输出	399	11.4.1	join 线程	454
10.6	Java 虚拟机读写其他进程的 数据	400	11.4.2	后台线程	455
10.7	RandomAccessFile	402	11.4.3	线程睡眠: sleep	456
10.8	对象序列化	406	11.4.4	线程让步: yield	457
10.8.1	序列化的含义和意义	406	11.4.5	改变线程优先级	459
10.8.2	使用对象流实现序列化	407	11.5	线程同步	460
10.8.3	对象引用的序列化	409	11.5.1	线程安全问题	460
10.8.4	自定义序列化	413	11.5.2	同步代码块	462
10.8.5	另一种自定义序列化机制	419	11.5.3	同步方法	464
10.8.6	版本	420			
10.9	NIO	421			
10.9.1	Java 新 IO 概述	422			
10.9.2	使用 Buffer	422			
10.9.3	使用 Channel	426			

11.5.4	释放同步监视器的锁定	466		
11.5.5	同步锁 (Lock)	467		
11.5.6	死锁	469		
11.6	线程通信	471		
11.6.1	传统的线程通信	471		
11.6.2	使用 Condition 控制 线程通信	475		
11.6.3	使用阻塞队列 (BlockingQueue) 控制线程通信	477		
11.7	线程组和未处理的异常	481		
11.8	线程池	483		
11.8.1	Java 5 实现的线程池	484		
11.8.2	Java 7 新增的 ForkJoinPool	486		
11.9	线程相关类	489		
11.9.1	ThreadLocal 类	490		
11.9.2	包装线程不安全的集合	492		
11.9.3	线程安全的集合类	492		
11.10	本章小结	494		
第 12 章	网络编程	495		
12.1	网络编程的基础知识	496		
12.1.1	网络基础知识	496		
12.1.2	IP 地址和端口号	497		
12.2	Java 的基本网络支持	498		
12.2.1	使用 InetAddress	499		
12.2.2	使用 URLDecoder 和 URLEncoder	499		
12.2.3	使用 URL 和 URLConnection	501		
12.3	基于 TCP 协议的网络编程	508		
12.3.1	TCP 协议基础	508		
12.3.2	使用 ServerSocket 创建 TCP 服务器端	509		
12.3.3	使用 Socket 进行通信	510		
12.3.4	加入多线程	512		
12.3.5	记录用户信息	516		
12.3.6	半关闭的 Socket	524		
12.3.7	使用 NIO 实现非阻塞 Socket 通信	525		
12.3.8	使用 Java 7 的 AIO 实现 非阻塞通信	531		
	 上面程序中好像没用到④ ⑤号代码的 get()方法的返 回值, 这两个地方不调用 get()方法行吗?	535		
12.4	本章小结	539		
	本章练习	539		
第 13 章	类加载机制与反射	540		
13.1	类的加载、连接和初始化	541		
13.1.1	JVM 和类	541		
13.1.2	类的加载	542		
13.1.3	类的连接	543		
13.1.4	类的初始化	543		
13.1.5	类初始化的时机	544		
13.2	类加载器	546		
13.2.1	类加载器简介	546		
13.2.2	类加载机制	547		
13.2.3	创建并使用自定义的类 加载器	549		
13.2.4	URLClassLoader 类	553		
13.3	通过反射查看类信息	554		
13.3.1	获得 Class 对象	554		
13.3.2	从 Class 中获取信息	555		
13.4	使用反射生成并操作对象	559		
13.4.1	创建对象	559		
13.4.2	调用方法	561		
13.4.3	访问属性值	564		
13.4.4	操作数组	565		
13.5	使用反射生成 JDK 动态代理	566		
13.5.1	使用 Proxy 和 InvocationHandler 创建动态代理	567		
13.5.2	动态代理和 AOP	569		
13.6	反射和泛型	573		
13.6.1	泛型和 Class 类	573		
13.6.2	使用反射来获取泛型信息	575		
13.7	本章小结	577		
	本章练习	577		

第 1 章

Java 语言概述

本章要点

- Java 语言的发展简史
- Java 语言各竞争对手的情况
- 编译型语言和解释型语言
- Java 语言的编译、解释运行机制
- 通过 JVM 实现跨平台
- 安装 JDK
- 设置 PATH 环境变量
- 编写、运行 Java 程序
- Java 程序的组织形式
- Java 程序的命名规则
- 初学者易犯的错误
- Java 的垃圾回收机制