



全国本科计算机应用创新型人才培养规划教材

Java程序设计教程

主 编 张剑飞

副主编 杜晓昕 金 涛

- 基础知识+经典案例+设计思想+代码实现四位一体的教学模式
- 将Java基础知识的学习与应用能力的提高有机地结合在一起
- 全面介绍J2EE、JSP、Struts、Hibernate、Spring等核心技术



北京大学出版社
PEKING UNIVERSITY PRESS

全国本科计算机应用创新型人才培养规划教材

Java 程序设计教程

主 编 张剑飞
副主编 杜晓昕 金 涛



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书按照基本概念、语句结构、程序思想、代码实现来介绍 Java 语言及涉及的相关技术, 将 Java 语言基础知识和实践应用有机结合起来。全书共分为 13 章, 第 1~8 章为基础部分, 主要内容包括 Java 语言的发展历程与特点、开发环境搭建与上机操作、Java 应用程序与 Java Applet、数据类型、运算符与表达式、控制结构、数组与字符串、对象与类、I/O 流与异常处理、多线程与图形用户界面、网络编程等; 第 9~13 章为提高部分, 主要内容包括 J2EE 运行环境搭建, JSP、Struts、Hibernate 和 Spring 的相关技术。

本书内容翔实、全面, 示例丰富, 有利于学生对照学习, 提高学生的学习效率, 并且所选实例密切联系实际, 力求知识性与实用性相结合。本书主要适用于计算机科学与技术、软件工程专业, 同时也可供相关专业学生学习和参考。

图书在版编目(CIP)数据

Java 程序设计教程/张剑飞主编. —北京: 北京大学出版社, 2010.9

(全国本科计算机应用创新型人才培养规划教材)

ISBN 978-7-301-19388-4

I. ①J… II. ①张… 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 166554 号

书 名: Java 程序设计教程

著作责任者: 张剑飞 主编

策划编辑: 郑 双

责任编辑: 郑 双

标准书号: ISBN 978-7-301-19388-4/TP·1184

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印 刷 者: 三河市富华印装厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16 开本 18.75 印张 430 千字

2011 年 9 月第 1 版 2011 年 9 月第 1 次印刷

定 价: 35.00 元

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

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

序

本套教材经过全国几十所高等学校老师一年多的努力，终于与广大读者见面了。我相信，它一定会受到全国高等学校计算机界老师和同学们的热烈欢迎。

随着信息技术的飞速发展，单一培养模式已经不能满足社会对计算机专业人才多样化的需求。应对这一变化的最佳办法，就是采用多种模式的培养方式。当前，高等学校的计算机教育正处于从过去的单一培养模式向多种培养模式的转变过程中，多种模式的培养方式将是必然的发展方向。

多种模式的培养方式包括：培养人才的类型不同(研究型，应用型)；专业方向不同(计算机软件，计算机网络，信息安全，信息系统，计算机应用技术等)；课程设置的多样性等。

同时，高等教育对科技人才培养的要求是：不但要培养研究型科技人才，还要为国家培养更多的应用型科技人才(或称工程型科技人才)。也就是说，培养应用型科技人才是百分之九十以上的普通高等学校的主要任务。

本套教材正是为适应多种模式培养方式的要求，并且着重于培养计算机领域高级应用型科技人才的需求，而组织编写的。

本套教材具有如下特点。

1. 基础理论够用

计算机专业所需的基础理论知识以够用为准，不是盲目扩张。如数字系统的基础知识，计算机的基本组成原理和体系结构的基础知识，离散数学的基础知识，数据结构和算法的基础知识，操作系统的基础知识，程序设计的基础知识等，都进行了必要的讲解介绍。

2. 强调理论联系实际，学以致用

每本教材的编写都将“理论联系实际，学以致用”的原则贯彻始终。例如，《计算机组成原理和体系结构》结合现代的计算机讲解，使学生学完之后，确切掌握现代计算机的组成、结构和工作原理；又如，《程序设计》结合实例讲解，使学生学完之后，真正能够动手编写程序。

3. 强调教材的配套性

根据多年组织教材的经验，只有配套性好的教材才最受教师和学生们的欢迎。我们这套教材，尽量做到了课堂教材、实训教材和教学课件完全配套，以方便教学使用。

另外，本套教材提供的是一套应用创新型计算机教育系列教材，可供不同类型学校依照自己的教学计划，根据自身的需要进行选用。

现在把这套教材奉献给全国计算机界的朋友们，真诚希望大家能够喜欢。本套教材难免会有诸多缺点或不到之处，还希望得到大家的批评和指正。

全国高等学校计算机教育研究会课程与教材建设委员会主任

李大友

2009年3月

前 言

Java 语言是在 C 和 C++语言的基础上进行简化和改进的一种新型语言，是随着 Internet 以及信息技术的飞速发展而发展起来的。它具有面向对象、与平台无关、安全、稳定、多线程以及强大的网络编程功能，已成为编程者开发的首选工具。目前，国内外的高等教育中，部分学校将 Java 语言列为计算机类本科生及相关专业的第一门编程语言。为了帮助读者更好、更快地学习和掌握 Java，编者编写了本书。通过本书，读者能够对 Java 基本知识、网络编程、网站开发有个系统的学习和掌握。本书也可作为 Java 初学者的首选参考书。

