

国家级精品课程配套教材

新
版

21世纪

高职高专系列教材

Java 程序设计 案例教程

◎钱银中 编著

◎眭碧霞 主审



提供源代码和电子教案



机械工业出版社
CHINA MACHINE PRESS



国家级精品课程配套教材
21 世纪高职高专系列教材

Java 程序设计案例教程

钱银中 编著

眭碧霞 主审



机械工业出版社

本书全面、系统地介绍了 Java 语言及其程序设计。全书共分 15 章,第 1 章至第 3 章介绍了 Java 语言基础,第 4 章至第 7 章介绍了面向对象程序设计,第 8 章介绍了 Java 处理异常的机制,第 9 章至第 11 章介绍了图形界面程序设计,第 12 章至第 14 章分别介绍了多线程、输入输出流和数据库程序设计,第 15 章通过案例讲述如何在实际的软件项目开发中综合应用 Java 技术。

本书在介绍基本理论的同时,注重培养学生的实践技能。书中提供了大量实例,通过对这些实例的分析和程序设计,学生可以提高分析和解决实际问题的能力。本书的附录介绍了 Java 的主流开发工具 Eclipse 的用法。

本书可作为高职高专软件技术、计算机应用技术、信息管理技术等专业的教材,也可作为 Java 语言的培训教材或 Java 语言爱好者的自学用书。

图书在版编目(CIP)数据

Java 程序设计案例教程 / 钱银中编著. —北京:机械工业出版社, 2008.2

(国家级精品课程配套教材. 21 世纪高职高专系列教材)

ISBN 978-7-111-23329-9

I. J… II. 钱… III. JAVA 语言-程序设计-高等学校:技术学校-教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 008771 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:王 颖

责任印制:李 妍

北京鑫海金澳胶印有限公司印刷

2008 年 2 月第 1 版·第 1 次印刷

184mm×260mm·19.25 印张·474 千字

0001—5000 册

标准书号: ISBN 978-7-111-23329-9

定价: 28.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话:(010) 68326294

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379753 88379739

封面无防伪标均为盗版

21 世纪高职高专计算机专业系列教材

编委会成员名单

主 任 周智文

副主任 周岳山 林 东 王协瑞 张福强 陶书中

龚小勇 王 泰 李宏达 赵佩华 陈 晴

委 员 (按姓氏笔画排序)

马 伟 马林艺 卫振林 万雅静 王兴宝

王德年 尹敬齐 卢 英 史宝会 宁 蒙

刘本军 刘新强 刘瑞新 余先锋 张洪斌

张 超 杨 莉 陈 宁 汪赵强 赵国玲

赵增敏 贾永江 陶 洪 康桂花 曹 毅

睦碧霞 鲁 辉 裴有柱

秘书长 胡毓坚

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国近 60 所高等职业院校的骨干教师对在 2001 年出版的“面向 21 世纪高职高专系列教材”进行了全面的修订和增补,并更名为“21 世纪高职高专系列教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中,本系列教材获得了较高的评价,并有多品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- (1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- (2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- (3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- (4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- (5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

Java 程序设计是高等职业技术学院软件技术、计算机应用技术、信息管理技术等专业的一门必修课。编者近几年一直承担高职院校 Java 程序设计的教学工作,通过教学深刻感到,教好这门课必须充分认识高职学生的学习特点和这门课程的特点,按照高职学生的教学要求实施教学。本教材正是按照这样的思路编写的。

首先,本教材体现了高职教学中理论够用、实践为重的教学要求。教材没有过多阐述 Java 语言涉及的计算机理论和算法理论,而是重点讲解基本知识,培养学生 Java 编程的岗位技能。

其次,本教材在内容安排上体现了前后内容的连贯性。这门课程的一个很大特点是连贯性强,前面章节学习的好坏直接影响到后续章节的学习效果。对于学生在学习过程中普遍感到困难的内容,进行了适当的分解,这样可以使学生循序渐进地学习,避免产生畏难情绪。

第三,本教材采用案例教学法和实践教学法,即采用案例教学法讲解基本知识,对每个重要的知识点采用实例进行讲解,这些实例融知识性和趣味性于一体,且全部取自学生熟悉的场景。采用实践教学法,以大量实验强化学生对基本理论和基本技能的练习,使学生可以直观地获得知识、技能和技巧。

第四,本教材强调编程语言和开发工具的统一。本书附录介绍了最流行的 Java 开发工具 Eclipse 的用法,所有实例和实验都安排在 Eclipse 上编程、调试和运行。通过这门课程的学习,学生不仅能够掌握 Java 语言程序设计,而且能够熟练掌握 Eclipse 的用法,为将来从事 Java 开发打好基础。

最后,本教材内容反映了 Java 和软件技术的最新发展。本教材使用 Java 的最新版本 Java SE 6.0,教材上的很多内容,如泛型、枚举、自动包装和自动解包等,都是 Java SE 5.0 以后才出现的。

