



普通高等教育“十一五”国家级规划教材

高等学校计算机程序设计课程系列教材

Java程序设计(第2版)

刘宝林 主编

胡 博 谢锋波 编



高等教育出版社
HIGHER EDUCATION PRESS

普通高等教育“十一五”国家级规划教材
高等学校计算机程序设计课程系列教材

Java 程序设计

Java Chengxu Sheji

(第2版)

刘宝林 主 编
胡 博 谢锋波 编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书为普通高等教育“十一五”国家级规划教材。

本书全面详细地介绍了 Java 程序设计的基本知识、方法与技巧。全书共分 9 章,第 1 章~第 6 章主要介绍 Java 语言的基础知识和面向对象的编程思想,其中包括 Java 的编程环境、Java 语言基础、类与对象、异常处理、基础类库与工具类库、输入/输出系统及文件操作。第 7 章~第 9 章主要介绍 Java 的多线程机制、图形用户界面及事件处理、网络编程等。附录中对 Java 编码规范作了简单的介绍。书中所有示例均在 J2SDK1.6.0 平台下编译通过。

本书既注重理论介绍,又强调实际应用,注重提高读者利用面向对象技术和 Java 语言解决实际问题的能力。本书可作为高等学校计算机相关课程的教材,也可作为各类成人教育学院 Java 程序设计课程、计算机水平考试和基础培训的教材,也可供同等程度的读者及计算机应用开发人员自学使用。

图书在版编目(CIP)数据

Java 程序设计 / 刘宝林主编; 胡博, 谢锋波编. —2 版. —北京:

高等教育出版社, 2011.6

普通高等教育“十一五”国家级规划教材

ISBN 978-7-04-031908-8

I. ①J… II. ①刘… ②胡… ③谢… III. ①Java 语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 083669 号

策划编辑	时 阳	责任编辑	时 阳	封面设计	于文燕	版式设计	范晓红
责任校对	金 辉	责任印制	张泽业				

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.landaco.com
印 刷	北京地质印刷厂		http://www.landaco.com.cn
开 本	787 × 1092 1/16		
印 张	20.5	版 次	2004 年 11 月第 1 版
字 数	500 000		2011 年 6 月第 2 版
购书热线	010-58581118	印 次	2011 年 6 月第 1 次印刷
咨询电话	400-810-0598	定 价	28.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 31908-00

主 编 简 介



刘宝林 博士，清华大学计算系副教授。长期从事计算机教学、视听觉信息认知计算和多媒体智能信息处理方面的研究工作。先后主持、承担国家自然科学基金面上项目、国家自然科学基金重点项目、863 计划重点项目、973 项目子课题等项目多项。近年来，发表 SCI、EI 论文近 30 篇。主编普通高等教育“十五”国家级规划教材、北京市精品教材《Java 程序设计与案例》及配套参考教材《Java 程序设计与案例习题解答与实验指导》各一部；主编普通高等教育“十一五”国家级规划教材《Java 程序设计（第 2 版）》和《Java 程序设计案例教程（第 2 版）》各一部。

前 言

本书为普通高等教育“十一五”国家级规划教材，可与刘宝林主编、高等教育出版社出版的普通高等教育“十一五”国家级规划教材《Java 程序设计案例教程（第2版）》配套使用。

计算机程序设计是高等学校各专业的一门核心课程。随着 Internet 的迅猛发展，Java 语言作为一种面向对象的程序设计语言，以其面向对象、简单易用、结构中立、与平台无关、稳固性、安全性、多线程以及强大的网络编程功能，为用户提供了良好的程序设计环境，成为最具前途的编程语言之一。

本书以非计算机专业学生和其他初学者为对象，针对 Java 编程的初学者编写。根据计算机程序设计课程的基本要求和作者多年来的教学经验和体会，从最基础的编程讲起，全面详细地介绍了 Java 程序设计的基本知识、方法与技巧，做到了由浅入深、循序渐进、深入浅出、理论与实际相结合。在教学安排上注重编程能力的培养，做到举一反三，打好程序设计的基本功。