与同类教材相比，本书具有以下特色。

(1) 教学便利。本书采用由基础到应用循序渐进的讲授模式，内容组织和结构合理，层次清晰，每章设有课后习题，方便读者对知识点的巩固，适合读者全面掌握 Java 语言。同时，教师可以利用电子教案、程序源码和习题参考答案等教学资源，可使备课、授课和课后辅导得心应手。

(2) 图文并茂。在基础部分，绝大多数 Java 程序都附有计算机运行后的图示；在提高部分，绝大多数 JSP 页面都附有图示；同时对开发环境的搭建过程和抽象概念的讲解也给出了相应的图示，这些图示可以增进读者对文字和案例的理解。

(3) 案例驱动。在基础部分，每个知识点都附有大量的案例，并且提供了案例的设计过程和详细解释，读者可以按照步骤做出实例，巩固知识点；在提高部分，以实用为原则，通过案例强调基于 Java 的网站开发的全过程，涉及目前网站建设中的关键技术，内容新颖，实用性强，对案例略加改变就可直接移植到其他网站的建设与开发中，可帮助读者学会设计完整的网络系统。

总之，本书集基础知识、案例驱动、技术实用与教学便利于一体，充分体现软件工程的理念，兼顾学习与应用，是一本适合 Java 程序设计初学者、高等院校教学和“卓越工程师”人才培养的教材。

全书在内容体系上共分为 13 章：第 1~8 章为基础部分，第 9~13 章为提高部分。第 1 章介绍 Java 语言的发展历程及特点、开发环境的搭建及开发工具的使用等；第 2 章介绍 Java 应用程序和 Java Applet 的编辑、编译与运行过程等；第 3 章介绍 Java 程序设计的基础知识，包括数据类型、运算符与表达式、流程控制、数组与字符串等；第 4 章介绍面向对象的概念、类的基本组成、对象的引用、方法的重载与重写、继承、抽象类、接口、包以及常用的修饰符等；第 5 章介绍 File 类、常见的输入输出流、异常概念、异常的处理机制以及自定义异常等；第 6 章介绍线程的概念与使用、线程同步与共享、多线程操作等；第 7 章介绍事件处理模型、AWT 和 Swing 中常用的组件以及布局管理等；第 8 章介绍 TCP/IP、UDP、Socket 协议，以及如何利用 Java 语言进行网络编程等；第 9 章介绍 J2EE 运行环境搭建，包括 Tomcat 服务器的安装、Mysql 数据库的安装与配置以及 MyEclipse 集成开发环境的搭建等；第 10 章介绍 JSP 常用的内置对象、标签以及如何运用 JSP 进行 Web 开发等；第 11 章介绍 Struts2 的相关概念、常用标签以及常用组件等；第 12 章介绍 Hibernate 与 Struts2 的整合；第 13 章结合实例介绍基于 Spring、Hibernate、Struts2 的项目开发过程。

本书由张剑飞担任主编，杜晓昕、金涛担任副主编，其中第 1、3、4 章由杜晓昕编写，第 5、6、7、9 章由金涛编写，第 2、8、10、11、12、13 章由张剑飞编写。本书得到齐齐哈尔大学 2010 重点教材项目资助。

由于编者水平有限，书中难免存在一些疏漏和不妥之处，敬请专家和读者批评指正。

编 者
2011 年 5 月

目 录

第 1 章 Java 语言概述	1
1.1 Java 语言简介	1
1.1.1 Java 语言的产生	1
1.1.2 Java 语言的特点	2
1.2 开发环境的搭建	3
1.2.1 下载 JDK	4
1.2.2 安装 JDK	4
1.2.3 设置 path 与 classpath	6
1.2.4 测试 Java 开发环境	7
1.3 辅助工具简介	8
1.4 Eclipse 集成开发工具简介	10
小结	13
习题	13
第 2 章 Java 应用程序与 Java Applet	14
2.1 Java 应用程序	14
2.1.1 源程序的编辑	14
2.1.2 源程序的编译	15
2.1.3 字节码文件的运行	16
2.2 Java Applet	17
2.2.1 编写 Applet 程序	17
2.2.2 Applet 程序的编译与运行	19
小结	20
习题	20
第 3 章 Java 程序设计基础	22
3.1 标识符与关键字	22
3.1.1 标识符	22
3.1.2 关键字	23
3.2 数据类型	23
3.2.1 基本数据类型	24
3.2.2 常量	24
3.2.3 变量	25
3.2.4 基本数据类型之间的转换	26
3.3 运算符与表达式	28
3.3.1 运算符	28
3.3.2 表达式	33
3.4 控制结构	34
3.4.1 顺序结构	34
3.4.2 选择结构	34
3.4.3 循环结构	41
3.4.4 跳转结构	45
3.5 数组	46
3.5.1 一维数组	47
3.5.2 多维数组	52
3.6 字符串	55
3.6.1 String 字符串创建	55
3.6.2 String 类的常用方法	56
3.6.3 字符串与数值之间的转换	59
小结	59
习题	59
第 4 章 Java 面向对象程序设计	62
4.1 面向对象的基本特征	62
4.2 类	63
4.2.1 类的定义	63
4.2.2 成员变量和局部变量	65
4.2.3 成员方法	67
4.3 对象的创建和构造方法	69
4.3.1 对象的声明	69
4.3.2 对象的创建	69
4.3.3 对象的使用	70
4.3.4 构造方法	70
4.4 方法重载	71
4.5 类的继承	72
4.5.1 继承的定义	72
4.5.2 成员变量的隐藏和 方法重写(覆盖)	74
4.5.3 this 与 super 关键字	77
4.5.4 继承中的构造方法	81