编译和运行本书上的实例和实验程序必须使用 Java SE 5.0 以上的版本,否则很多程序将无法运行。如果在 Eclipse 上编译、运行,首次使用 Eclipse 时应该按照附录 A.6.1 中的说明设置 JDK 的编译版本。

本书实例和实验的源代码可以从机械工业出版社网站 www.cmpedu.com 上下载,源代码和电子教案可从编者主页 <http://soft.ccit.js.cn/teacher/qyz/index.htm> 上下载。本书使用的软件可以从下列站点下载:

- Java SE 6.0 从 <http://java.sun.com/javase/downloads/index.jsp> 单击 JDK6 下的 Download 按钮下载,Java SE 6.0 英语版 API 规范从同一站点单击 Java SE6 Documentation 下的 Download 按钮下载。
- Eclipse 3.3 从 <http://www.eclipse.org> 上下载。
- MySQL 5.0 从 <http://dev.mysql.com/downloads/mysql/5.0.html#win32> 上下载。

本书由钱银中编著。感谢哇碧霞教授认真细致地审阅了全稿,并提出了许多宝贵建议。由于作者水平有限,书中的不足和欠妥之处恳请读者批评指正。

编者

目 录

出版说明

前言

第 1 章 概述 1

1.1 Java 语言的发展历史 1

1.2 Java 语言的特点 2

1.3 Java 虚拟机 3

1.4 Java 的应用 3

1.5 Java 开发环境的建立 5

1.5.1 安装 Java SE 6.0 5

1.5.2 Java 的集成开发工具 7

1.6 编辑、编译和运行 Java 应用程序 7

1.6.1 使用命令行工具编译并运行 Java 应用程序 8

1.6.2 使用 Eclipse 编辑并运行 Java 应用程序 9

1.6.3 使用 Jcreator 编辑、编译并运行 Java 应用程序 9

1.7 实验 编辑并运行 Java 应用程序 10

1.8 习题 10

第 2 章 Java 语言基础 12

2.1 标识符和关键字 12

2.1.1 Java 的字符集 12

2.1.2 标识符 12

2.1.3 关键字 13

2.2 变量和常量 13

2.2.1 变量声明 13

2.2.2 变量初始化 14

2.2.3 常量 14

2.3 基本数据类型 14

2.3.1 字面值 14

2.3.2 整型数据 15

2.3.3 浮点型数据 16

2.3.4 强制类型转换 16

2.3.5 字符型数据 18

2.3.6 布尔型数据 19

2.4 运算符 19

2.4.1 算术运算符 19

2.4.2 关系运算符 20

2.4.3 逻辑运算符 20

2.4.4 位运算符 21

2.4.5 赋值运算符 22

2.4.6 其他运算符 22

2.4.7 运算符的优先级和结合性 22

2.5 表达式、语句和语句块 23

2.5.1 表达式 23

2.5.2 语句 24

2.5.3 语句块及作用域 24

2.6 字符串 24

2.6.1 字符串字面值 25

2.6.2 字符串变量 25

2.6.3 字符串+运算 25

2.6.4 String 类的常用方法 25

2.7 输入和输出数据 26

2.7.1 输出数据到控制台 26

2.7.2 从控制台读取数据 27

2.8 实验 29

2.8.1 实验 1 基本数据类型之间的强制类型转换 29

2.8.2 实验 2 变量的作用域 30

2.9 习题 30

第 3 章 控制语句和数组 33

3.1 分支语句 33

3.1.1 if 语句 33

3.1.2 多 else 分支的 if 语句 34

3.1.3 嵌套的 if 语句 35

3.1.4 switch 语句 37

3.2 循环语句 39

3.2.1 for 循环 39

3.2.2	for 循环的其他形式	40	4.10.1	设置类的访问权限	75
3.2.3	while 循环	42	4.10.2	设置类成员的访问权限	75
3.2.4	do while 循环	43	4.11	垃圾回收	78
3.2.5	循环嵌套	43	4.12	实验	78
3.3	转移语句	45	4.12.1	实验 1 Eclipse 自动生成 getters/setters 方法和构造方法 ...	78
3.3.1	break 语句	45	4.12.2	实验 2 类变量和实例变量	79
3.3.2	continue 语句	46	4.12.3	实验 3 定义和访问指定包 中的类	79
3.4	数组	47	4.12.4	实验 4 成员变量和成员方法的 访问权限修饰符	80
3.4.1	数组的声明和创建	47	4.13	习题	82
3.4.2	使用数组元素	48	第 5 章	类的继承	87
3.4.3	引用变量之间的赋值	50	5.1	超类和子类	87
3.4.4	二维数组和多维数组	50	5.2	继承规则	88
