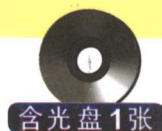


程序设计视频教程丛书



Java 程序设计 视频教程

尹立民 王兴东 等编著

视频教程，让程序设计不再难学

- ◆ 详尽的编程方法讲解
- ◆ 丰富的经验、技巧与提示
- ◆ 完整的实例源代码
- ◆ 典型实例的全程视频演示



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

程序设计视频教程丛书

Java 程序设计视频教程

尹立民 王兴东 等编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书详细介绍了安装 Java 运行环境、语法规则、Applet 程序、网络编程等知识，并通过大量实例程序进行了说明，每章均附有习题和参考答案。作者根据多年的教学和写作的丰富经验，把 Java 写得深入浅出，易于掌握。全书概念清晰，通过正反对比帮助读者理解编程思想，例题丰富，逻辑性强，语言通俗流畅。凡是对计算机操作有一些认识的读者都能通过学习本书掌握 Java 编程的关键。

本书光盘收录了书中典型实例的视频演示全过程，读者跟随光盘的演示可以轻松掌握 Java 的编程方法。光盘中还提供了全书所有实例的源代码，方便读者学习使用。

本书适合大专院校相关专业教学使用，也特别适合希望学习程序设计的爱好者自学。

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

Java 程序设计视频教程 / 尹立民等编著. —北京：电子工业出版社，2005.6
(程序设计视频教程丛书)
ISBN 7-121-01224-3

I J... II.尹... III.JAVA 语言—程序设计—教材 IV.TP312

中国版本图书馆 CIP 数据核字 (2005) 第 048077 号

责任编辑：张瑞喜

印 刷：河北省邮电印刷厂

出版发行：电子工业出版社出版

北京市海淀区万寿路 173 信箱 邮编：100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：16.25 字数：343 千字

印 次：2005 年 6 月第 1 次印刷

印 数：5000 册 定价：29.00 元 (含光盘 1 张)

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。
联系电话：(010)68279077。质量投诉请发邮件至 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

丛 书 前 言

近年来,软件业的发展非常迅速,软件范围之广,版本更新之快,令人目不暇接。对于一个初学程序设计的读者来说,首先面临的问题是:选择什么样的开发工具?选择经典版本还是最新版本?有没有超越具体编程语言和开发工具的一般编程要素?

●► 选择开发工具

一般来说,从事应用软件的开发,可以选择的开发工具主要有Visual Basic、Visual C++、Delphi、C++ Builder、Java等,这些开发工具经过了市场的重重考验,有鲜明的特点和优点,应用较广。现在的大型软件项目开发都是以项目组方式运行,也就是需要项目组中的人分工合作完成,为了最后系统集成的顺利,一般都尽量使用统一的开发工具。所以,从使用这些著名的软件开发工具开始学习程序设计,可以少走弯路。

●► 选择版本

软件版本的选择,是令很多初学者苦恼的一件事:“听说想学的软件版本很快就要升级了,我是不是等到新版本出来了再学?要不然学了很快就过时了,没用了”。其实这种担心完全没有必要,软件的新版本不是横空出世,而是在旧版本的基础上进行改进而产生的,有了以前的使用经验,你只需要了解一下新特性,花很短的时间将自己的知识升级一下,就可以适应新版本了。

还有一个关于版本的问题也常常令许多人苦恼:“我是不是总要用软件开发工具的最新版本去工作?”其实,只要能够满足需要,实现所需功能就可以。“无论黑猫白猫,捉住耗子就是好猫”,如果这个版本完全能够满足需要,使用起来也得心应手,完全不必赶时髦非用最新版本不可。一般来说,当程序员使用自己所熟悉的开发工具时,其工作效率要比使用陌生的开发工具高得多。

当然,对于最新版本的了解是必要的,也许以前要调用多个函数、使用若干技巧才能实现的功能,在新版本中只需通过一个属性设置或者一行代码就可以实现。总之,应该选择能够实现所需功能且实现方法快捷的版本。