4.6 包(package).....	83	5.5.2 对象解序列化.....	118
4.6.1 包的声明.....	83	5.6 异常.....	120
4.6.2 包的导入.....	83	5.6.1 异常的概念.....	120
4.7 权限控制.....	84	5.6.2 异常处理.....	121
4.7.1 公有访问修饰符: public.....	84	5.6.3 使用 throws 声明异常.....	124
4.7.2 保护访问修饰符: protected.....	85	5.6.4 使用 throw 抛出异常.....	125
4.7.3 默认访问修饰符: default.....	86	5.6.5 异常的多态.....	126
4.7.4 私有访问修饰符: private.....	87	5.6.6 自定义异常.....	126
4.8 关键字 final 与 static.....	88	小结.....	127
4.8.1 关键字 final.....	88	习题.....	128
4.8.2 关键字 static.....	91	第 6 章 多线程程序设计.....	131
4.9 抽象类与接口.....	94	6.1 线程的概念.....	131
4.9.1 抽象类.....	94	6.2 线程的创建和启动.....	132
4.9.2 接口.....	96	6.2.1 继承 Thread 类.....	132
4.9.3 抽象类和接口的异同.....	97	6.2.2 实现 Runnable 接口.....	134
4.10 多态.....	99	6.2.3 两种线程创建方式比较.....	135
小结.....	100	6.3 线程的状态与控制.....	137
习题.....	100	6.3.1 线程的状态.....	137
第 5 章 I/O 流与异常.....	104	6.3.2 线程的控制.....	138
5.1 File 类.....	104	6.4 线程的同步.....	145
5.1.1 构造方法.....	104	6.4.1 同步方法.....	145
5.1.2 File 类的主要方法.....	105	6.4.2 同步块.....	149
5.2 流.....	106	6.4.3 多线程产生死锁.....	150
5.2.1 流的基本概念.....	106	小结.....	151
5.2.2 输入输出流.....	106	习题.....	151
5.3 字节流.....	108	第 7 章 图形用户界面.....	152
5.3.1 InputStream 和 OutputStream.....	108	7.1 图形用户界面概述.....	152
5.3.2 FileInputStream 和 FileOutputStream.....	109	7.2 事件处理.....	153
5.4 字符流.....	110	7.2.1 事件处理模型.....	153
5.4.1 Reader 和 Writer.....	110	7.2.2 事件类.....	155
5.4.2 InputStreamReader 和 OutputStreamWriter.....	111	7.2.3 事件监听器.....	155
5.4.3 FileReader 和 FileWriter.....	112	7.2.4 事件及其相应的监听器接口... ..	156
5.4.4 BufferedReader 和 BufferedWriter.....	113	7.3 Swing 组件.....	159
5.4.5 PrintStream 和 PrintWriter.....	114	7.3.1 窗体——JFrame 类.....	160
5.5 序列化.....	115	7.3.2 面板——JPanel 类.....	161
5.5.1 对象序列化.....	116	7.3.3 标签——JLabel 类.....	162
		7.3.4 按钮——JButton 类.....	164
		7.3.5 文本框——JTextField 类与 JPasswordField 类.....	165