3.5	增强型 for 循环	51	5.2.1	超类和子类在同一个包中	89
3.6	实验	52	5.2.2	超类和子类不在同一个包中	90
3.6.1	实验 1 break 在 switch 语句中的 作用	52	5.3	用 super 调用超类的构造方法 ...	92
3.6.2	实验 2 通过 main() 的参变量向程序 传送数据	53	5.4	超类变量引用子类对象、 instanceof 运算	94
3.7	习题	53	5.5	覆盖	95
第 4 章	类和对象	56	5.6	Object 类的常用方法	97
4.1	面向对象程序设计方法	56	5.7	最终类和最终方法	98
4.2	定义类	57	5.7.1	最终类	98
4.3	创建对象	58	5.7.2	最终方法	98
4.4	定义方法和调用方法	60	5.8	抽象类和抽象方法	99
4.4.1	定义方法	61	5.9	实验	101
4.4.2	调用方法	62	5.9.1	实验 1 继承规则	101
4.4.3	方法中不同变量的作用域	63	5.9.2	实验 2 最终类和抽象类	103
4.4.4	引用数据类型作为方法的参数和 返回值	64	5.10	习题	104
4.5	定义构造方法	66	第 6 章	接口和类库	107
4.6	方法的重载	68	6.1	定义接口	107
4.7	this 关键字	70	6.2	实现接口	108
4.8	static 修饰符	71	6.3	接口引用变量	112
4.8.1	实例变量和类变量	71	6.4	类库及其 API 规范	113
4.8.2	实例方法和类方法	72	6.5	Math 类	114
4.9	包	73	6.6	String 和 StringTokenizer 类	115
4.9.1	定义包	73	6.7	基本数据类型包装类	116
4.9.2	导入类	74	6.7.1	包装类	116
4.10	访问权限	75			

6.7.2	自动包装和自动解包	118
6.8	处理日期和时间的类	119
6.8.1	Date 类和 SimpleDateFormat 类	119
6.8.2	GregorianCalendar 类	120
6.9	实验	121
6.9.1	实验 1 接口的实现及接口引用 变量	121
6.9.2	实验 2 学习使用类库的 API 规范 文档	122
6.10	习题	122
第 7 章	枚举和泛型	125
7.1	枚举	125
7.2	枚举的类特性	127
7.3	泛型类	128
7.4	泛型方法	133
7.5	泛型接口	134
7.6	有多个类型变量的泛型	135
7.7	约束类型变量	136
7.8	习题	137
第 8 章	异常处理	139
8.1	异常的种类	139
8.2	异常处理	140
8.3	使用多个 catch 语句	142
8.4	finally 语句	143
8.5	手动抛出异常	144
8.6	抛出异常的方法	146
8.7	常用的预定义异常类	148
8.8	自定义异常类	149
8.9	实验 try/catch 语句的执行 流程	151
8.10	习题	151
第 9 章	图形界面程序设计	154
9.1	AWT 组件和 Swing 组件	154
9.2	容器类、组件类和辅助类	155
9.3	框架 JFrame	155
9.4	在框架中添加组件	157
9.5	面板 JPanel	158
9.6	颜色 Color	159
9.7	事件响应	160

9.8	事件的种类	165
9.9	适配器	168
9.10	绘图	170
9.11	实验 JFrame 的用法	175
9.12	习题	176
第 10 章	Swing 组件编程	178
10.1	按钮、标签和文本输入类	178
10.1.1	按钮	178
10.1.2	标签	179
10.1.3	文本框	181
10.1.4	密码框	181
10.1.5	文本区	183
10.2	布局管理器	186
10.2.1	FlowLayout 布局	186
10.2.2	BorderLayout 布局	188
10.2.3	GridLayout 布局	189
10.2.4	CardLayout 布局	190
10.3	选择组件	191
10.3.1	复选框	191
10.3.2	单选按钮	193
10.3.3	组合框	196
10.3.4	滑块	198
10.4	菜单	200
10.5	工具栏	203
10.6	对话框	206
10.6.1	标准对话框	206
10.6.2	文件对话框	207
10.7	习题	210
第 11 章	applet 应用程序	213
11.1	编写和运行 applet 应用程序	213
11.2	applet 的生命周期	215
11.3	applet 图形界面程序	215
11.4	习题	218
第 12 章	多线程	219
12.1	线程基本概念	219
12.2	Runnable 接口和 Thread 类	220
12.3	继承 Thread 类创建线程	220
12.4	实现 Runnable 接口创建 线程	222