本书内容通俗易懂、简单明了、实例丰富，书中所列程序易于读者理解和掌握，读者通过边学边练，在有限的学时内，可掌握面向对象程序设计的基本知识、基本方法和技巧。

本书共分 9 章，第 1 章~第 6 章主要介绍 Java 语言的基础知识和面向对象的编程思想，其中包括 Java 的编程环境、Java 语言基础、类与对象、异常处理、基础类库与工具类库、输入/输出系统及文件操作。第 7 章~第 9 章主要介绍 Java 的多线程机制、图形用户界面及事件处理、网络编程。书中每章都配有学习目标和小结，便于读者有的放矢，总结提高。

本书既注重理论介绍，又强调实际应用，注重提高读者利用面向对象技术和 Java 语言解决实际问题的能力。

本书由刘宝林主编。胡博编写第 1 章~第 7 章，谢锋波编写第 8 章~第 9 章，王忠宁、伍广宁、孟显耀参加了本书程序的调试，全书最后由刘宝林统一修改定稿。特别感谢王行言教授认真审阅了全书，并提出了许多宝贵的建议。

由于作者水平所限，书中难免有欠妥之处，恳请专家、读者批评指正。

刘宝林

2011 年 3 月于清华园

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010) 58581897 58582371 58581879

反盗版举报传真 (010) 82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第 1 章 Java 概述 1	3.1.2 Java 与面向对象 33
1.1 Java 语言简介 1	3.2 类与对象 33
1.1.1 Java 语言简史 1	3.2.1 类的定义 33
1.1.2 Java 语言、Java 虚拟机 和 Java 平台 2	3.2.2 域和方法 33
1.1.3 Java 语言的主要特点 2	3.2.3 构造方法与对象的创建 34
1.2 Java 语言的应用 3	3.2.4 对象的使用 36
1.3 Java 开发环境设置及开发工具介绍 4	3.3 对象的清除与垃圾回收 37
1.3.1 开发环境设置 4	3.3.1 垃圾回收的机制 37
1.3.2 开发工具介绍 5	3.3.2 finalize()方法 38
1.3.3 Java 程序的编译和运行 5	3.4 变量与变量的传递 39
1.4 Java 程序示例 6	3.4.1 基本数据类型变量和引用型变量 39
1.4.1 Application 程序示例 6	3.4.2 变量的作用域 39
1.4.2 Applet 程序示例 7	3.4.3 变量的传递与返回 40
小结 8	3.5 访问控制符 42
习题 8	3.5.1 成员访问控制 42
第 2 章 Java 语言基础 9	3.5.2 类访问控制 45
2.1 Java 语法基础 9	3.6 非访问控制符和其他修饰符 45
2.1.1 变量、常量和数据类型 9	3.6.1 非访问控制符 45
2.1.2 运算符和表达式 13	3.6.2 其他修饰符 49
2.2 流程控制 18	3.7 接口 50
2.3 数组 25	3.7.1 接口的定义 50
2.3.1 一维数组 25	3.7.2 接口的实现 50
2.3.2 多维数组 26	3.7.3 接口的应用 53
2.3.3 对象数组 27	3.7.4 接口与抽象类 53
2.4 Java 的一般程序结构 29	3.8 包 60
小结 30	3.8.1 package 与 import 60
习题 31	3.8.2 CLASSPATH 61
第 3 章 类与对象 32	3.9 继承 62
3.1 面向对象程序设计 32	3.9.1 概念和语法 62
3.1.1 面向对象的程序设计方法概述 32	3.9.2 隐藏和覆盖 66
	3.9.3 构造方法的重载和继承 71

