

主 编：龚赤兵 王宗亮
副主编：李建俊 黄锐军

Java

面向对象程序设计 实训教程



四川出版集团·四川科学技术出版社

Java 面向对象程序设计实训教程

主 编 龚赤兵 王宗亮

副主编 李建俊 黄锐军

四川出版集团·四川科学技术出版社

图书在版编目(CIP)数据

Java 面向对象程序设计实训教程/龚赤兵,王宗亮
主编. - 成都:四川科学技术出版社,2005. 12
ISBN 7-5364-5846-0

I. J... II. ①龚... ②王... 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 117968 号

Java 面向对象程序设计实训教程

主 编 龚赤兵 王宗亮
副 主 编 李健俊 黄锐军
责任编辑 田 霞
封面设计 李 庆
版面设计 康永光
责任出版 周红君
出版发行 四川出版集团·四川科学技术出版社
成都盐道街 3 号 邮政编码 610012
成品尺寸 260mm × 185mm
印张 19.5 字数 470 千
印 刷 郫县犀浦印刷厂
版 次 2005 年 12 月成都第一版
印 次 2005 年 12 月成都第一次印刷
定 价 35.00 元

ISBN 7-5364-5846-0/TP·98

■ 版权所有·翻印必究 ■

■ 本书如有缺页、破损、装订错误,请寄回印刷厂调换。

■ 如需购本书,请与本社邮购组联系。

地址/成都盐道街 3 号 电话/(028)86671039 86672823
邮政编码/610012

前 言

Java 是目前最流行的面向对象程序设计的开发语言之一。Java 从诞生到今天, 不过十年的历史, 但已经形成了以桌面开发为主的 J2SE 平台、以移动等手持设备开发为主的 J2ME 平台以及大量使用的以企业级开发为主的 J2EE 平台。在这三个平台中, J2SE 是最基本的, 其他两个平台是在它的基础上的特殊化、复杂化和扩展化。因此, 要学习 Java, 首先需要学习 J2SE 平台。本书主要学习 Java 语言的基础, 学习平台是 J2SE。

本书具有以下特点:

(1) 以案例为驱动的教程——所有的知识点均通过精心构造的案例来讲解。每个章节配有一个案例, 集中讲解一个或几个主要概念, 理论精练够用, 主要通过案例的运行、分析和实现, 让读者学会如何运用这些知识, 在实际运用中去解决实际问题。因此本书强调的是实际技能的提高。每个章节的开头均有学习目的, 结尾都有实训内容和实训总结。这也是本书称之为实训教程的原因。

(2) 附有实训代码——在讲解案例的实现原理、关键代码之后, 不像其他书籍只附有片断代码, 我们在书中附上了案例的全部源代码, 主要是方便读者通过本书来阅读、理解这些源代码。在相关的网站上附有这些案例源代码编译后的相关文件, 读者可以轻松地运行这些案例, 但没有案例的全部源代码, 网站上只有案例不完整的代码, 这些不完整的代码, 我们称之为实训代码。读者通过学习、分析教程中的完整源代码后, 修改、增添并完善这些实训代码, 使得实训代码能够编译并顺利运行, 从而真正掌握教程中的内容。本书的相关代码保存在网站: <http://www.btpub.net/java2book/CD.rar> 中, 请下载后解压到相应的目录。在教程中提到的光盘指的就是该目录中的相应内容。

(3) 面向对象的设计代码——所有源代码按照面向对象的思想设计。为了引导读者能够阅读、理解结构合理、质量高的代码, 本书首次在高职高专教材中对设计模式进行了探讨, 讲解了 MVC 设计模式与工厂方法模式在案例中的实际运用。

(4) 采用 UML 语言——在案例的分析、描述和实现过程中, 我们采用了 UML 这种可视化建模语言, 使得类的描述图形化, 类的结构可视化, 更易于我们对案例的掌握。

本书由龚赤兵和王宗亮主编, 李建俊和黄锐军为副主编。其中龚赤兵编写第 7、8、9、10 章, 王宗亮编写第 1、2、3、11 章, 李建俊编写第 4、12、13 章, 黄锐军编写第 5、6、14 章。全书由龚赤兵统稿、修改、定稿完成。