●► 编程要素

Visual Basic、Visual C++、Delphi、Java等都提供了面向对象的集开发、调试、编译于一体的集成开发环境,尤其令用户感到方便的是它们都提供了方便、好用的控件,这样甚至不用编写代码就可以生成一个可运行的、界面漂亮的程序,所以很多读者都沉湎于学习各种各样的控件使用技巧,而忽视了编程的基本要素。编程最基本的要素是什么呢?翻看各种软件开发工具的帮助就可以发现,它们是常量、变量、数组、数据类型、条件分支结构、循环控制结构等。虽然在不同的开发工具中,它们的语法形式略有不同,但是本质是相同的,能够正确地使用它们是实现高效率程序设计的关键。

本丛书在编写过程中,选择经典开发工具,以目前对初学者普遍适用、易学的版本为主,通过大量的实例,对核心编程要素进行了重点讲解,帮助读者快速掌握各种程序设计的方法。

本丛书的编写独具特色，主要体现在以下几个方面。

视频教程为学生自学和教师教学提供方便

本书每本书均配有一张多媒体视频教程光盘，光盘中收录了该书中典型实例的视频演示全过程，读者跟随光盘的演示可以轻松掌握该软件的安装方法和典型应用。通过观看视频演示过程，可以使初学者发现在自学过程中容易“卡壳”的问题所在，使读者能够一看就会，无师自通。对于有条件进行多媒体教学的学校，课堂演示也会给教学带来更好的效果。另外，光盘中还提供了书中实例的源代码，方便读者学习使用。

通过正反对比帮助读者理解

在讲解过程中，对初学者常犯的错误、不恰当的编程方法进行列举和分析，通过正反对照，使读者体会正确的编程思路，并获得更多的程序设计经验和技巧。

注重编程思维培养

在编写过程中融合了编者多年的编程思想以及相应的软件工程知识，从而与实例相互配合达到内外兼修的效果。“说明”、“提示”、“技巧”等经验之谈，不仅使读者认识到如何使用语句，而且还了解如何正确、高效地使用，避免混乱、低效的使用方式，帮助读者养成良好的编程思维，为以后的发展打下坚实的基础。

本书适合大专院校相关专业教学使用和程序设计爱好者自学使用，同时也适合于那些理论基础较好但几乎不懂得什么编程诀窍的学生们阅读。新程序员们的经验主要是靠自己——吃一堑，长一智——获得的，这往往是一个艰苦而缓慢的过程。通过本丛书的学习，可以使读者直接获得大量的经验和技巧，帮助程序员尽快成长。

本书中的《程序设计基础视频教程》一书，以通俗、简明的语言，由浅入深地讲解了进行应用程序设计开发的基础知识。包括编码系统、软件工程、数据结构，流程图绘制、常用算法、数据库和网络编程基础知识等内容。建议没有编程基础知识的读者先学习此书，打好基础，然后根据自身的需要，选择本书中的其他相关书籍进行学习。

编 者

2005年5月

前 言

Java 最大的特点是跨平台，Java 应用程序可以在任何计算机系统上运行，只要计算机配备了 Java 解释器和运行环境即可。

如果你稍稍留心，就可以发现，很多软件有类似如下说明：本软件要求在 Windows XX 操作系统下运行。如果操作系统由 Windows 95 改为 Windows 2000，软件很可能无法运行。

Java 突破了这个限制，这也是 Java 如此风行的重要原因。

一般入门书籍的大部分篇幅，在介绍开发环境提供的各种功能的使用，而本书投入了比较多的精力介绍 Java 的常量、变量、数组、数据类型、条件分支结构、循环控制结构等，这些都是最基础的并且是最重要的，也是很多学习编程的人的薄弱环节。

本书各章的主要内容如下。

●第 1 章：本章是读者开始学习 Java 的前奏曲，主要介绍 Java 的发展过程以及常用的开发工具，并且教给读者如何在自己的计算机上安装并配置 Java 的开发环境。本章最后还运行了一个简单的 Java 程序，以使读者对 Java 有一个直观的认识。

●第 2 章：自然界中的信息是多种多样的，应用程序能够帮助人处理这些信息，减轻人的工作量。相应的，就需要将这些信息抽象成各种类型的数据，并根据它们的特点制订运算规则。本章介绍了 Java 的数据表示法及运算。