12.5	sleep()方法	223	数据库	256	
12.6	线程优先级	225	14.3.1	新建数据库	256
12.7	线程的同步	226	14.3.2	新建 ODBC 数据源	256
12.7.1	定义同步方法	227	14.3.3	Java 程序访问 SQL Server	
12.7.2	使用同步语句	228	数据库	257	
12.8	习题	229	14.4	使用 PreparedStatement 接口	259
第 13 章	输入输出流	231	14.5	实验 安装和使用 MySQL	
13.1	File 类	231	数据库	261	
13.2	字符流和字节流	233	14.6	习题	262
13.3	使用字符流读写文件	234	第 15 章	Java 技术的综合应用	264
13.3.1	FileReader 和 FileWriter 类	234	15.1	模块程序功能简介	264
13.3.2	BufferedReader 和 BufferedWriter 类	236	15.2	定义公共类	264
13.4	使用字节流读写文件	239	15.3	设计学生基本信息输入程序	268
13.4.1	FileInputStream 和 FileOutputStream 类	239	15.4	设计学生基本信息查询程序	271
13.4.2	DataInputStream 和 DataOutputStream 类	240	15.5	单元模块的集成	274
13.4.3	BufferedInputStream 和 BufferedOutputStream 类	242	15.6	习题	277
13.5	随机读写文件	242	附录 A	Eclipse 使用简介	278
13.6	习题	244	A.1	Eclipse 的下载、安装和运行	278
第 14 章	数据库程序设计	245	A.2	编写 Java 程序	279
14.1	JDBC	245	A.3	Eclipse3.3 的 Java 透视图	282
14.1.1	安装驱动器	246	A.4	使用代码助手和错误提示	284
14.1.2	建立连接	246	A.5	Eclipse3.3 开发 Java 项目	286
14.1.3	创建执行对象	247	A.5.1	创建 Java 项目	287
14.1.4	执行 SQL 语句	247	A.5.2	创建 Java 包	288
14.1.5	处理查询结果	247	A.5.3	创建 Java 类	289
14.2	JDBC 访问 MySQL 数据库	249	A.5.4	自动生成 getter/setter 方法	289
14.2.1	MySQL 数据库的安装和使用	249	A.5.5	自动生成构造方法	290
14.2.2	Java 程序访问 MySQL 数据库	251	A.5.6	自动生成 import 语句	291
14.2.3	包含 Java 变量的 SQL 语句	253	A.5.7	重命名元素	291
14.2.4	Java 访问 MySQL 产生的 乱码及其处理	254	A.5.8	移动元素	292
14.3	JDBC 访问 SQL Server		A.5.9	删除项目	292
			A.6	编译和运行	293
			A.6.1	设置编译和运行环境	293
			A.6.2	自动编译和手工编译	294
			A.6.3	运行程序及设置运行参数	294
			A.6.4	访问.jar 和.class 文件	296
			参考文献	297	

第1章 概述

1.1 Java 语言的发展历史

Java 语言是为了应对异构环境和网络环境下的软件开发而出现的。软件在异构环境中使用,要求程序必须具有良好的跨平台性,能够在不同的硬件和软件平台上运行。Internet 的兴起从根本上改变了计算机的处理方式,使原先孤立的 PC 连接在一起,这些软件运行环境的改变导致了 Java 语言的产生。

1991 年, SUN 公司成立了一个由 James Gosling 和 Patrick Naughton 领导的开发小组,开发一种嵌入式消费类电子产品的应用程序。由于不同厂商的控制器选择了不同的 CPU,程序不能依赖特定的体系结构。然而,当时大多数计算机语言只能用于一个特定环境。例如,当时开发小组使用的 C++ 语言就是这样。为了找到更好的解决方案, James Gosling 和开发人员尝试开发一种可移植的、具有跨平台性的语言,使该语言编写的程序能够在不同环境下的不同 CPU 上运行。这种语言最初被命名为 Oak (橡树)。后来发现, Oak 是另一种计算机语言的名字,于是将其改名为 Java。

正当 James Gosling 带领他的开发人员设计 Java 的时候,出现了万维网 (World Wide Web, WWW) 和 Internet。万维网的关键技术是将超文本页面转换到浏览器中显示,其主要创作语言是 HTML (Hypertext Markup Language)。HTML 能够提供文本、图片、音乐和录像等静态的信息,但是不能与用户交互。

Internet 是由许多类型的计算机、操作系统、CPU 组成的网络空间,编写 Internet 上的交互程序同样要求程序具有良好的跨平台性。Java 设计人员在开发嵌入式消费类电子产品时遇到的问题在 Internet 编程时同样存在。由于万维网具有广阔的发展前景,随后,Java 语言关注的重点从消费类电子产品转移到了 Internet 程序设计。

1995 年,Java 语言的设计者用 Java 编写了第一个支持 Java 的浏览器 HotJava,并且让 HotJava 能够执行网页中内嵌的 applet 代码,这一成果引发了人们延续至今对 Java 的热情。

1996 年初, SUN 公司发布了 Java 1.0 版,但很快发现存在明显的缺陷,不能用于真正的应用开发。后来的 Java 1.1 版改进了事件响应,并为 GUI 增加了新的事件处理模型,但仍有很大的局限性。