Java 的应用是无穷无尽的, 而作者的水平是有限的, 因此, 我们欢迎读者发送邮件与我们交流、切磋。邮件地址: spencergong@yahoo.com。

编著者
2005 年 8 月

目 录

第一篇 Java 基础

第 1 章 幻灯片播放程序——布局	1
1.1 应用程序概述	1
1.2 Application 应用程序和 Applet 小应用程序	3
1.3 开发环境的建立	4
1.4 图形用户界面 GUI	7
1.5 容器和组件	7
1.6 布局	8
1.7 图像显示	9
1.8 幻灯片播放程序设计与实现	10
1.9 实训内容	16
1.10 实训总结	18
第 2 章 计算器程序——数据类型、运算符和表达式	19
2.1 应用程序概述	19
2.2 数据类型	22
2.3 运算符和表达式	25
2.4 计算器程序的设计与实现	29
2.5 实训内容	47
2.6 实训总结	47
第 3 章 商场促销计价程序——分支语句	48
3.1 应用程序概述	48
3.2 Java 语言的语句	50
3.3 Java 语言的注释	55
3.4 程序设计与实现	55
3.5 实训内容	62
3.6 实训总结	62

第 4 章 考试成绩排名程序——数组和循环语句	63
4.1 应用程序概述	63
4.2 Java 语言的数组类型	66
4.3 Java 语言的循环语句	70
4.4 程序设计与实现	71
4.5 实训内容	81
4.6 实训总结	81
第 5 章 单词拆分与统计程序——字符串	82
5.1 应用程序概述	82
5.2 字符串类 String	82
5.3 程序代码	84
5.4 字符串类 StringBuffer	87
5.5 单词拆分与统计	89
5.6 字符串常用方法	94
5.7 实训内容	98
5.8 实训总结	100
第 6 章 面积及周长计算程序——类、抽象和接口	101
6.1 应用程序概述	101
6.2 类及抽象	101
6.3 抽象类	104
6.4 基类与派生类的关系	107
6.5 应用程序设计	109
6.6 接口	114
6.7 应用接口	117
6.8 实训内容	118
6.9 实训总结	120

第二篇 Java 面向对象程序设计

第 7 章 温度转换程序——封装	121
7.1 应用程序概述	121
7.2 程序代码	122
7.3 封装的实现	130
7.4 封装的好处	130

7.5 实训内容	130
7.6 实训总结	131
第 8 章 自定义按钮程序——继承及 JButton 类	132
8.1 应用程序概述	132
8.2 继承实现代码重用	133
8.3 JButton 类	134
8.4 程序代码	135
8.5 实训内容	139
8.6 实训总结	140
第 9 章 画图程序——多态及 Graphics 类	141
9.1 应用程序概述	141
9.2 多态的实现	143
9.3 Graphics 类中的画图方法	146
9.4 实现画图程序	147
9.5 多态实现程序的可扩展性	160
9.6 实训内容	161
9.7 实训总结	162
第 10 章 温度转换程序扩展——MVC 设计模式	163
10.1 应用程序概述	163
10.2 MVC 模式	164
10.3 基本 MVC 模式实现	165
10.4 一个模型、两个视图和一个控制器	172
10.5 一个模型、两个视图和两个控制器	177
10.6 实训内容	183
10.7 实训总结	184

第三篇 Java 高级应用

第 11 章 智能交通控制程序——多线程	185
11.1 应用程序概述	185
11.2 多线程	189
11.3 程序设计与实现	194
11.4 实训内容	206
11.5 实训总结	206

第 12 章 数据查询器——JTable 组件、JDBC 和工厂方法模式	207
12.1 应用程序概述	207
12.2 JTable 组件	211
12.3 使用 JDBC 连接数据库	215
12.4 工厂方法模式实现数据库的可扩展性连接	218
12.5 实训内容	225
12.6 实训总结	226
第 13 章 网络蜘蛛程序——事件处理、网络编程、JTree 组件、html 文档分析	227
13.1 应用程序概述	227
13.2 程序设计	230
13.3 事件处理	231
13.4 网络编程	237
13.5 JTree 组件	238
13.6 程序算法	244
13.7 html 文档分析	245
13.8 程序代码	256
13.9 实训内容	270
13.10 实训总结	271
第 14 章 记事本程序——菜单、文件操作、对话框	272
14.1 应用程序概述	272
14.2 记事本菜单的设计	273
14.3 文件操作	277
14.4 对话框的设计	284
14.5 记事本 Notepad 类	291
14.6 实训内容	301
14.7 实训总结	303

第一篇 Java 基础

第 1 章 幻灯片播放程序——布局

本章目的:

- 学习布局
- 使用布局来管理图形用户界面
- 设置卡片布局, 实现幻灯片播放程序

布局 (Layout), 是用来编排、布置对象在容器界面中放置位置的管理方式, 如教室中按行排列座位, 学生上课时按顺序入座, 坐满第一行坐第二行, 坐满第二行再坐第三行等, 这是一种顺序布局。又如考试时要前后分隔一个座位坐, 则是另一种菱形梅花状的布局。