●第 3 章：结构化程序设计有三种基本结构——顺序结构、选择结构和循环结构。顺序结构就是按照语句的位置，从前向后执行；选择结构是根据条件执行代码中的某一部分；循环结构是反复执行某些语句。本章介绍在 Java 中实现选择结构和循环结构的语句，实现控制程序流程。

●第 4 章：Java 语言是一种纯的面向对象的语言，本章介绍了如何利用 Java 进行面向对象编程。本章的内容既多又繁杂，希望读者在语法上有了清晰概念的同时，又能准确地把握概念的内涵，并且能够在编程中熟练地使用。

●第 5 章：Java Applet 是用 Java 语言编写的一些小应用程序，这些程序是直接嵌入到页面中的，由支持 Java 的浏览器(IE 或 Netscape)解释执行能够产生特殊效果的程序。本章介绍了如何进行 Applet 程序设计，并且在网页中正确使用 Applet。

●第 6 章：本章介绍了程序设计中常用的文件的操作方法，其中包含文本文件、二进制文件和配置文件的操作。

●第 7 章：本章介绍用 AWT 来设计图形用户界面的方法，尤其是组件、容器、布局管理器等概念。学习 AWT 事件处理模型，掌握事件源、事件、事件处理者等概念，让程序能够响应用户的操作。

●第 8 章：Java 本身就是一种网络编程语言，关于网络开发的内容很多，本章旨在起一个抛砖引玉的作用，使读者初步了解 Java 网络程序设计，能够编写基于 TCP 协议的

Client/Server 程序。

参加本书编写的主要人员有：尹立民、王兴东、高发、冯欢、杜清杰、曹明玉、赵永峰、刘仙、张洁、李娜、娄竞秋、吴敬波、赵洪举、刘安华、高卫平等。由于作者水平有限，书中难免存在一些不足之处，恳请读者批评指正。

作者的电子邮箱：yinlimin@sina.com

作 者

2005 年 4 月

目 录

第 1 章 Java 概述	1
1.1 Java 的诞生与发展	2
1.2 Java 的特点	3
1.3 Java 开发工具	5
1.3.1 JDK (Java Development Kit)	5
1.3.2 Java Workshop	6
1.3.3 Borland 的 JBuilder	6
1.3.4 IBM 的 Visual Age for Java	7
1.3.5 Microsoft Visual J++	7
1.4 Java 的几个重要概念	8
1.5 JDK 安装与配置	12
1.5.1 下载 JDK1.4.2	12
1.5.2 安装 JDK1.4.2	12
1.5.3 JDK 配置	15
1.6 用 Java 编写程序	18
1.6.1 Java 程序的编译和运行	18
1.6.2 Java 的第一个实例	18
实例 1: HelloWorld 示例	18
实例 2: 编写一个简单的窗口程序	20
1.6.3 HelloWorld 程序代码分析	22
1.6.4 常见的 Hello World 程序错误 及解决方法	22
1.7 习题	24
1.7.1 选择题	24
1.7.2 简答题	24
1.7.3 编程题	24
第 2 章 Java 编程基础	25
2.1 数据类型	26
2.2 常量与变量	26
2.2.1 命名常量	27
实例 3: 计算圆面积	27
2.2.2 命名规则	29