1998 年,Java 1.2 版发布时, SUN 公司将其改名为 Java 2 标准版软件开发工具箱 1.2 版 (Java 2 Standard Edition Software Development Kit Version 1.2, J2SDK1.2)。J2SDK1.2 用精细的图形工具箱取代了早期版本中玩具式的 GUI,并且更接近“一次编写,随处运行”的目标。Java 1.2 标准版发布的同时, SUN 推出了用于嵌入式设备的 Java 微型版 (J2ME) 以及用于服务器的 Java 企业版 (J2EE)。

J2SDK 1.3 和 J2SDK 1.4 版扩展了类库,增加了新特性,提高了系统性能。

2004 年底, J2SDK 1.5 版发布,该版本后来改名为 Java SE 5.0,它是自从 Java 发布以来

改动最大的一次。Java SE 5.0 增加的新特性包括泛型、枚举、自动包装和自动解包、for each 循环、可变元参数、元数据和静态导入。

2006 年, SUN 公司发布 Java SE 6.0。Java SE 6.0 对桌面应用程序提供了强大的支持, 拥有更多、更强的桌面 API。

表 1-1 列出了标准版不同版本的 API 类和接口数目。

表 1-1 Java 不同版本的 API 类和接口数目

版 本	类和接口数目	版 本	类和接口数目
1.0	212	1.4	3020
1.1	504	5.0	3279
1.2	1781	6.0	3777
1.3	2130		

1.2 Java 语言的特点

Java 语言的创始人在《Java 语言环境白皮书》中说明了设计 Java 语言的目的和最终取得的成就, 总结出 Java 语言具有如下 11 个特点。

1. 简单

Java 语言的创始人原先都是 C++程序员, 因此 Java 语言的语法很接近 C++语言。但是, Java 语言舍弃了 C++语言中许多很少使用、难以理解、令人混乱的特性, 使其变得更简单。另一方面, 由于 Java 最初的目标是用在消费类电子产品上, 因此它具有小型化的特点, 基本的编译器和类库只需要大约 40KB。另外, 标准类库和线程支持只需要增加 175KB。

2. 面向对象

在过去的 30 多年中, 面向对象技术已经证明了自身的价值。在日益复杂、日益网络化的环境中, 编程必须采用面向对象概念。Java 语言是完全的面向对象语言, 所有代码都在类中定义和使用。Java 提供了一个清晰和高效的面向对象开发平台。

3. 相似性

Java 语言与 C++语言非常相似, Java 语言保持了 C++语言的许多面向对象的特性。这表明, C 或 C++程序员可以很快过渡到 Java 平台。

4. 健壮性

Java 语言的设计目标之一是编写高可靠性的软件。Java 语言提供了编译时检查和运行时检查, 语言特性会引导程序员形成可靠的编程习惯。用户可以满怀信心地编写 Java 代码, 因为开发过程中系统将会发现很多错误, 不至于主要错误推迟到产品发布时才发现。

5. 安全性

Java 语言适用于分布式环境, 这意味着安全是至关重要的。由于 Java 语言本身和 Java 虚拟机都具有安全特性, 使用 Java 语言可以构建外部无法入侵的系统。网络环境下, 用 Java 语言开发的应用系统可以阻止未经授权的代码进入系统、传播病毒或入侵文件系统。

6. 不依赖体系结构

Java 语言有很好的跨平台性, 同一个经过编译的 Java 应用程序能够在不同的硬件平台

和不同操作系统上运行。

7. 可移植性

Java 的可移植性一方面体现在它不依赖体系结构, 另一方面, Java 规定了基本数据类型 的字节长度, 例如, int 类型的整数永远是 32 位。程序在任何平台上数据格式是一致的, 不存在不同硬件和操作系统上数据类型不兼容的问题。

8. 解释型

Java 编译器编译产生的不是可执行代码, 而是字节码。字节码是由 Java 虚拟机执行的高度优化的一系列指令, Java 程序编译成字节码后, 虚拟机逐行解释执行。尽管不同平台上的 Java 虚拟机不同, 但它们解释执行的 Java 字节码是一样的。因此, 解释字节码是创建具有跨平台性的可移植程序的有效方法。

9. 高性能

Java 的自动垃圾收集线程在极低的优先级下运行, 一旦需要, 就可以立即通过收集垃圾获得内存资源, 导致了更好的性能。

10. 多线程

网络应用程序通常要求同时做多件事。多个任务同时执行能提高系统资源的利用率, 加快程序运行的速度。Java 的多线程技术提供了构建含有许多并发线程的应用系统的途径和方法。

11. 动态性

Java 语言采用动态链接方式, 类只有被使用后才被链接, 也就是说, 只有需要某些代码时, 才将其添加到正在运行的程序中。

1.3 Java 虚拟机

Java 语言最令人瞩目的特性是它具有跨平台性, 这一特性的基础是 Java 虚拟机 (Java Virtual Machine, JVM), 也叫运行时系统。Java 代码编译后生成的.class 文件不是可执行代码, 而是字节码。字节码是经过高度优化的一系列指令序列。多数程序设计语言出于性能考虑, 使用编译方式运行程序, 即一次性编译生成可执行文件。而 Java 编译后生成的是字节码, 最终由 JVM 解释并执行。Java 程序运行时, 虚拟机逐一读取并翻译执行这些字节指令。程序解释执行要比编译执行慢, 但是运行性能上的这点损失我们很难感觉到。