为了使图形用户界面 (GUI) 具有良好的平台无关性, Java 语言引入了“布局”来管理容器中组件对象的位置摆放。

Java 语言的布局有: 从头到尾顺序排放组件的顺序布局 (FlowLayout); 按东南西北中共 5 个区域放置组件的边界布局 (BorderLayout); 组件像名片一样重叠在一起, 而每次只能显示其中一张的卡片布局 (CardLayout); 如计算器面板上整齐划一的按钮般的网格布局 (GridLayout) 等等。此外, Java 语言也允许像 VB 语言那样, 由编程者随心所欲地安排组件的位置, 这称为空布局 (null), 也叫手工布局。

Java 应用程序的图形用户界面中, 每个容器都对应有 一个布局。对于不同的容器, 可通过设置不同的布局来管理其组件的放置情况。

使用布局的优点是: 当容器界面缩小或放大时, 容器上面的组件能自动调整位置和大小, 以适应界面的变化。而手工布局 (空布局) 则无法适应界面的变化。

在容器上设置卡片布局, 可存放多张卡片, 再在每张卡片上放置内容不同的文字, 通过上下翻页等操作来观看各卡片内容, 便可实现所谓幻灯片 (即卡片) 的播放, 这就是本章的实训程序内容。

1.1 应用程序概述

在上课、作报告、演讲等场合下, 我们常常使用 PowerPoint 来制作图文并茂的幻灯片。这里, 我们用 Java 语言编写了类似的幻灯片播放程序。运行该程序, 就可把有关的文字、图片等内容一页页地展示出来。

1. 运行前的准备

在运行本程序之前, 必须在您的计算机上安装 Java 的开发运行环境 J2SE。见 1.3 节开发

环境的建立。

2. 运行程序

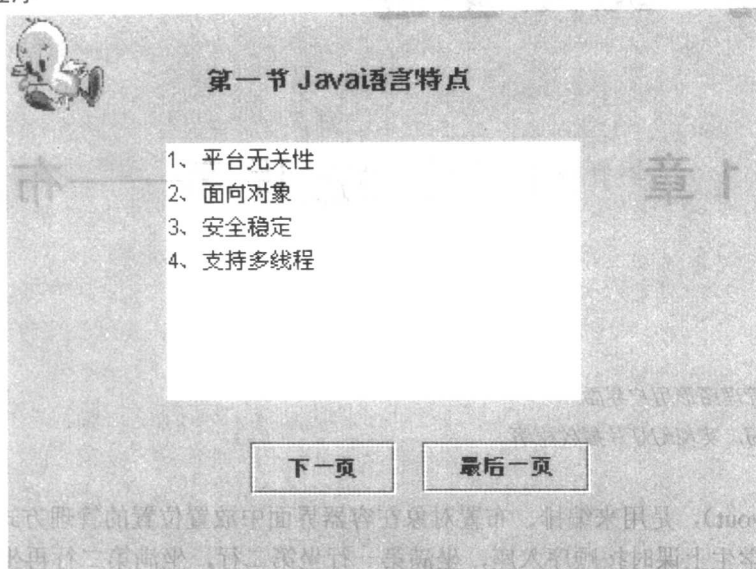


图 1-1 幻灯片播放界面 1

在 Windows 操作系统的图形界面下，通过桌面图标“我的电脑”、“我的文档”或“资源管理器”等找到幻灯片播放程序的路径（光盘上“Examples\chapter1\code”），然后直接用鼠标双击其中的超文本标记语言文件 `slidePlay.html`，即可运行。这时，在打开的 IE 浏览器上将出现图 1-1 所示的幻灯片播放界面 1。