光盘目录

第 1 章

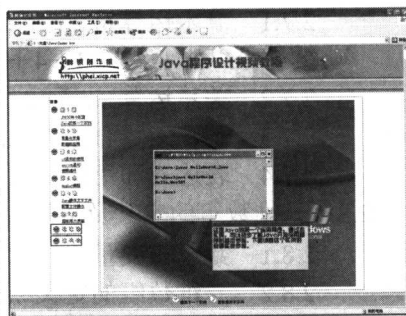


视频演示

JDK 安装与配置



Java 的第一个实例



实例代码

实例 1: HelloWorld 示例

实例 2: 编写一个简单的窗口程序

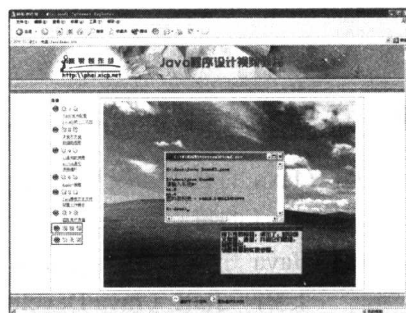
2.2.3	使用命名常量的好处	29
2.2.4	为何使用变量	29
2.2.5	定义变量	30
2.2.6	给变量命名及注意事项	31
2.2.7	初始化变量	31
	实例 4: 未初始化的变量	33
2.2.8	变量的取值范围	34
	实例 5: 测试整数的溢出	34
2.3	整型数据	35
2.3.1	整型常量	35
2.3.2	整型变量	36
2.4	实型数据	36
2.4.1	实型常量	36
2.4.2	实型变量	37
2.4.3	赋值的错误操作	37
2.4.4	为什么不用实型代替整型	37
2.5	字符型数据	37
2.5.1	字符常量	37
	实例 6: 使用转义字符	38
2.5.2	字符串常量	38
2.5.3	字符变量	39
2.5.4	字符变量的特殊性	39
	实例 7: 字符与数值	39
2.6	布尔型数据	40
2.7	运算符、表达式、语句	41
2.7.1	算术运算符	41
	实例 8: 求 $1+2+3+4+5$	42
2.7.2	关系运算	46
2.7.3	逻辑运算	47
2.7.4	表达式和语句	47
2.8	一维数组的定义及应用	48
2.8.1	数组与变量	48
2.8.2	一维数组的定义	49
2.8.3	如何给数组中的元素赋值	50
2.8.4	一维数组的应用—— 求数组中的最大值	52
	实例 9: 不使用数组, 实现让用户输入 5 个数, 然后计算最大值	53
	实例 10: 使用数组求最大值	55
2.8.5	一维数组的应用——冒泡排序	56
	实例 11: 冒泡排序实例	58

第 2 章



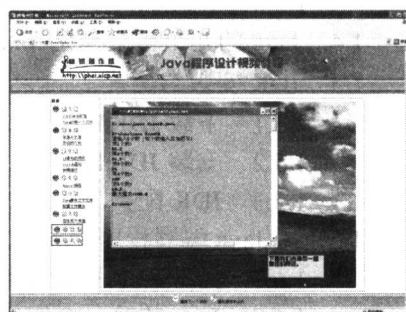
视频演示

常量与变量



视频演示

数组的应用



实例代码

实例 3: 计算圆面积

实例 4: 未初始化的变量

实例 5: 测试整数的溢出

实例 6: 使用转义字符

实例 7: 字符与数值

实例 8: 求 $1+2+3+4+5$

实例 9: 不使用数组, 实现让用户 输入 5 个数, 然后计算 最大值

实例 10: 使用数组求最大值

实例 11: 冒泡排序实例

实例 12: 一维数组的应用——选 择排序

实例 12: 一维数组的应用——选择排序 ..59	
2.8.6 一维数组的应用——找数组元素 ...61	
实例 13: 查找数组中的元素	62
2.9 多维数组	63
2.9.1 多维数组的定义	63
2.9.2 二维数组的初始化	64
2.9.3 二维数组的应用	64
实例 14: 将一个 3×4 的矩阵 a 转置后输出	64
2.10 习题	66
2.10.1 选择题	66
2.10.2 简答题	66

第 3 章 Java 中的程序控制结构 67

3.1 if 语句	69
3.1.1 编写 if 语句	70
实例 15: 用 if 语句判断大小写字母	71
实例 16: 演示 if 语句的两种格式	72
3.1.2 if 语句的正确流程	75
3.1.3 if 语句的错误用法	76
3.2 switch 语句	79
3.2.1 case 子句的有效组织方法	80
3.2.2 使用 switch 语句要注意的问题	80
实例 17: 用 switch 语句判断成绩等级	81
3.3 循环语句	82
3.3.1 while 循环	83
实例 18: 用 while 循环语句求 1000 之内 的所有水仙花数	83
3.3.2 避免无限循环	84
3.3.3 do 循环	85
3.3.4 for 循环	86
3.3.5 提前退出循环	87
3.3.6 使用 for 循环易出现的错误	87
3.3.7 正确使用 for 循环	88
实例 19: 用 for 循环求 1000 之内的所有 完全数	88
实例 20: 百钱买百鸡问题	89
3.3.8 嵌套循环	90
实例 21: 演示循环语句的综合应用	91
3.3.9 循环的类型	92
3.3.10 正确编写循环	93