3.9.4 this 和 super	77	5.3.5 Map	135
3.10 组合	80	5.3.6 集合数据遍历	137
3.10.1 组合的概念	80	5.3.7 排序和查找	138
3.10.2 继承和组合	82	5.4 日期和时间	142
3.11 多态	86	5.4.1 Date	142
3.11.1 多态的概念	86	5.4.2 Calendar	143
3.11.2 多态的应用	86	小结	146
3.11.3 构造方法与多态	89	习题	146
3.11.4 对象的初始化过程	91	第6章 Java 输入/输出系统	147
3.12 内部类与匿名类	95	6.1 流	147
小结	101	6.1.1 InputStream	147
习题	101	6.1.2 OutputStream	148
第4章 异常处理	104	6.1.3 FilterInputStream 和 FilterOutputStream	149
4.1 异常处理	104	6.1.4 Reader	150
4.1.1 异常处理机制	104	6.1.5 Writer	151
4.1.2 异常的捕获与处理	108	6.1.6 常见的输入/输出流	152
4.1.3 finally	110	6.2 标准输入/输出	153
4.2 自定义异常类	112	6.2.1 标准输入/输出	153
4.3 异常规范	115	6.2.2 重定向标准输入/输出	154
4.4 标准异常介绍	115	6.3 文件访问	154
小结	115	6.3.1 File	154
习题	116	6.3.2 文件读/写	160
第5章 基础类库和工具类库	118	6.3.3 RandomAccessFile	167
5.1 Java 基础类	118	6.3.4 压缩文件读/写	168
5.1.1 基础类库	118	6.4 对象的可串行化	171
5.1.2 Object	119	6.4.1 Serializable 接口	171
5.1.3 基本数据类型的封装	123	6.4.2 Externalizable 接口和 transient 关键字	174
5.1.4 System	124	6.5 NIO	176
5.1.5 Math	124	6.5.1 NIO 介绍	176
5.2 字符串相关类	126	6.5.2 Buffer	176
5.2.1 String	126	6.5.3 Channel	179
5.2.2 StringBuffer	128	6.5.4 Selector	180
5.2.3 StringTokenizer	129	6.5.5 charset	180
5.3 集合	130	小结	182
5.3.1 集合 API	130	习题	182
5.3.2 Set	130	第7章 多线程	184
5.3.3 List	132	7.1 线程	184
5.3.4 Iterator	133		

7.1.1	Java 中的线程	184	8.4.7	滚动窗格	255
7.1.2	线程对象的创建	185	8.4.8	树	257
7.1.3	多线程	187	8.4.9	表格	261
7.2	线程调度	188	8.5	多媒体	262
7.2.1	线程状态和生命周期	188	8.5.1	图像的使用	262
7.2.2	线程优先级	189	8.5.2	动画的实现	265
7.2.3	线程的基本控制	190	8.5.3	声音的使用	270
7.2.4	线程的同步	193	8.6	Applet	270
7.2.5	线程的死锁	196	8.6.1	Applet 工作原理	271
7.2.6	线程间的同步控制	197	8.6.2	Applet 类	273
小结		203	8.6.3	Applet 的使用局限	277
习题		203	小结		277
第 8 章	图形用户界面	206	习题		277
8.1	概述	206	第 9 章	网络编程	279
8.1.1	AWT 和 Swing	206	9.1	HTTP 的基本概念	279
8.1.2	窗口基本原理	207	9.1.1	URL	280
8.2	布局管理	208	9.1.2	URLConnection	281
8.2.1	布局管理简介	208	9.2	Socket	284
8.2.2	常用布局管理器	209	9.2.1	java.net.Socket	284
8.3	事件处理	222	9.2.2	java.net.ServerSocket	286
8.3.1	事件响应模型	222	9.2.3	简单的服务器和客户机程序	287
8.3.2	事件监听器接口	224	9.2.4	多线程服务实现	291
8.3.3	事件适配器 Adapter 类	226	9.3	数据报	296
8.3.4	使用授权事件模型	229	9.4	非阻塞的 Socket	299
8.3.5	用内部类实现事件处理	237	9.4.1	Selector	299
8.4	Swing 组件	240	9.4.2	SocketChannel 和 ServerSocketChannel	300
8.4.1	Swing 包	240	小结		304
8.4.2	图标和标签	240	习题		304
8.4.3	文本域	243	附录		305
8.4.4	按钮	244	参考文献		318
8.4.5	组合框	251			
8.4.6	选项窗格	253			

第 1 章 Java 概述

学习目标

- (1) 了解 Java 语言的发展历史和 Java 平台的应用划分。
- (2) 了解 Java 语言的特点及其实际应用领域。
- (3) 了解 Java 开发环境及开发工具, 掌握如何进行开发环境设置及如何编译和运行 Java 程序。
- (4) 熟悉几种不同类型的 Java 程序。