7.3.6 文本区——JTextArea 类.....	166	10.1.1 JSP 概述.....	203
7.3.7 列表组件——JComboBox 类 和 JList 类.....	167	10.1.2 JSP 的生命周期.....	203
7.3.8 复选框和单选按钮—— JCheckBox 类和 JRadioButton 类.....	171	10.2 JSP 的基本语法.....	204
7.4 布局管理器.....	173	10.2.1 JSP 指令.....	204
小结.....	178	10.2.2 JSP 中脚本元素.....	205
习题.....	178	10.2.3 常用的动作标签.....	206
10.2.4 注释.....	209	10.3 JSP 的内建对象.....	209
第 8 章 Java 的网络程序设计	180	10.3.1 page 内建对象.....	209
8.1 基础知识.....	180	10.3.2 request 内建对象.....	210
8.1.1 TCP/IP 分层结构.....	180	10.3.3 response 内建对象.....	210
8.1.2 套接字(Socket)概述.....	183	10.3.4 session 内建对象.....	211
8.2 Java 网络包(java.net).....	184	10.3.5 application 内建对象.....	211
8.2.1 服务器端 ServerSocket.....	184	10.3.6 pageContext 内建对象.....	211
8.2.2 客户端 Socket.....	185	10.3.7 out 内建对象.....	212
8.2.3 使用 BufferedReader 从 Socket 上读取数据.....	187	10.3.8 exception 内建对象.....	212
8.2.4 使用 PrintWriter 写 数据到 Socket 上.....	187	10.4 JSP 登录验证.....	213
8.3 Socket 编程实例.....	187	小结.....	218
8.3.1 单客户端通信.....	187	习题.....	218
8.3.2 多客户端聊天程序.....	189	第 11 章 Struts2 应用	219
小结.....	193	11.1 Struts2 入门.....	219
习题.....	194	11.1.1 Struts1 与 Struts2 的关系.....	219
第 9 章 J2EE 运行环境搭建	196	11.1.2 Struts2 的工作流程.....	221
9.1 服务器搭建与安装.....	196	11.1.3 Struts2 深入剖析.....	221
9.1.1 Tomcat 的安装.....	196	11.2 Struts2 实例分析.....	222
9.1.2 Mysql 安装与配置.....	197	11.2.1 配置 Struts2 类库.....	222
9.2 在 MyEclipse 中创建 Web 工程.....	198	11.2.2 编写登录 Action.....	222
9.2.1 绑定 Tomcat 服务器.....	198	11.2.3 Struts2 配置文件.....	223
9.2.2 新建工程向导.....	200	11.2.4 配置 web.xml.....	224
9.2.3 Web 工程目录结构和在 MyEclipse 中添加类库.....	201	11.2.5 JSP 登录界面.....	224
小结.....	202	11.3 Struts2 配置文件详解.....	226
习题.....	202	11.3.1 Struts2.xml 配置文件.....	226
第 10 章 JSP 应用	203	11.3.2 Struts.properties.....	227
10.1 JSP 基础概述.....	203	11.3.3 package 配置.....	229
		11.3.4 拦截器的配置.....	229
		11.3.5 action 配置.....	230
		11.3.6 result 配置.....	231
		11.4 Action 详解.....	231
		11.4.1 Actionsupport 类的作用.....	231

11.4.2	不继承任何类的 Action.....	232	小结.....	259
11.4.3	Action 执行方法的配置.....	232	习题.....	260
11.4.4	Action 中获取 request、 response 和 session 对象.....	233		
11.5	Struts2 标签.....	234	第 13 章 结合 Spring、Hibernate、 Struts2 实例开发.....	261
11.5.1	OGNL 表达式语言.....	234	13.1 Spring 概述.....	261
11.5.2	控制标签.....	235	13.1.1 Spring 的起源和背景.....	261
11.5.3	数据标签.....	238	13.1.2 Spring 框架.....	262
11.5.4	表单标签.....	240	13.2 网站系统和数据库设计.....	263
11.6	Struts2 拦截器.....	241	13.2.1 网站需求分析.....	263
11.6.1	拦截器的定义.....	241	13.2.2 系统运行环境搭建.....	264
11.6.2	Struts2 中默认拦截器.....	242	13.2.3 网站系统设计.....	265
11.6.3	实现自定义拦截器.....	243	13.2.4 信息系统初始化模块.....	266
小结.....		246	13.2.5 系统数据库设计.....	268
习题.....		246	13.3 ssh 项目开发流程.....	271
第 12 章 整合 Hibernate 与 Struts2.....		247	13.3.1 项目搭建.....	271
12.1	Hibernate 概述.....	247	13.3.2 项目功能流程简介.....	273
12.1.1	ORM 简介及实现原理.....	247	13.3.3 修改 web.xml 配置文件.....	273
12.1.2	Hibernate 的作用.....	248	13.3.4 修改 applicationContext-*.xml 配置文件.....	275
12.1.3	Hibernate 的优点.....	248	13.3.5 修改 Struts2 配置文件.....	278
12.2	Struts2 与 Hibernate 整合.....	248	13.3.6 编写 Action 表示层代码.....	279
12.2.1	添加 Hibernate 特性.....	248	13.3.7 编写 Service 层代码.....	280
12.2.2	配置 Admin 实体类持久化.....	249	13.3.8 编写 Dao 持久层代码.....	281
12.2.3	Hibernate 的配置文件.....	251	13.4 相关技术阐述.....	281
12.2.4	Dao 层代码.....	252	13.4.1 数据库连接池技术实现.....	281
12.2.5	Action 层代码.....	254	13.4.2 Struts2 拦截器的实现.....	282
12.2.6	页面视图.....	255	13.4.3 Spring 容器对 Hibernate 事务的管理.....	283
12.2.7	运行效果.....	257	13.4.4 页面的分页查询.....	284
12.3	Hibernate 深入学习.....	258	小结.....	284
12.3.1	Hibernate 的核心接口.....	258	习题.....	284
12.3.2	数据检索.....	258	参考文献.....	285
12.3.3	hibernate.cfg.xml 的 作用及编写说明.....	259		
12.3.4	entity.hbm.xml 的 作用及编程说明.....	259		

第 1 章 Java 语言概述

学习目标

内 容	要 求
Java 语言的产生	了解
Java 语言的特点	熟悉
JDK 的下载与安装	掌握
环境变量的设置	掌握
EditPlus 的安装与使用	掌握
Eclipse 的安装与使用	掌握