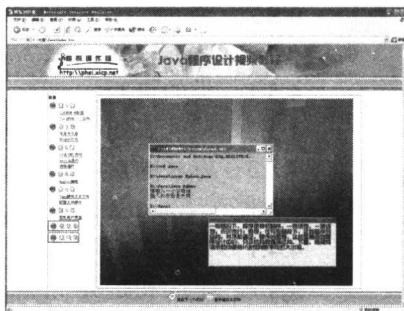
实例 13: 查找数组中的元素
实例 14: 将一个 3×4 的矩阵 a
转置后输出

第 3 章



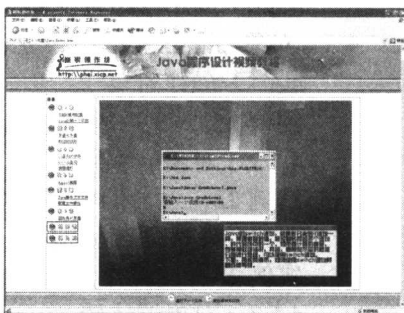
视频演示

用 if 语句判断大小写字母



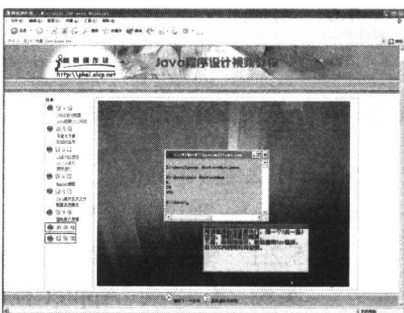
视频演示

用 switch 语句判断成绩等级



视频演示

用 for 循环求 1000 之内的所有完全数



3.4 习题.....	94
3.4.1 选择题.....	94
3.4.2 编程题.....	95
第4章 Java 面向对象编程	97
4.1 面向对象的概念	98
4.2 类与对象	98
4.2.1 对象	98
4.2.2 类	99
4.3 创建类	101
4.3.1 类的声明	101
4.3.2 域的声明——实现属性	103
4.3.3 方法的声明——实现行为	105
4.4 继承	107
4.4.1 为什么要用继承	107
4.4.2 子类和超类	107
4.4.3 继承的层次结构	110
4.4.4 继承的实现	112
4.4.5 单继承和多继承	113
4.4.6 一个继承的实例分析	114
4.5 对象	115
4.5.1 创建对象	116
实例 22: 类实例创建表达式的使用	117
4.5.2 对象的引用	120
实例 23: 对象的引用与对象之间的	
关系和区别	120
实例 24: 变量作为参数传递	122
4.5.3 对象的比较	124
实例 25: 对象的比较	124
实例 26: 字符串的比较	125
4.5.4 确定对象是类的实例	126
实例 27: 确定对象是否是类的实例	126
4.6 习题	128
4.6.1 选择题	128
第5章 Applet 编程基础	131
5.1 什么是 Applet	132
5.2 Applet 程序开发步骤	132
5.2.1 编写一个简单的 Applet 程序	133
实例 28: 演示 Applet 程序的开发过程 ...	133
5.2.2 对例子的详细解释	135