Java 支持.gif 图形交互格式的文件，在运行界面的左上角可看见.gif 图像中卡通人物的小手在不断挥动。

3. 进入下一画面

用鼠标单击图 1-1 所示的播放界面 1 的“下一页”按钮，便进入如图 1-2 所示的幻灯片

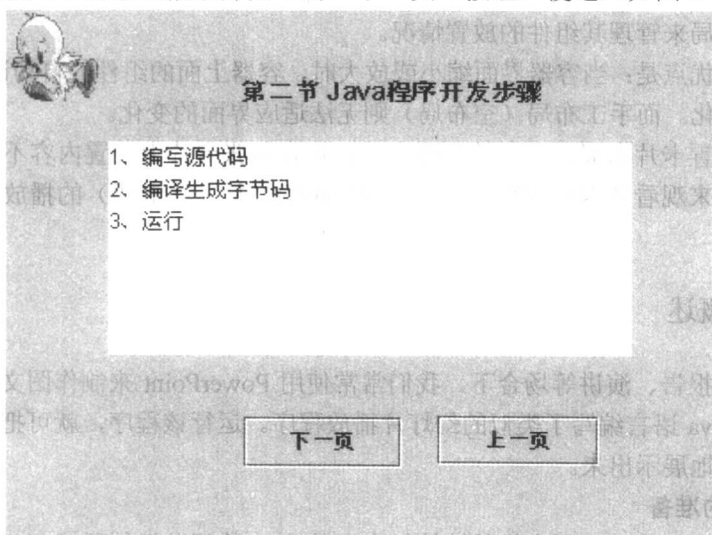


图 1-2 幻灯片播放界面 2

播放界面 2，再单击该界面的“下一页”按钮，便进入如图 1-3 所示的幻灯片播放界面 3。

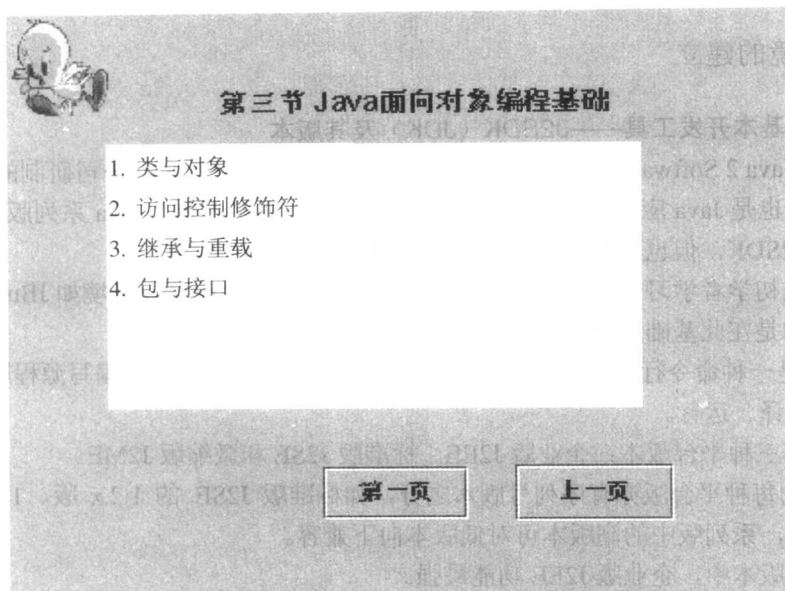


图 1-3 幻灯片播放界面 3

为简单起见，本程序只给出三张幻灯片，这些界面的其余按钮“最后一页”、“上一页”、“第一页”等具有相应的翻页功能。

4. 退出程序

用鼠标单击 IE 浏览器窗口右上角的关闭小按钮，即可退出幻灯片播放程序。

1.2 Application 应用程序和 Applet 小应用程序

Java 的应用程序具有平台无关性，即不管哪个操作系统，只要安装了 Java 语言创建者 Sun 公司的 Java 应用开发与运行环境——Java 虚拟机 JVM，如 Java2 标准版 J2SE 中的 Java 2 软件开发工具 J2SDK，均能运行。即所谓“一次编写，到处运行”。

Java 客户端的应用程序分为两类：Application 和 Applet。

其中，Java Application 应用程序是由 Java 虚拟机 JVM 独立解释执行的。在您的计算机中，只要安装了 JVM，均能顺利运行。一般地，Java Application 应用程序是由若干个类定义组成的能独立运行的 Java 应用程序，其中必有一个包含 main() 方法的主类。Application 源程序必须由 JVM 中的编译器 javac.exe 编译，生成二进制字节码的后缀为.class 的文件，才可由解释程序 java.exe 解释执行。

而 Java Applet 小应用程序是不能独立运行的 Java 应用程序，它必须嵌入到 HTML 文件中、或者在浏览器中，或者通过 appletviewer.exe 程序才能运行，故称“小”应用程序。小应用程序的编辑、编译过程与 Application 应用程序相同。一般地，小应用程序是由若干个类定义组成的程序，其中必有一个类继承 Applet 类或 JApplet 类；其编译生成的.class 字节码文件要嵌入到 HTML 文件才能运行。

Java Application 应用程序和 Applet 小应用程序运行时均可产生图形用户界面。此外，

Application 应用程序也可只在字符界面（如 DOS 命令行窗口）上运行。

1.3 开发环境的建立

1. Java 基本开发工具——J2SDK (JDK) 及其版本

J2SDK (Java 2 Softwart Development Kit) 是 Java 语言创建者 Sun 公司研制的 Java 应用软件开发工具，也是 Java 应用程序的运行环境。J2SDK 前身是 JDK，Java 系列版本 1.2 版发行后才改名为 J2SDK，但也习惯简称 J2SDK 为 JDK，或简称为 SDK。

J2SDK 是初学者学习 Java 语言的最基本的开发环境，其余的开发环境如 JBuilder、Eclipse、NetBeans 等均是在此基础上建立起来的。

J2SDK 是一种命令行开发工具，一般在记事本等文本编辑软件中编写源程序，然后在命令行窗口中编译、运行。

J2SDK 有三种平台版本：企业版 J2EE、标准版 J2SE 和微缩版 J2ME。

J2SDK 的每种平台版亦有序列号版本之分，如标准版 J2SE 的 1.2.x 版、1.3.x 版、1.4.x 版等。一般地，系列版中的高版本可对低版本向下兼容。