Java 是由 Sun 微系统公司研发的一种新型的面向对象的编程语言。它是随着 Internet 及信息技术的飞速发展而发展起来的，是目前最常用的一个功能强大的跨平台的计算机编程语言，是主要的网络开发语言之一，也是发展迅速的嵌入式操作系统的绝佳组合。由于 Java 语言开源、并且提供功能丰富的类库，而且具有面向对象、分布式、多线程、可移植性、安全性高和稳定性强等特点，目前重量级的公司都广泛采用 Java 语言进行项目的开发。Java 语言已经在众多的高级语言中脱颖而出。

1.1 Java 语言简介

在学习 Java 语言之前，首先了解一下 Java 语言的发展历程及其特点。

1.1.1 Java 语言的产生

1991 年，美国 Sun Microsystems 公司为了能够在消费电子产品上开发应用程序，成立了“绿色项目组”(Green Project)，该小组主要由 James Gosling 负责，成员主要包括 Patrick Naughton、Chris Warth、Ed Frank 和 Mike Sheridan 等。这个小组最初的目标是能够在诸如电冰箱、电视机、PDA 等数字控制的电子消费产品上开发应用程序，然而消费电子产品种类繁多，即使是同一类消费电子产品其采用的处理芯片和操作系统也不相同，也存在着跨平台的问题。当时最流行的编程语言是 C 和 C++ 语言，该小组的研究人员就考虑是否可以采用 C++ 语言来编写消费电子产品的应用程序，但是研究表明，对于消费电子产品而言 C++ 语言过于复杂和庞大，并不适用，安全性也并不令人满意。于是该小组就以 C++ 为基石，融合 C 和 C++ 等传统语言的优点，开发了一种独立于硬件平台的、面向对象的程序设计语言，并命名为 Oak(取名自 Gosling 办公室外的一棵橡树)。当时，Oak 语言并没有引起人们的注意。

直到 1994 年，随着互联网和 WWW 的飞速发展，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，Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点非常符合互联网的需要。于是，“绿色项目组”将他们的开发转向了 Internet，用 Oak 编写了一系列网络应用程序，例如，网络浏览器 WebRunner 等。

1995年,由于商标冲突,Oak语言被改为Java语言。同年,WebRunner正式改名为HotJava。HotJava浏览器得到了Sun公司首席执行官Scott McNealy的支持,并得以研发和发展。通过这个完全用Java语言设计的浏览器不仅充分显示了Java语言环境的威力,而且为在更复杂、离散、异构的Internet网上进行分布式Java编程提供了一个理想的平台。后来,Sun公司又决定让程序开发者免费使用Java,这才真正地将Java推向了全世界。

其实Java名字的由来还流传着一个故事,一天,Java小组成员正在喝咖啡时,议论给新语言起个什么名字的问题,有人提议用Java(Java是印度尼西亚盛产咖啡的一个岛屿),这个提议得到了其他成员的赞同,于是就采用Java来命名此新语言。

Sun公司虽然推出了Java,但这仅仅是一门编程语言,如果想开发比较大的项目则必须要有一个强大的开发类库,于是Sun在1996年推出了JDK1.0。该版本包括两个方面:JRE(Java Runtime Environment)、JDK(Java Development Kit)。在JRE中包括API(核心API、用户界面API、集成API)、发布技术、JVM(Java Virtual Machine);JDK包括编译Java程序的编译器(javac命令)、解释器(java命令)等。Sun在1997年推出JDK1.1,新增了JIT——即时编译器。它与传统编译器的区别在于,传统编译器只能编写一条语句,运行完后扔掉,再编写下一条语句;而JIT则是将经常用到的指令保存在内存中,当下次调用时不需要再编译,大大提高了JDK的效率。

一直以来,Java主要应用在网页的Applet上以及一些移动设备中。但是,到了1996年底,Flash的面世击败了Java在网页Applet上的应用地位。

虽然从1995年Java诞生到1998年,Java依然是互联网上使用最广的语言,但是Java并没有找到它自己准确的位置。直到1998年底,Sun推出了JDK1.2,这是Java发展史上最重要的版本,其中加入了许多新的设计。鉴于1.2版本与之前版本的巨大差别,Sun公司将1.2版本以及以后版本命名为“Java2”。目前,最新版本为JDK1.6。

由于Java提供了强大的图形、图像、音频、视频、多线程和网络交互能力,它已经成为当今推广最快的最为流行的网络编程语言。Java的出现引起了软件开发的重大变革,成为推动IT业蓬勃发展的最新动力。它的出现对整个计算机软件业的发展产生了重大而深远的影响。

目前,Java技术通常分为3大部分:J2SE(Java 2 Standard Edition)、J2ME(Java 2 Micro Edition)和J2EE(Java 2 Enterprise Edition)。

J2SE是Java2的标准版,主要用于桌面应用程序的编程,为台式机和 workstation 提供一个开发和运行的平台。它是最基础的Java技术,定义了一般的Java语言规范,如程序界面、I/O、多线程和网络编程等。本书在学习Java的过程中,主要是采用J2SE来进行开发。