1.1 Java 语言简介

1.1.1 Java 语言简史

Java 语言是由美国 Sun 公司创造和发展的一门完全面向对象的程序语言。由于 Java 语言独特的运行机制, 对网络编程支持的便利, 方便的平台可移植能力及其他优点, 使得 Java 语言从诞生到现在, 在不长的时间内, 得到迅速地推广。据统计, 目前全世界的 Java 开发人员已达到 1 000 多万。

Java 语言的历史最早可以追溯到 1991 年, 当时 Sun 公司启动了绿色计划项目 (Green Project), 着手发展消费性电子产品以及其上的软件开发, 开发小组主要成员是 Patrick Naughton、James Gosling 和 Mike Sheridan 等, 项目开发的时候采用的语言是 C/C++ 和小组内自己研发的语言 Oak (取名自 Gosling 办公室窗外的一棵橡树), 由于商标的冲突, Oak 随后被改称 Java。虽然绿色项目本身因为各种原因没有得到继续发展, 但 Java 却没有随之被埋没。

1994 年, 万维网得到飞速的发展, Gosling 认为市场需要一种不依赖实际硬件和软件环境, 安全可靠、可交互的浏览器, 于是启动了使用 Java 语言开发新的 Web 浏览器的项目。1994 年秋天, WebRunner 开发成功; 1995 年 5 月 23 日, WebRunner 被正式改名为 HotJava 发表后, 引起了业界的轰动, Java 语言也随之得以推广, 并奠定了其在编程语言大家族中的地位。

与 C/C++ 相比, Java 语言给开发人员提供了更为简洁的语法; 取消指针虽然损失了一些编程的灵活性, 但带来的却是更高的代码质量; 完全的面向对象使得开发人员从设计开始就必须采用面向对象的软件设计方法; 独特的运行机制使得 Java 语言有着天然的可移植性。这些优点使得 Java 迅速被开发人员接受, 同时也促进了 Java 自身的发展。

1.1.2 Java 语言、Java 虚拟机和 Java 平台

在详细讲述 Java 语言之前，很重要是正确的区分和理解 Java 语言、Java 虚拟机和 Java 平台三者间的关系。Java 语言就是编写 Java 应用程序时所用的编程语言，是一种面向对象的编程语言，语法和 C 语言类似。设计者在保证 Java 语言强大的同时，竭力避免使 Java 语言和其他面向对象语言（如 C++）一样出现过于复杂的语法，而且对开发人员而言，使用 Java 语言编写强鲁棒性的代码也同样方便快捷。出色的设计，使得 Java 语言广受赞誉。

Java 语言的源代码编译后，得到的中间结果称为字节码。字节码可以在一种机器语言解释环境上解释执行，这个类似 CPU 结构的运行环境称为 Java 虚拟机（简称为 JVM）或 Java 解释器。Java 语言的初始设计目标，就是保证 Java 程序可以在所有拥有 Java 虚拟机的环境中运行。目前大部分常见的操作系统上都拥有相应的 Java 虚拟机，如 Windows、Solaris、MacOS 等，除了这些个人计算机系统，Java 虚拟机也存在于诸如电视机顶盒、移动外设等领域。

和前两者不同，Java 平台是指 Java 类库的集合。有的时候，Java 平台也和 Java 运行时环境和核心 Java APIs 联系在一起。所有的 Java 程序都是建立在一些预定义的 Java 类库基础之上的。

1.1.3 Java 语言的主要特点

由于 Java 语言独特的设计思想，使得 Java 语言拥有大多数其他编程语言所没有的优点。

简单

Java 语言语法的简单使得编写 Java 程序更为容易，不需要长时间的培训就可满足一般的需求。

面向对象

Java 语言是完全的面向对象语言，面向对象的设计思想始终贯穿在 Java 开发过程中。

平台无关性

Sun 公司把“Write once, run anywhere”定义为 Java 的核心特点，独特的运行机制使得大多数 Java 程序拥有二进制级的可移植性。使用 Java，开发人员可以迅速地编写与平台无关的应用程序，大大降低了软件项目开发、维护和管理的成本。

安全性

Java 语言独特的设计是它安全性的保证。从语言一级来看，Java 语言摒弃了 C/C++ 中容易造成安全漏洞的指针，提供了自动的内存管理机制，保证了 Java 程序运行的可靠性；在运行架构上，Java 语言拥有一个设计优良的安全模型，使未经授权的代码不能进行危害用户本地资源的操作。