三种平台版本中，企业版 J2EE 功能最强。

标准版 J2SE 则是 Sun 公司免费发行的 Java 开发工具，可从 Sun 公司的站点上自由下载，还可从国内其他站点中获取，借用 google 等搜索引擎可很容易搜索到有关 J2SE 的下载点。

例如，在 Sun 公司网站下载 J2SE 1.4.2 版本的 Java 开发和运行环境，其网址为：

<http://java.sun.com/j2se/1.4.2/download.html>

打开该网页，其界面如图 1-4 所示。

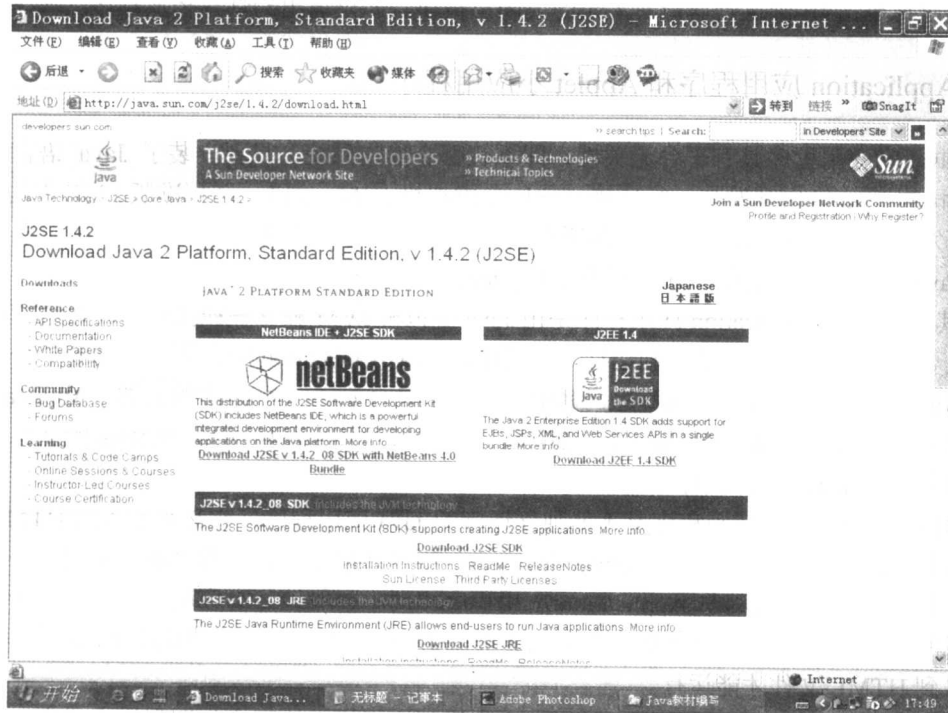


图 1-4 Sun 公司网站下载 J2SE 1.4.2 的网页

在该网页中能见到可供下载的有关版本信息：

J2SE 1.4.2

Download Java 2 Platform, Standard Edition, v 1.4.2 (J2SE)

在“J2SE v 1.4.2_08 SDK includes the JVM technology”栏目下面，可下载 J2SE 的软件开发工具“[Download J2SE SDK](#)”。

而在“J2SE v 1.4.2_08 JRE includes the JVM technology”栏目下面，则可下载 J2SE 的 Java 运行环境“[Download J2SE JRE](#)”。

注意：本书的程序代码均是在标准版 J2SE 的 1.4 版上编译运行的，故建议大家下载并安装 J2SE 1.4（或 1.4.x）版的软件开发工具 SDK。

三种平台版本中的微缩版 J2ME 是为小型消费类电子产品如手机、掌上型电脑、机顶盒等使用的。

2. 标准版 J2SE 的使用

J2SE 1.4 安装过程比较简单，一般情况下，按默认设置便可正常安装。这里不再赘述。

正常情况下，还需手工设置环境变量 PATH 和 CLASSPATH，可在控制面板——系统——系统特性对话框的“高级”选项卡——环境变量中设置，也可在“命令提示符”黑色窗口中设置，还可在 C 盘根目录下自动执行的批处理文件 AUTOEXEC.BAT 内设置。

设 J2SE 1.4 安装在 C:\jdk1.4 目录下，则在“命令提示符”黑色窗口中的设置方法如下：

```
set PATH = %PATH%; C:\jdk1.4\bin
```

```
set CLASSPATH= . ; C:\jdk1.4\lib
```