在任何平台上, 只要运行了 Java 虚拟机, 就可以运行 Java 程序。尽管不同平台上的虚拟机不一样, 但是这些虚拟机解释执行 Java 字节码的方式是一样的, 解释执行的结果也是一样的。假如 Java 程序编译成可执行代码, 每个 Java 程序要为运行该程序的每种 CPU 准备一种版本, 这样就无法实现跨平台性, 可见, 解释执行字节码是创建可移植程序的最简单方法。

Java 解释执行也确保 Java 更加安全, 因为每个 Java 程序的执行都处于 JVM 的控制下, JVM 可以防止程序中的恶意代码侵入系统。

1.4 Java 的应用

经过 10 多年的发展, Java 已经渗透到全球每个角落。2005 年的一次市场调查显示, 全

全球有 25 亿部电子设备采用了 Java 技术，Java 智能卡发行量已经达到 10 亿张，采用 Java 技术的手机达 7 亿多部，7 亿台 PC 中应用了 Java 技术，140 家电信运营商支持 Java 技术，全球有超过 450 万 Java 开发人员。

目前，Java 的主要应用领域是 Web 开发，Java Web 应用占 Java 开发领域的一半以上。Java Web 使用的是 Java 技术和在 Java 基础上发展起来的 J2EE 技术。SUN 公司的 Java 企业版（JavaEE）就是一个免费的 J2EE 服务器。由于 J2EE 技术在企业中的普及应用，出现了众多支持 J2EE 技术的服务器，如 Bea 公司的 Weblogic，IBM 公司的 WebSphere，SUN 公司的 SUNONE，以及自由软件 Tomcat、JBoss 等。运行在这些服务器上的企业应用软件广泛使用在金融、保险、证券、学校、制造企业、政府机关等部门。图 1-1 是使用 Java Web 技术开发的常州市地方税务申报管理系统。

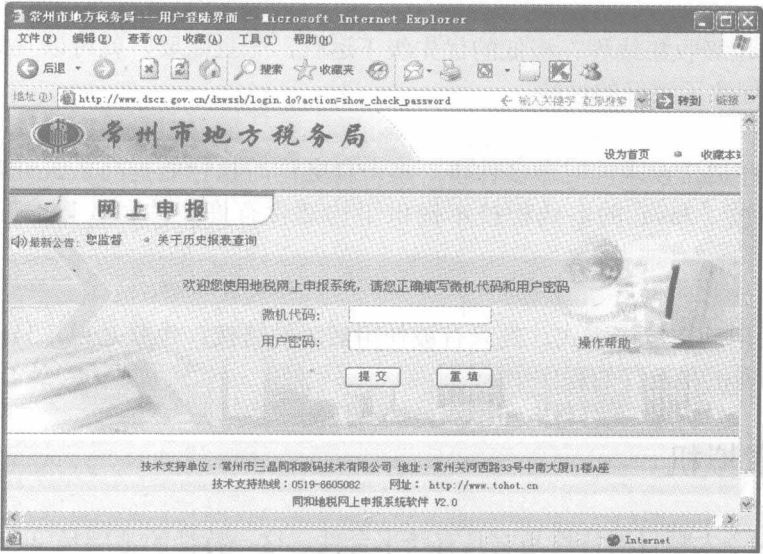


图 1-1 Java Web 技术的应用

使用 Java 标准版（JavaSE）开发图形界面接口（Graphical User Interface，GUI）程序尽管有不尽如人意之处，例如它没有 Delphi、C#等开发工具提供的图形拖放功能，但 Java 仍然被很多开发者证明是一种很适合用于开发运行在多种操作系统平台上的桌面应用软件。创建图形界面的 AWT 和 Swing 组件是 Java 基础类库的重要组成部分，Java 为图形界面程序提供了丰富的图形功能和交互性能。图 1-2 是使用 Java 的 Swing 组件编写的用户账号管理程序，左图用个性化的小图标显示所有用户，右图用表格显示了所有用户。

Java 最初是为嵌入式消费类电子产品的应用程序设计的。凭借 Java 微型版（JavaME）和 Java 卡技术版（Java Card Technology），Java 回到了嵌入式系统领域。

Java 最主要的特点是跨平台。这个特点对消费类电子产品市场是十分重要的。JavaME 就是在此基础上为可编程、资源有限的消费类产品定义的架构，希望通过该版本把 Java 技术带给手机、寻呼机、机顶盒、汽车仪表、数字电视及其他设备。在设计上，为了满足客户和嵌入式市场对设备灵活性和按需定制的要求，JavaME 采用了模块化和可扩展的结构。