高效的网络编程

Java 语言是面向网络的，这和 Sun 公司“网络就是计算机”的观点是一致的。Java 平台中存在大量用于开发网络应用程序的类库。基于这些类库，应用程序可以很方便地获取网络资源，快捷地编写基于网络的各种应用程序。

分布式

优秀高效的网络编程使得基于 Java 的分布式开发非常简单。

多线程

Java 语言是现在为数不多的内部支持多线程的编程语言。

方便的国际化功能

Java 语言从设计开始就把国际化作为核心目标。大多数其他编程语言还在采用 8-bit 的字符集的时候，Java 语言已经在使用 16-bit 的 Unicode 编码。因此用 Java 开发国际化的应用程序更为便捷。

开发的高效

对于开发人员而言，基于 Java 的软件项目可以大大节省开发时间，使得软件开发可以更迅速地配合市场的发展。

1.2 Java 语言的应用

1999 年 6 月 15 日，Sun 公司宣布将 Java 平台划分成 J2SE、J2EE 和 J2ME，Java 技术被按照具体的应用领域进行了划分，这三个平台分别着眼于桌面应用程序开发、企业应用开发和移动设备开发。如表 1.1 所示是目前 Java 提供的各种技术名称及其应用领域（其中 J2EE 涉及的技术过于庞杂，这里只列出其中一部分）。

表 1.1 Java 主要技术及其应用领域列表

J2SE 平台	
名 称	简 介
JavaBeans	一种全新的组件体系结构
JavaDoc	从源文件注释生成文档的工具
JFC/Swing	一套构建 GUI 程序的基础类库
HotSpot 虚拟机	高效的 Java 虚拟机
JPDA	Java Platform Debugger Architecture, Java 平台的 debugging 支持实现
Java Web Start	一种全新的 Java 应用程序的发布形式
JDBC	定义 Java 应用程序与关系型数据库连接的规范
RMI	远程方法调用，用于分布式开发
Java Security	Java 的安全体系结构
JAI	一组功能强大的图像处理类库
JAAS	用户认证服务
JCA	与诸如传真机等外设通信的规范
JCE	加密规范
JDO	对象持久化规范
JavaMail	提供整套的 E-mail 应用支持类库
JMX	Java 管理扩展类库
JMF	对多媒体播放提供支持的类库
JNDI	对目录服务提供支持的类库
Java 3D	应用 Java 进行 3D 程序开发

续表

J2EE 平台	
JAXP、JAXR、JAX-RPC 等	与 XML 应用相关的 Java 实现
SAAJ	提供 SOAP 支持
Java IDL	使 Java 支持 CORBA
JavaMail	同 J2SE
JMS	Java 消息服务
JSP 和 Servlets	实现动态 Web 服务
JDBC	同 J2SE
EJB	企业 JavaBeans 技术
J2EE CC/PP	提供对其他网络终端的支持
J2EE Connector	提供应用服务器和 EIS 系统的通信标准
J2ME 平台	
CLDC	对诸如移动电话等小型设备的支持规范与实现
J2ME WTK	J2ME 的开发包
CDC	对一些拥有固定网络的设备提供支持
Java Card	智能卡业务的支持
Java Telephony、Java TV 等	对电视机顶盒等其他设备提供 Java 支持

1.3 Java 开发环境设置及开发工具介绍

1.3.1 开发环境设置

下面介绍 J2SE SDK (Software Development Kit) 的设置。以 Windows 操作系统为例, 首先需要下载 J2SE SDK 的 Windows 版本, 读者可以从 <http://www.oracle.com/technetwork/java/javase> 下载到需要的版本。目前能下载到的正式版本号 1.6.0。

J2SE 是 Java 平台中面向一般应用程序开发的平台, J2SE SDK 通常简称 JDK, 是 J2SE 平台下的软件开发包, 包含了开发一般应用程序需要的编译器、常用的 Java 类库、Java 运行时环境和一些其他 Java 命令行工具等。由于不同操作系统下安装的 JDK 内容有些细微的差别, 为了避免混淆, 以后内容都是以 Windows 下的 JDK 为例说明。