也可把上面两行加入到自动执行的批处理文件 C:\AUTOEXEC.BAT 中，或在“控制面板——系统——高级——环境变量”中分别设置变量 PATH 和变量 CLASSPATH 的值为上述赋值号“=”后面的内容。

安装完标准版 J2SE 1.4 后，便可在命令提示符下使用 Java 的编译命令 javac 或运行命令 java，编译或运行 Java 的应用程序。

注意：进入命令提示符（即 MS-DOS 方式）的方法：由 Windows 操作系统的开始菜单——程序——附件——命令提示符等进入，或开始菜单——运行——键入“cmd”再确定而进入。

进入黑色的命令提示符窗口后，在编译或运行您的应用程序之前，一般先要转换驱动器（改变盘符），再用 CD <目录> 进入您的应用程序所在的目录（文件夹）。

例如，假定应用程序所在的目录路径为“F:\第1章 幻灯片播放\sourceCode”，则转换盘符、进入该目录的操作如图 1-5 所示。

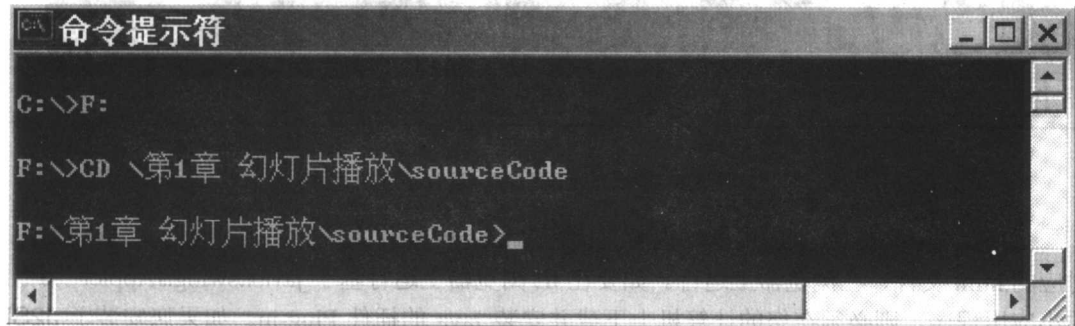


图 1-5 转换盘符、改变目录方法

进入了应用程序所在的目录后,便可用 Java 语言编译器(即编译程序)javac.exe 编译 Java 源程序。编译 Java 源程序的方法如下:

```
javac <Java 源程序文件名(含扩展名.java)>
```

注意:Java 源程序,简单情况下,可用记事本编写,也可用 UltraEdit 之类的文本编辑软件编写,或在集成开发工具环境如 JBuilder、Visual J++等下编写。

标准版 J2SE 开发环境下,采用 Java 语言解释器(即解释型运行程序)java.exe,运行 Java Application 类的经编译后的应用程序。

运行 Application 应用程序方法如下:

```
java <主类名>
```

如果是 Java Applet 小应用程序,则编译后的主类文件(或主类名)必须嵌入到一个后缀名为.html(超文本标记语言)的文件中,这时,可在命令提示符(即 J2SE 开发环境下)下,用小应用程序浏览工具 appletviewer.exe 来运行 HTML 文件,再由该文件自动调用主类运行。

在命令提示符下运行 Java 小应用程序,方法如下:

```
appletviewer <主类所嵌入的 HTML 文件>
```

例如,幻灯片播放源程序 slidePlay.java,经编译后得字节码文件 slidePlay.class,故可编写超文本标记语言文件 slidePlay.html,其内容如表 1-1 所示,其中的 width=400 和 height = 300 表示运行界面宽度为 400 个像素点,高度为 300 像素点。

表 1-1 slidePlay.html 代码

```
<html>
<applet code= "slidePlay.class" width = 400 height = 300>
</applet>
</html>
```