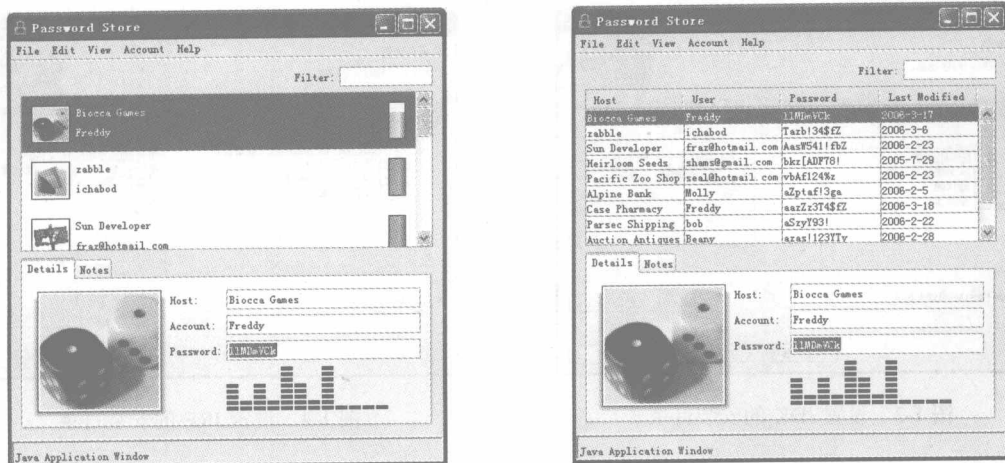


图 1-2 使用 Java 的 Swing 组件编写的 GUI 桌面应用程序

Java 卡技术版是为 Java 智能卡设计的。Java 智能卡具有应用与操作系统无关、支持一卡多应用、应用程序可在卡片发行后动态并安全地下载或更新应用程序等特点。Java 智能卡技术已经成为第三代移动通信（3G）用户身份识别卡(USIM)的事实标准，在金融或银行信用卡领域，Java 智能卡也得到了广泛应用。

1.5 Java 开发环境的建立

当前 Java 标准版的最高版本是 Java Platform, Standard Edition 6，简称 Java SE6，本书以这个版本讲述 Java 程序设计。不同版本的 Java 产品可以从 SUN 公司的网站上免费下载。

Java SE6 提供了两个软件产品，Java 运行时环境（Java SE Runtime Environment, JRE）和 Java 开发工具箱（Java SE Development Kit, JDK）。JRE 提供类库、Java 虚拟机以及运行 Java 应用程序和小应用程序所需的其他组件。JDK 包括命令开发工具，如 javac、java、appletviewer 等，以及编译器和调试器。Java SE6 安装完毕后，将在安装目录下看到这两个产品的根目录 jre1.6.0 和 jdk1.6.0。

1.5.1 安装 Java SE 6.0

下载 Java SE 6.0 后，安装过程可以参阅 SUN 公司提供的安装指南。读者可以到 <http://java.sun.com/javase/6/webnotes/install/index.html> 查找不同操作系统下的安装软件和安装指南。对于 Windows XP，下载后双击安装软件就可以按照提示安装完毕。安装过程中，单击图 1-3 或图 1-4 中的更改按钮可以更改 JDK 和 JRE 的安装目录。默认条件下，全部安装到 C:\Program Files\Java（根目录与操作系统所在的目录一致）。

如果在 DOS 命令窗口编译并运行 Java 程序，安装结束后，还要设置环境变量 Path 和 classpath 的值。环境变量 Path 是设置 JDK 命令文件所在路径，设置环境变量 Path 后，可以在任何路径下执行 JDK 命令。环境变量 classpath 是设置类库所在的路径，设置后 Java 程序就可以访问类库中的类了。