J2ME是一种高度优化的Java运行环境,主要是面向消费类电子设备(如手机、机顶盒、PDA等)提供的Java运行平台。

J2EE主要是为实现分布式企业开发提供的-一个应用服务器的运行和开发平台。

1.1.2 Java语言的特点

Java语言是一门重要的网络编程语言,它所具有的特点如下。

1. 简单性

Java语言是在C和C++语言的基础上进行简化和改进的一种新型语言,它的语法与C和C++语言的语法类似,简单且容易掌握。同时Java语言摒弃了C和C++语言的复杂、不安全

特性，例如：摒弃了 C 语言的全局变量、宏定义、全局函数，以及结构、联合和指针数据类型、指针的操作和内存的管理等。此外，Java 语言提供了种类丰富、功能强大的类库，并且通过垃圾自动回收机制简化了程序内存管理，使 Java 程序变得简单容易编写。Java 程序的简单性是其得以迅速普及的最重要原因之一。

2. 完全面向对象

在现实世界中，任何实体都可以看作是一个对象。面向对象模型是一种模拟人类社会和人解决实际问题的模型，它更符合人们的思维习惯。Java 语言是一种完全面向对象的编程语言，它将数据封装于类中，利用类的优点，实现了程序的简洁性和便于维护性。面向对象也是 Java 语言最重要的特性。

3. 平台无关性

平台无关有两种：源代码级和目标代码级。C 和 C++具有一定程度的源代码级平台无关，用 C 和 C++写的应用程序不用修改只需重新编译就可以在不同平台上运行。Java 的平台无关性是目标代码级的，是指 Java 语言编写的应用程序的目标文件直接可以在不同的软、硬件平台上运行。这也是 Java 语言具有“一次编译，到处运行”口号的原因。Java 语言的平台无关性主要是由 Java 虚拟机(JVM)实现的。

4. 安全性

现今的 Java 语言主要用于网络应用程序的开发，因此对安全性有很高的要求。Java 语言去除了 C/C++中易造成错误的指针，增加了自动内存管理等措施，同时，Java 语言提供了异常处理机制，有效地避免了因程序编写错误而导致的死机现象，保证了 Java 程序运行的安全稳定性。

5. 多线程

多线程类似于多进程机制，多线程机制使一个进程能够被划分为若干线程并并发执行。多线程能够带来更好的交互性能和实时控制性能。C 语言和 C++语言采用单线程体系结构，而 Java 语言支持多线程技术。

1.2 开发环境的搭建

要编写一个 Java 程序，必须先安装开发环境，开发环境包括开发 Java 程序必须的 JDK 工具和一个编辑软件。

JDK(Java Development Kit)是 Java 软件开发工具箱，提供了编译和运行 Java 程序的所有工具和常用的类库。

编辑软件可以使用计算机上的任何一个文本编辑器，如记事本、UltraEdit、EditPlus、TextPad 等。另外，对于大型项目开发来说，为了用户更方便地进行程序的编写及调试，可以使用功能强大的集成开发环境(Integrated Developing Environment, IDE)，如 JCreator、Eclipse、JBuilder 等，这些 IDE 都提供了拼写检查、代码自动完成、关键字特殊显示、第三方插件等功能。

本小节主要对 JDK 的下载、安装、配置和测试进行详细的讲解。

1.2.1 下载 JDK

甲骨文公司(Oracle 公司)于 2010 年 1 月完成对 Sun 公司的收购,所以, JDK 可以从 Oracle 公司的官方网站 <http://www.oracle.com> 免费下载。JDK 当前的最新版本为 jdk1.6.0_23, 读者可以根据不同的操作系统平台来下载相应的 JDK, 本书以 32 位的 Windows XP 系统为例, 介绍其下载的具体过程。

(1) 在浏览器的地址栏中输入 <http://www.oracle.com>, 打开如图 1.1 所示的页面。

(2) 在页面里单击 Downloads 按钮, 并选择 Popular Downloads 中的 Java for Developers 命令, 如图 1.2 所示。

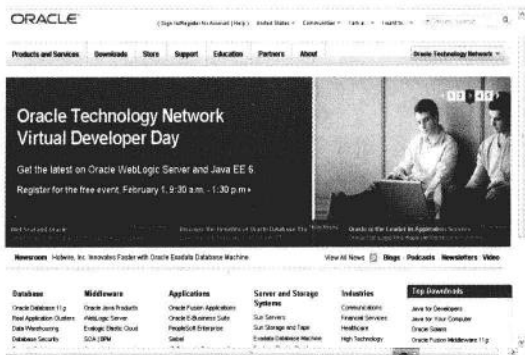


图 1.1 进入 Java 的官方网站



图 1.2 选择菜单项 Java for Developers

(3) 进入 Java SE 下载页面后, 单击 Java Download 按钮, 进行下载, 如图 1.3 所示。根据机器的操作系统, 选择相应的 JDK 版本, 这里选择 Windows, 并选中 I agree to the Java SE Development Kit 6u23 License Agreement 复选框, 单击 Continue 按钮, 进入 JDK 文件下载页面, 如图 1.4 所示。

Java SE Development Kit 6u23

Provide information, then Continue to Download

Select Platform and Language for your download:

Platform:

Language: Multi-language

I agree to the [Java SE Development Kit 6u23 License Agreement](#).