用 appletviewer 命令运行该 HTML 文件,具体见图 1-6 所示。运行后便可得到类似图 1-1 的程序画面。

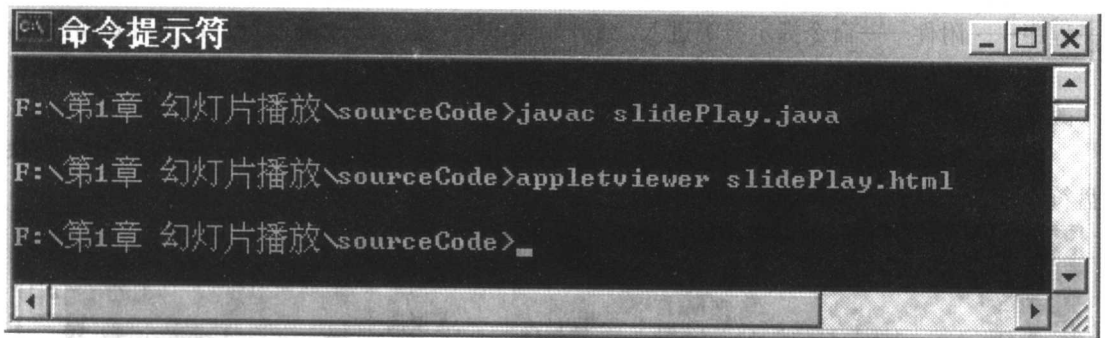


图 1-6 编译 Java 源程序、运行小应用程序

注意:Java Applet 小应用程序也可直接在 Windows 操作系统的图形界面下,通过双击其对应的.html 文件,在浏览器上运行。如要在 IE 浏览器上运行基于 javax.swing.JApplet 类派生的小应用程序,则必须在您的计算机上下载并安装 Java 的插件 Plug-in。如果您安装了 J2SE 的 1.4.2 版本的 JRE(运行阶段环境),则已包含了 Java 的插件 Plug-in。

3. JBuilder——Java 的集成开发工具

除了 J2SDK (或 JDK) 命令行开发工具外, 我们还可借助于功能强大的集成化开发工具如 JBuilder、NetBeans 等编写 Java 应用程序。在集成开发环境 (IDE) 下, 程序的编写、编译、运行是一体化的, 即集中在一个工具软件中完成的。

如果您的机器上安装了 JBuilder 9 版本, 则在其安装目录下 (例如目录 C:\JBuilder9), 已包含有 J2SDK 的内容 (目录 C:\JBuilder9\jdk1.4 中)。因此, 安装了集成开发工具 JBuilder 的计算机, 可不用单独安装 J2SDK。

1.4 图形用户界面 GUI

编写 GUI 的应用程序, 可借助可视化集成开发软件 (如 JBuilder 等), 先创建各界面元素, 设置它们的属性和布局, 再定义相应的事件过程, 以实现 GUI 与用户的交互功能。

Java 语言的基础类库 JFC 中包含若干个子库——包, 每个包中又有若干个特定功能和相互关系的类和接口。其中, java.awt 包和 javax.swing 包中含有 GUI 界面元素, 如: 按钮 (Button、JButton)、文本框 (TextField、JTextField)、文本区 (TextArea、JTextArea)、标签框 (Label、JLabel)、容器类的组件 Applet 和 JApplet 等。

其中 javax.swing 包中的组件前面一般带有字母 J, 是属于轻量级的, 这些组件或直接或间接地从 java.awt 包中的重量级组件中派生而来。

如图 1-1、1-2、1-3 所示的均是图形用户界面, 各图形界面上有一个图像、一个标签、一个文本区和两个按钮。其中的标签、文本区和按钮来源于 javax.swing 包, 属于轻量级的组件。

基于 javax.swing 包的由 JApplet 类而派生出的小应用程序, 相比基于 java.awt 包的由 Applet 类派生出的小应用程序, 在平台无关性等性能方面有更大的优势, 但是它有个缺陷, 即支持 javax.swing 包编写的小应用程序的浏览器不多, 如要在 IE 浏览器上成功运行 JApplet 小应用程序, 必须在您的计算机上下载并安装 Java 的 插件 Plug-in。

1.5 容器和组件

GUI 界面元素有: 容器、组件 (也叫控件, 即控制组件), 还有用户自定义成分, 如在 GUI 中绘制的圆、矩形等图形。

容器是容纳、存放、安排其他组件及成分的特殊控制组件。

容器之间可层层嵌套, 以构成复杂的界面系统。

容器类有: Container、Applet、Panel、Window、ScrollPane、Applet、Frame、Dialog、JApplet、JApplet、JWindow、JFrame、JDialog 等。

组件是组成 GUI 的部件, 通过对不同事件的响应来完成与用户的交互操作或组件之间的交互操作。

组件一般作为一个对象, 按一定的布局放置在容器组件内, 通常将组件对象注册给别的事件监听者对象, 以响应相应的事件处理。

常用的基本组件 (非容器组件) 有: 选择类的单选按钮、复选按钮、下拉列表; 文字处理类的文本框、文本区、标签; 命令类的按钮、菜单等。

1.6 布局

为了使图形用户界面 GUI 具有良好的平台无关性, Java 语言引入了“布局”来管理容器中组件对象的位置摆放。

管理容器中组件的编排、布置及相互关系的类, 称为布局管理器, 简称布局。

Java 的布局管理类有: `FlowLayout` (顺序布局, 也叫流式布局)、`BorderLayout` (边界布局)、`CardLayout` (卡片布局)、`GridLayout` (网格布局)、`GridBagLayout` (网格包布局) 等。此外, 没有布局管理器的容器布局称为 `null` 空布局, 也叫手工布局。