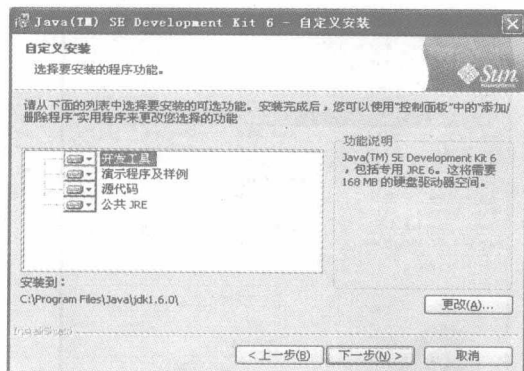


图 1-3 设置 JDK 的安装目录

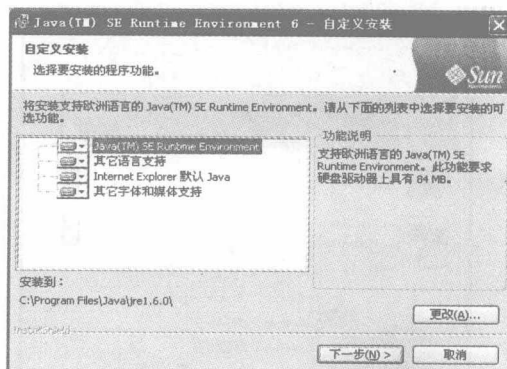


图 1-4 设置 JRE 的安装目录

设置 path 的方法如下：

- 1) 用鼠标右键单击桌面上的“我的电脑”，选择“属性”，打开“系统属性”界面。
- 2) 单击标签“高级”，并单击其中的“环境变量”按钮，打开“环境变量”对话框，见图 1-5。
- 3) 如果“系统变量”选择框中无“Path”，单击下面的“新建”按钮，打开“新建系统变量”对话框，并在变量名中输入 Path。如果选择框中有“Path”，选中它并单击下面的“编辑”按钮，打开“编辑系统变量”对话框，见图 1-6。

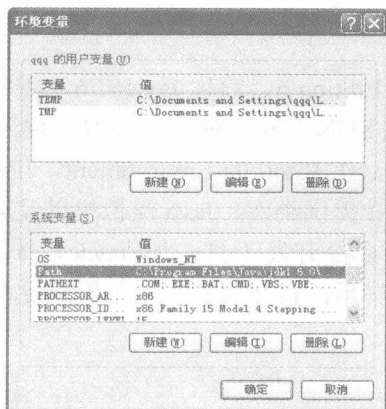


图 1-5 “环境变量”对话框

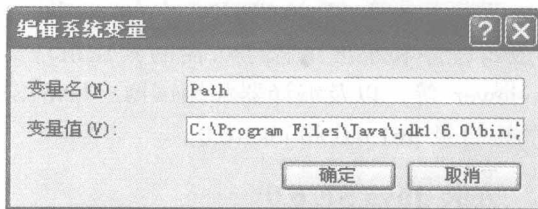


图 1-6 设置环境变量 Path

- 4) “编辑系统变量”对话框中，在“变量值”文本框的最前方增加“C:\Program Files\Java\jdk1.6.0\bin;”，如果安装时更改了 jdk 的安装目录，应该使用实际的安装目录值。

- 5) 设置完毕后单击“确定”按钮，Path 设置完毕。

设置 classpath 方法如下：

- 1) 在图 1-5 中，如果“系统变量”选择框中无“classpath”，单击下面的“新建”按钮，打开“新建系统变量”对话框，并在变量名中输入 classpath。如果选择框中有“classpath”，选中它并单击下面的“编辑”按钮，打开“编辑系统变量”对话框。
- 2) 在“新建/编辑系统变量”对话框中，变量值文本框的最前方增加“.;C:\Program

Files\Java\jre1.6.0\lib\rt.jar”。如果安装目录不是默认的，应该用实际的安装目录值。注意，分号前的 . 不能少，它表示当前目录，分号后面的路径是类库的路径。

3) 设置完毕后单击“确定”按钮，classpath 设置完毕。然后单击“确定”按钮关闭其他对话框。

1.5.2 Java 的集成开发工具

开发 Java 项目通常要使用集成的开发工具，Eclipse 和 JCreator 是深受程序员欢迎的两种工具。

本书所有的实例和实验都在 Eclipse 上完成，书后的附录 A 详细介绍了 Eclipse 的用法。

运行 Eclipse 前必须安装 JDK。Eclipse 不需要安装过程，下载并解压缩后就可以直接使用。

Jcreator 有两种版本：Pro 版和 LE 版。使用 Pro 版需要向 XINOX 公司购买许可证，不过购买前有一个月的试用期。使用 LE 版不需要购买许可证，但是功能受到限制。这两种版本都可以从 <http://www.jcreator.com/download.htm> 下载。同样，运行 Jcreator 前也必须安装 JDK。Jcreator 安装过程很简单，可以按照安装过程中的提示安装完毕。

1.6 编辑、编译和运行 Java 应用程序

Java 程序分为 Java 应用程序 (Java Application) 和小应用程序 (Java Applet)。Java 应用程序可以独立运行，小应用程序嵌入在 Web 页面，在浏览器中运行。本章介绍 Java 应用程序的编辑、编译和运行方法，小应用程序在第 11 章介绍。

【例 1-1】基本的 Java 应用程序。

问题：编写一个 Java 应用程序，在控制台上输出“Hello World!”。

分析：Java 应用程序必须有 main() 方法，运行 Java 应用程序就是运行 main() 方法中的代码，本程序就是在 main() 中调用 System.out.println("Hello World!") 方法输出字符串“Hello World!”。Java 是完全面向对象的语言，所有方法都在类中，main() 方法不能独立存在，必须定义一个类包含它。

程序源代码：

```
/**
 * Hello.java
 * version 1.0 2007-03-20
 * author CCIT
 */
public class Hello
{
    public static void main(String[] args)
    {
        // 在显示器上输出"Hello World!"
        System.out.println("Hello World!");
    }
}
```