Continue »

图 1.3 设置下载条件的页面

Download Information and Files

Instructions: Click the file name to start the download.

Available Files

File Description and Name	Size
Java SE Development Kit 6u23 jdk-6u23-windows-i586.exe	76.32 MB

图 1.4 JDK 文件下载的面

(4) 单击 [jdk-6u23-windows-i586.exe](#) 链接进行下载。

1.2.2 安装 JDK

下载完成后, 即可进行安装, 下面以 Windows XP 的操作系统为例介绍 JDK 的安装步骤。双击已下载的安装程序 [jdk-6u23-windows-i586.exe](#), 运行 Java SE 的安装程序, 如图 1.5 所示。

(1) 选择需要安装的功能组件, 单击“更改”按钮, 可更改 JDK 的安装目录, 如图 1.6 所示。完成设置后单击“下一步”按钮, 继续安装。

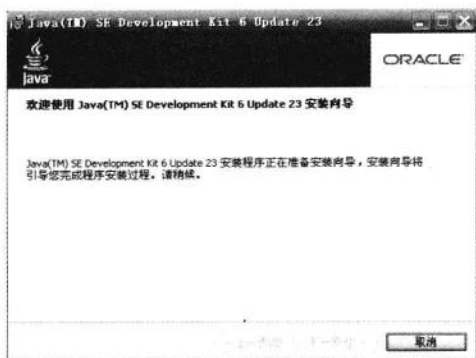


图 1.5 安装 JDK 的初始界面

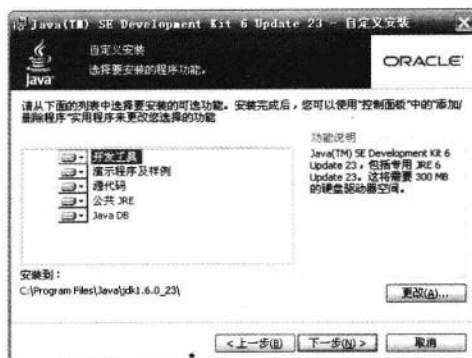


图 1.6 JDK 安装界面

(2) JDK 安装完毕后, 如果是第一次安装 JDK, 会自动跳转到 JRE 安装界面, 如图 1.7 所示, 单击“更改”按钮, 可更改安装 JRE 的目录。这里采用默认设置, 完成设置后单击“下一步”按钮, 继续安装。

(3) JRE 安装完成后, 会弹出如图 1.8 所示的界面, 单击“完成”按钮, 完成安装。



图 1.7 JRE 安装界面

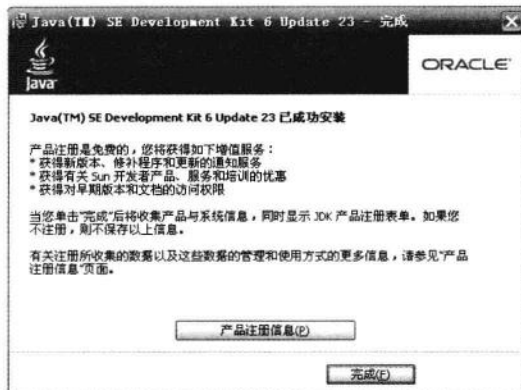


图 1.8 安装完成界面

(4) 安装完 JDK 后, 在 JDK 安装目录下(本例的 JDK 的安装目录为 C:\Program Files\Java\jdk1.6.0_23)有几个重要的子目录, 见表 1-1。

表 1-1 JDK 目录

目 录	包含的内容
bin	Java 开发工具(以可执行文件为主)
demo	演示程序
include	与 C 相同的头文件
jre	Java 运行环境文件
lib	Java 开发类库
sample	示例程序

另外, 在安装目录下有库源文件的压缩文件 src.zip, 该压缩文件中包含了 Java 库公开部分的源代码, 有兴趣的读者可以解压该文件, 去深入研究 Java 源代码。

(5) 在 bin 目录下包含的是 Java 开发工具, 下面介绍几种常用的 Java 开发工具, 见表 1-2。

表 1-2 Java 主要开发工具

工 具	名 称	作 用
javac	Java 编译器	将 Java 源代码转换成字节码
java	Java 解释器	直接从类文件执行 Java 应用程序字节码文件
appletviewer	小程序浏览器	执行 HTML 文件上的 Java Applet 的浏览器
javadoc	Java 文档生成器	从 Java 源文件生成 HTML 格式的 API 文档
jdb	Java 语言调试器	可以逐行执行程序、设置断点和检查变量
javap	Java 反编译器	将 class 反编译成 Java bytecodes

1.2.3 设置 path 与 classpath

安装完 JDK 后, 为了使系统能自动找到命令所在的目录, 需配置环境变量。配置的环境变量主要包括 path 和 classpath 这两个环境变量, 下面给出在 windows 操作系统中配置环境变量的主要步骤。