每个容器组件均对应有一个布局管理对象。根据实际应用的需要, 我们要为每个容器设置一个合适的布局。

使用布局管理器的优点是, 当容器的界面缩小或放大时, 容器上面的组件能自动调整位置或大小, 以适应版面的变化。而空布局则无法适应版面的变化。

1. `FlowLayout` 顺序布局

顺序布局是容器中的组件从左到右、从上到下的编排布局。顺序布局是面板类容器 `Panel` 及其子类 `Applet` 的缺省布局。

在容器 `r` 中设定顺序布局的方法:

```
r.setLayout(new FlowLayout())
```

顺序布局类的构造函数还可带有参数, 如:

```
FlowLayout(int align, int hgap, int vgap)
```

其中, `align` 为行对齐, 可取静态常量 `LEFT`、`CENTER`、`RIGHT`。 `hgap` 为列间的水平间距, `vgap` 为行间的垂直间距, 单位均为像素点。行组件缺省对齐为中对齐, 缺省间距为 5 个像素点。

构造函数还有带一个参数的: `FlowLayout(int align)`。

在容器 `r` 中加入组件的方法: `r.add(组件名)`。

2. `BorderLayout` 边界布局

边界布局把容器空间分为东 (右)、南 (下)、西 (左)、北 (上)、中共 5 个区域。

边界布局的例子见图 1-7 所示。

边界布局的每个区域最多只能装入一个组件, 若要在一个区域中装入两个以上的组件, 必须使用容器嵌套技术, 即在区域中先放一个容器对象, 再在上面放置组件。

构造函数有: `BorderLayout(int hgap, int vgap)`

还有无参数的构造函数, 这时, 各组件间的纵横间距为 0。

3. `CardLayout` 卡片布局

使用卡片布局的容器, 可同时容纳如卡片 (幻灯片) 般叠放起来的多个组件, 但每次只能选一个组件来显示, 如同一叠扑克牌只能显示最上面的一张一样。

例如, 设置窗格容器 `JPanelC` 的布局为卡片布局: `JPanelC.setLayout(cardLayout1)`。

卡片布局类 `CardLayout` 提供了一些方法来翻阅这些卡片 (幻灯片), 如: 下翻 `next(container)`、上翻 `previous(container)`、最后一个 `last(container)`、第一个 `first(container)` 等, 以达到前后控制的目的。

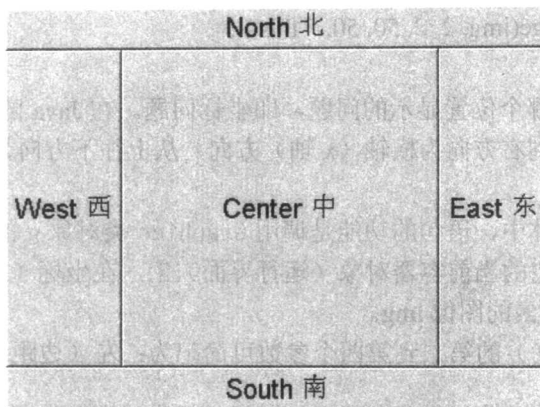


图 1-7 BorderLayout 边界布局

4. GridLayout 网格布局

网格布局将容器版面分成若干个大小相同的网格，每个网格可放置一个组件。例如计算器面板中的按钮布局。

创建网格布局对象时，可以给出网格的行数(rows)和列数(columns)。

网格布局的特点：组件排列整齐划一，且其大小相同。

如希望某个网格为空，可加入一个空的标签：add(new Label())。

将容器 r 设置成网格布局的方法：r.setLayout(new GridLayout(rows, columns))。

5. GridBagLayout 网格包布局

网格包布局是最灵活，也是最复杂的一种布局。它把容器中的每个组件按行、列放置，但各个组件所占的空间可以互不相同。

6. null 空布局（手工布局）

空布局即不使用布局管理器对象，而采用手工定位的方式来安排容器中组件的位置，类似 Visual Basic 中在窗体等容器上直接放置控件对象。

如设置标签的位置和大小为：左边距 106、顶边距 23、宽度 170、高度 26（均为像素单位），可用语句：jLabel1.setBounds(new Rectangle(106, 23, 170, 26))。

1.7 图像显示

在 Java 中，能显示的图像文件格式有：gif、bmp、jpg 等。

通过获取图像文件中的图像可生成图像类 Image 的对象，再由 Graphics 类对象的 drawImage() 方法显示图像。例如，声明图像变量 img：

```
Image img;
```

设图像文件 Cookey.gif 存放在编译后字节码.class 文件所在的目录下，则从图像文件中获取图像并赋给 img 的语句为：

```
img = this.getImage(this.getCodeBase(), "Cookey.gif");
```

显示图像使用 paint() 绘制方法，如：