实例代码

实例 15: 用 if 语句判断大小写字母

实例 16: 演示 if 语句的两种格式

实例 17: 用 switch 语句判断成绩等级

实例 18: 用 while 循环语句求 1000 之内的所有水仙花数

实例 19: 用 for 循环求 1000 之内的所有完全数

实例 20: 百钱买百鸡问题

实例 21: 演示循环语句的综合应用

第4章



实例代码

实例 22: 类实例创建表达式的使用

实例 23: 对象的引用与对象之间的关系和区别

实例 24: 变量作为参数传递

实例 25: 对象的比较

实例 26: 字符串的比较

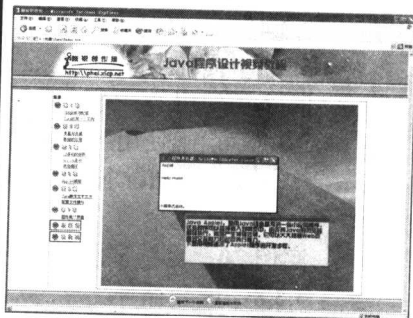
实例 27: 确定对象是否是类的实例

第5章



视频演示

演示 Applet 程序的开发过程



5.3	Applet 的生命周期.....	136
5.3.1	Applet 生命周期的划分	136
5.3.2	实例演示 Applet 的生命周期	137
	实例 29: 演示 Applet 的生命周期	137
5.4	使用 Font 类改变 Applet 的字体	139
5.4.1	Font 类中常用的方法	139
	实例 30: 显示当前系统使用的默认 字符的属性	140
5.4.2	创建自己的 Font	142
	实例 31: 显示不同大小的 Font	142
5.5	在 HTML 中使用 Applet	145
5.5.1	APPLET 标记	145
	实例 32: 使用 APPLET 标记	147
5.5.2	在 HTML 中传递 Applet 使用 的参数	148
	实例 33: 在 HTML 中传递 Applet 使用 的参数	148
5.6	实现文字显示	151
	实例 34: 实现文字显示	151
5.7	Java Applet 编程显示图像	152
5.7.1	装载一幅图像	153
5.7.2	显示图像文件	153
	实例 35: 显示图像	153
5.8	利用 Java Applet 播放声音文件	154
5.8.1	Java 支持的声音文件格式	155
5.8.2	播放声音文件的方法	155
	实例 36: 播放 music.wav 声音文件	156
5.9	Java Applet 编程响应鼠标键盘	156
5.9.1	鼠标事件	156
	实例 37: 鼠标时间跟踪	158
	实例 38: 鼠标单击处理	160
5.9.2	键盘事件	162
	实例 39: 键盘响应	162
5.10	习题	164
5.10.1	简答题	164
5.10.2	编程题	164

第 6 章 文件处理技术 165

6.1	面向存储的文件技术	166
6.1.1	文件概述	166
6.1.2	文件类型	166



实例代码

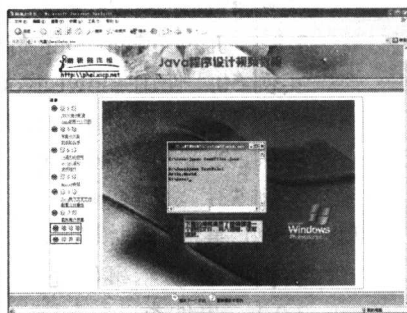
- 实例 28: 演示 Applet 程序的开发过程
- 实例 29: 演示 Applet 的生命周期
- 实例 30: 显示当前系统使用的默认字符的属性
- 实例 31: 显示不同大小的 Font
- 实例 32: 使用 APPLET 标记
- 实例 33: 在 HTML 中传递 Applet 使用的参数
- 实例 34: 实现文字显示
- 实例 35: 显示图像
- 实例 36: 播放 music.wav 声音文件
- 实例 37: 鼠标时间跟踪
- 实例 38: 鼠标单击处理
- 实例 39: 键盘响应

第 6 章



视频演示

Java 操作文本文件



6.2	File 类简介	166
	实例 40: 创建目录与文件	169
6.3	二进制文件的处理	170
	实例 41: 二进制文件的读写	171
6.4	使用 Java 操作文本文件	172
	实例 42: 使用 FileReader 读取文件	173
	实例 43: 使用 InputStreamReader 读取文件	174
	实例 44: 使用 BufferedReader 读取文件	175
	实例 45: 使用 FileWriter 写文件	176
6.5	Java 的基本数据类型与流	177
	实例 46: 序列化的应用	179
6.6	配置文件	181
	实例 47: 配置文件的应用	181
6.7	习题	182
6.7.1	编程题	182