Sun 制定了 Java 的各个标准规范, 也提供了完整的 JDK 下载, 其他厂商也根据规范进行了自己的实现, 例如 IBM JDK、Blackdown JDK 等。不同的 JDK 在性能和提供的类库上有些差异, 但在使用上还是一致的。同样, 将在以后的内容中讲述 Sun 的 JDK。

Sun JDK 推出至今, 经历了从最初的 1.0 到最近的 1.6.0 的几个大的版本更新。其中版本 1.2 的推出是一个里程碑, 自此 Sun 正式使用名称 Java 2。自 1.2 版本起, Swing 开始作为 JDK 的默认类库出现; 目前的 1.6.0 版本除了性能的大幅度提升外, 同时调整了一些标准类库中的 API。对于一般的学习和应用, 推荐使用 1.3 版本以上的 JDK。

对于一个标准的 JDK 安装, 其安装根目录下包括如表 1.2 所示的主要部分。

表 1.2 JDK 安装根目录下包括的主要部分列表

bin	一些命令行工具，包括 Java 编译器的启动命令
demo	一些 Java 的例程
include	编写 JNI 等程序需要的 C 头文件
jre	Java 运行时环境
lib	除去 jre 中包含的类库，JDK 额外需要的一些类库
src.zip	部分 JDK 的源代码

安装好 J2SE SDK 后，为了开发的方便，请按如下步骤设置。

1. 添加环境变量 `JAVA_HOME=%java_installation_dir%`。把 `%java_installation_dir%` 换成实际的 JDK 安装目录路径。
2. 修改环境变量 `PATH`，添加 `%java_installation_dir%/bin`。这样，每次在命令行下不用额外设置就可以直接使用 `bin` 目录下的 Java 命令行工具。
3. 设置环境变量 `CLASSPATH=%JAVA_HOME%/jre/lib/lib.jar;.。` 注意后面的 “`;`”，这说明当前目录也位于 `CLASSPATH` 中。

1.3.2 开发工具介绍

目前已经有很多 Java IDE (Java Integrated Development Environment)，可以大幅度加快 Java 应用的开发速度，下面介绍其中几种。

1. IntelliJ IDEA。JetBrains 公司的商业 Java 开发环境，是目前最优秀的 Java IDE 之一。可以从 <http://www.intellij.com> 网站下载到试用版本。

2. Eclipse。由 IBM 直接支持的开源项目，极为出色的 Java IDE，可以免费使用，使得其更为普及。可以从 <http://www.eclipse.org> 网站下载到最新的版本。

3. JBuilder。Borland 公司的老牌 Java IDE，对企业开发应用集成提供了非常好的支持。可以从 <http://www.borland.com> 网站下载到免费的个人版本。

4. Netbeans 和 Forte。前者是 Sun 支持开发的开源项目，后者则是在前者基础上衍生的商业产品。可以从 <http://www.netbeans.org> 网站下载到 Netbeans 的最新版本。

5. Codeguide。Codeguide 是 Omnicore 公司 (<http://www.omnicore.com>) 推出的优秀的 Java IDE，是商业产品。

6. Emacs JDEE。Emacs 是非常优秀的文本编辑环境，作为 Emacs 的插件，Emacs JDEE 提供了优秀的 Java 开发环境。可以从 <http://jdee.sunsite.dk> 网站下载。

7. JEdit 和 Jext。是两个轻量级的 Java IDE，都是非常优秀的开源项目。可分别从 <http://jedit.sf.net> 和 <http://jext.sf.net> 网站下载相应的版本。

还有其他优秀的 Java IDE，互有优劣，感兴趣的读者可以从 http://www.javaworld.com/channel_content/jw-tools-index.shtml 网站得到更多的信息。

1.3.3 Java 程序的编译和运行

按照 1.3.1 小节的介绍设置 Java 的开发环境后，就可以在命令行下直接使用 `%JAVA_HOME%/bin` 目录下的 JDK 命令。

命令 `javac` 是编译 Java 程序的主命令，使用格式如下：

`javac [ options ] [ sourcefiles ] [ @files ]`

如果要编译少量的程序文件，可以直接在命令行中输入，如果有很多文件需要编译，可以把需要编译的文件名写在一个文件中，不妨假定文件是 `files.lst`，使用 `javac @files.lst` 即可编译。