(1) 右击“我的电脑”图标, 在弹出的快捷菜单中选择“属性”命令, 弹出系统属性对话框, 如图 1.9 所示。

(2) 在“系统属性”对话框中, 选择“高级”选项卡, 如图 1.10 所示。

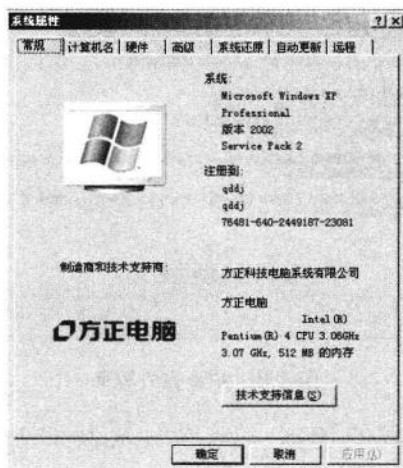


图 1.9 “系统属性”对话框

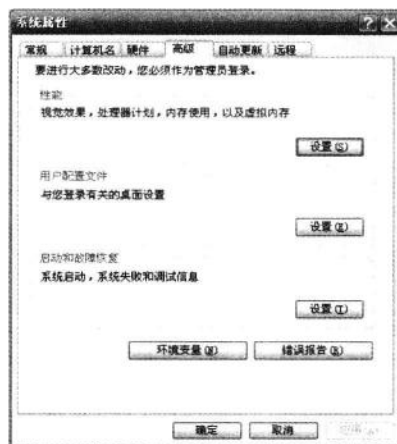


图 1.10 “高级”选项卡

(3) 单击“环境变量”按钮, 弹出“环境变量”设置对话框, 如图 1.11 所示。

(4) 在“系统变量”列表框中找到变量名“path”, 并选中, 双击进入“编辑系统变量”对话框, 如图 1.12 所示。

(5) 在“变量值”文本框的最后面输入“; C:\Program Files\Java\jdk1.6.0_23\bin”, 注意有分号, 主要是为了与前面原来的内容分隔开。完成编辑后, 单击“确定”按钮。

(6) 在“系统变量”对话框中单击“新建”按钮, 弹出“新建系统变量”对话框, 如图 1.13 所示。在“变量名”文本框中输入“classpath”, 在“变量值”文本框中输入“.; C:\Program Files\Java\jdk1.6.0_23\lib\tools.jar; C:\Program Files\Java\jdk1.6.0_23\lib\dt.jar”, 如图 1.14 所示。完成编辑后, 单击“确定”按钮, 完成环境变量的配置。



图 1.11 “环境变量”对话框

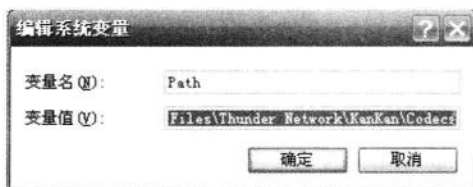


图 1.12 编辑 path 系统变量对话框

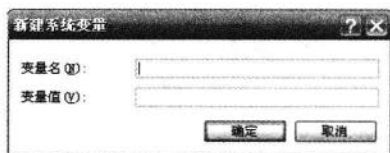


图 1.13 “新建系统变量”对话框

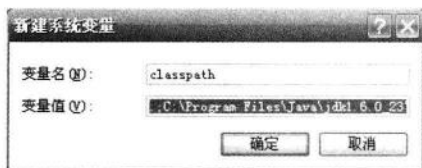


图 1.14 classpath 系统变量设置

说明：①设置 path 环境变量的目的是为了指向 JDK 的 bin 目录，在 bin 目录下放置了各种编译执行命令，通过该环境变量的设置，不管源文件在任何路径上，都可以通过该环境变量直接找到相应的命令对源文件进行编译执行，否则，必须将源程序复制到 bin 目录下，方可进行编译执行；②设置 classpath 环境变量的目的是当需要导入已经定义好的类时，可以直接从 classpath 类路径中查找，“.”代表的是当前目录。

1.2.4 测试 Java 开发环境

配置完成后，需要测试配置是否正确，其具体步骤如下。

(1) 单击“开始”按钮，在弹出的“开始”菜单中再选择“运行”命令，弹出如图 1.15 所示的对话框。在“打开”文本框中输入“cmd”，单击“确定”按钮，弹出如图 1.16 所示的界面。

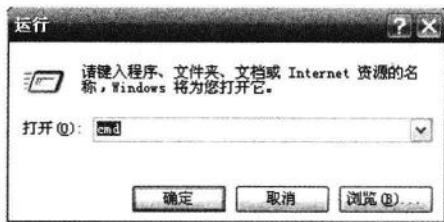


图 1.15 “运行”对话框

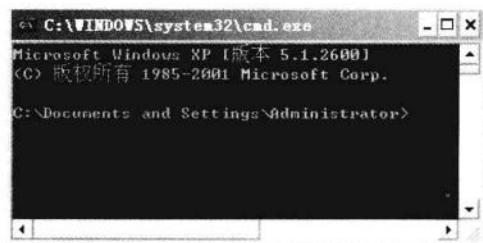


图 1.16 DOS 窗口

(2) 在 DOS 窗口中输入“java -version”命令，此命令用于显示 Java 的版本信息，若安装