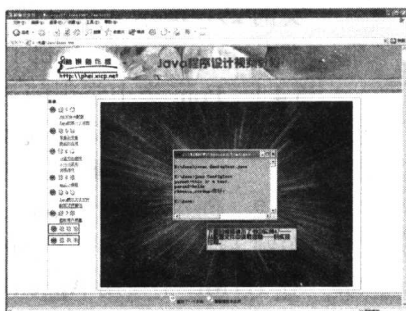
第 7 章 AWT 图形用户界面183

7.1	用 AWT 生成图形化用户界面	184
7.1.1	java.awt 包	184
7.1.2	组件和容器	184
7.1.3	常用容器	185
	实例 48: 使用 Frame 容器	186
	实例 49: 使用 Panel 容器	187
7.1.4	LayoutManager 布局管理器	189
	实例 50: 使用 FlowLayout 布局	189
	实例 51: 使用 BorderLayout 布局	191
	实例 52: 使用 GridLayout 布局	193
	实例 53: 容器嵌套	194
7.2	AWT 事件处理模型	196
	实例 54: 演示事件处理	196
7.2.1	事件类	198
7.2.2	事件监听器	199
7.2.3	AWT 事件及其相应的 监听器接口	200
	实例 55: 事件处理模型的应用	200
7.2.4	事件适配器	204
7.3	AWT 组件库	205
7.3.1	按钮 (Button)	205
	实例 56: 按钮的使用	205



视频演示

配置文件的应用



实例代码

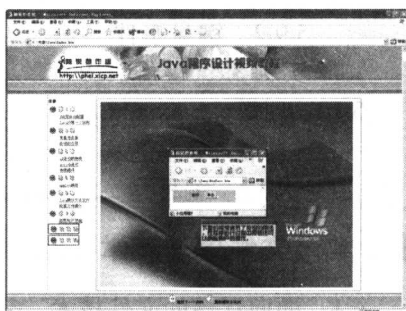
- 实例 40: 创建目录与文件
- 实例 41: 二进制文件的读写
- 实例 42: 使用 FileReader 读取文件
- 实例 43: 使用 InputStreamReader
 读取文件
- 实例 44: 使用 BufferedReader 读
 取文件
- 实例 45: 使用 FileWriter 写文件
- 实例 46: 序列化的应用
- 实例 47: 配置文件的应用

第 7 章



视频演示

图形用户界面



7.3.2	复选框 (Checkbox)	207
	实例 57: 复选框的使用	207
7.3.3	单行文本框 (TextField)	209
	实例 58: 单行文本框的使用	209
7.3.4	文本区 (TextArea)	211
	实例 59: 文本区的使用	211
7.3.5	单选按钮 (CheckboxGroup)	213
	实例 60: 单选按钮的使用	213
7.3.6	组件与监听器的对应关系	215
7.4	习题	216
7.4.1	选择题	216
7.4.2	多项选择题	216
7.4.3	填空题	216
7.4.4	程序填空题	216

第 8 章 Java 网络编程 219

8.1	网络编程的基本概念	220
8.1.1	网络的基础知识	220
8.1.2	网络的基本概念	220
8.1.3	两类传输协议——TCP、UDP	221
8.2	基于 URL 的高层次 Java 网络编程	221
8.2.1	统一资源定位器 URL	221
8.2.2	URL 的组成	222
8.2.3	创建一个 URL	222
8.2.4	解析一个 URL	223
	实例 61: 获取 URL 对象的属性	223
8.2.5	从 URL 读取 WWW 网络资源	224
	实例 62: 从 URL 读取 WWW 网络资源	224
8.3	基于 Socket (套接字) 的低层次	
	Java 网络编程	225
8.3.1	Socket 通信	225
8.3.2	Socket 通信的一般过程	226
8.3.3	创建 Socket	226
8.3.4	客户端的 Socket	227
8.3.5	服务器端的 ServerSocket	227
8.3.6	打开输入/输出流	228
8.3.7	关闭 Socket	228
8.3.8	简单的 Client/Server 程序设计	229
	实例 63: 简单的 Client/Server 程序	229
8.3.9	支持多客户的 Client/Server 程序设计	233