`javac` 命令还可以使用如下参数：

<code>-classpath</code>	为待编译的程序指定额外的类库
<code>-d</code>	指定编译后的输出路径
<code>-g</code>	在目标代码中生成所有的 Debug 信息
<code>-g:none</code>	不生成任何 Debug 信息
<code>-nowarn</code>	不输出警告信息

假定在 `c:/src` 目录下有要编译的程序 `Test1.java`、`Test2.java` 和 `Test3.java`，希望编译的结果保存到 `c:/class` 目录下，可以使用如下命令：

```
javac -d c:/class c:/src/Test1.java c:/src/Test2.java c:/src/Test3.java
```

或者创建文件 `files.lst`，写入如下内容：

```
c:/src/Test1.java
c:/src/Test2.java
c:/src/Test3.java
```

然后使用命令：

```
javac -d c:/class @files.lst
```

运行 Java 程序，需要使用命令 `java`，使用格式如下：

`java [ options ] class [ argument... ]`

Java 命令支持的参数设置如下：

<code>-cp</code> 和 <code>-classpath</code>	指定运行程序需要的类库
<code>-Dproperty=value</code>	为运行的程序指定额外的属性设置
<code>-verbose:class</code>	输出运行程序加载的类名
<code>-verbose:gc</code>	输出垃圾回收器的运行信息
<code>-version</code>	显示 JDK 的版本信息，然后退出
<code>-showversion</code>	显示 JDK 的版本信息，然后继续运行程序
<code>-help</code>	显示帮助信息

假定要运行的程序是 `c:/class/Main.class`，需要的类在 `c:/lib` 目录下，则在 `c:/class` 目录下运行命令：

```
java -cp c:/lib Main
```

1.4 Java 程序示例

下面是两个典型的 Java 程序范例，演示不同形态的 Java 程序，功能都是输出“Hello World!”。

1.4.1 Application 程序示例

在控制台输出 “Hello World!”，源程序如下。

```

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}

```

命令行下编译：

```
javac HelloWorld.java
```

产生 HelloWorld.class 二进制字节码文件。

命令行下执行：

```
java HelloWorld
```

则在控制台输出 “Hello World!”。

1.4.2 Applet 程序示例

Applet 是 Java 的另外一种应用形式，Applet 通常作为浏览器的嵌入对象存在。下面的代码将在浏览器中显示 “Hello World!” 的字样。

```

import java.applet.*;    // Don't forget these important statements!
import java.awt.*;

```

```

public class FirstApplet extends Applet {
    // This method displays the applet.
    // The Graphics class is how you do all drawing in Java.
    public void paint(Graphics g) {
        g.drawString("Hello World!", 25, 50);
    }
}

```

命令行下编译：

```
javac FirstApplet.java
```

产生 FirstApplet.class 二进制字节码文件。

Applet 小程序不能直接在命令行下执行，需要编写相应的 HTML 文件，把 Applet 嵌入其中。本例编写如下的 FirstApplet.html 文件。

```

<HTML>
<BODY>
<APPLET CODE="FirstApplet.class" HEIGHT=200 WIDTH=300>
</APPLET>
</BODY>
</HTML>

```

命令行下执行：

```
AppletViewer FirstApplet.html
```


或者通过 IE 浏览器、NetScape 浏览器等直接打开 FirstApplet.html 文件。

小结

本章简要叙述了 Java 的发展历史。从发展历史可以看出, Java 从一开始就拥有平台无关的目标和定位, 以及独特的虚拟机系统设计; Java 拥有的一些优良特性, 也使得 Java 无论是在一般应用程序开发, 还是在企业级应用和嵌入式设备应用开发上都拥有很广阔的市场。通过 Java 开发环境和开发工具的介绍, 学会配置开发环境, 编译运行 Java 程序, 并利用开发工具达到快速开发的目的。通过 Java 程序示例, 初步认识了不同的 Java 程序类型。

习 题

1. 安装 J2SE SDK, 熟悉 J2SE SDK 的基本命令。
2. Java 语言的主要特点有哪些?
3. Java 语言主要有哪些方面的应用?
4. 试述 Java 语言、Java 虚拟机和 Java 平台三者之间的关系。
5. 编写一个 Application, 在屏幕上显示如下的信息:

Welcome <你的姓名>

6. 将上题改写为 Applet 小程序。