实例代码

- 实例 48: 使用 Frame 容器
- 实例 49: 使用 Panel 容器
- 实例 50: 使用 FlowLayout 布局
- 实例 51: 使用 BorderLayout 布局
- 实例 52: 使用 GridLayout 布局
- 实例 53: 容器嵌套
- 实例 54: 演示事件处理
- 实例 55: 事件处理模型的应用
- 实例 56: 按钮的使用
- 实例 57: 复选框的使用
- 实例 58: 单行文本框的使用
- 实例 59: 文本区的使用
- 实例 60: 单选按钮的使用

第 8 章



实例代码

- 实例 61: 获取 URL 对象的属性
- 实例 62: 从 URL 读取 WWW 网络资源
- 实例 63: 简单的 Client/Server 程序
- 实例 64: 支持多客户的 Client/Server 程序

实例 64: 支持多客户的 Client/Server	
程序	234
8.3.10 数据报 (Datagram) 通信	240
8.4 习题	241
8.4.1 判断题	241
8.4.2 填空题	242
附录 A 习题答案	243

CHAPTER



第 1 章

Java 概述



知识准备

- > 具备计算机操作的基本技能
- > 了解常见的计算机术语



本章要点

- > Java 的起源
- > Java 开发环境的安装与配置
- > 各种 Java 开发环境的比较
- > 开发一个简单的 Java 程序

本章是读者开始学习 Java 的前奏曲，主要介绍 Java 的发展过程以及常用的开发工具，并且教读者学会在自己的计算机上安装并配置 Java 的开发环境。本章最后还通过运行一个简单的 Java 程序，以使读者对 Java 有一个直观的认识。

1.1 Java 的诞生与发展

Java 来自于 Sun 公司的一个叫 Green 的项目，其最初的目的是为家用消费电子产品开发一个分布式代码系统，这样我们可以把 E-mail 发给电冰箱、电视机等家用电器，和它们进行信息交流从而对它们进行控制。

Java 的前身

在研究开发过程中，开发小组的领导人，也是 Sun 公司的一位杰出的工程师，James Gosling 深刻体会到消费类电子产品和 workstation 产品的开发理念之间的差异。

- 消费类电子产品要求高可靠、低费用、标准化、使用简单。
- workstation 用户要求强大的计算能力，而不在乎价格以及操作的复杂性。
- 消费类电子产品用户并不关心 CPU 的型号，也不欣赏专用昂贵的处理器，他们需要建立在一个标准基础之上，一系列可选的方案。

为了使整个系统与平台无关，Gosling 首先从改写 C++ 编译器着手。但是 Gosling 在改写过程中感到仅仅 C++ 是无法满足需要的，于是在 1991 年 6 月开始准备开发一个新的语言，那么给它起一个什么名字呢？

Gosling 向窗外望去，看见一棵老橡树，于是建立了一个目录取名为 Oak，这就是 Java 语言的前身（后来发现 Oak 已是 Sun 公司的另一个语言的注册商标，才改名为 Java，即爪哇——太平洋上一个盛产咖啡的岛屿名字）。

Oak 最初是一种用来给蜂窝电话及遥控器类的设备编程的语言。Oak 程序可以在需要时实时下载，这样，就可以取代以往的设备出厂前的预先编程工作，而且当设备新增功能时，客户可以及时更新，而不用再将设备送回工厂。1993 年，Sun 采用此技术建立了遥控器样机，但是没有得到销售商的广泛支持。这样，这个在技术上很成功的产品，在市场上却遇到了挫折。

说明

Java 的取名还有一个趣闻。一天，几位 Java 成员组的会员讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java(爪哇)咖啡，有一个人灵机一动说：“就叫 Java 怎样？”得到了其他人的赞赏，于是，Java 这个名字就这样传开了。

Java 进入 WWW

到了 1994 年，WWW 已如火如荼地发展起来。Gosling 意识到 WWW 需要一个中性的