

计算机实践指导系列教材

Java上机实践 指导教程

曹德胜 张玲 等编著



 机械工业出版社
CHINA MACHINE PRESS

计算机实践指导系列教材

Java 上机实践指导教程

曹德胜 张 玲 等编著



机械工业出版社

本书以高校目前普遍使用的 Java 教材为背景,针对 Java 编程的特点,精心策划,准确定位,概念清晰,深入浅出,通过一些经典例题来阐述 Java 知识,每章有多个实践,每个实践都是先给出该实践例题的运行结果,再重点分析该例题,这样可以使读者更容易地理解和掌握程序。“程序分析与注意事项”是本书的精华,详细说明了程序和操作中容易出现的错误。每章后还配有习题和对部分习题的提示。

本书可作为高等院校计算机及相关专业的 Java 语言上机实践课程的指导书,也可以作为有一定 Java 语言基础知识的自学者的参考书。

图书在版编目(CIP)数据

Java 上机实践指导教程/曹德胜等编著. —北京:机械工业出版社, 2003.10

(计算机实践指导系列教材)

ISBN 7-111-13187-8

I . J... II . 曹... III . Java 语言—程序设计—教材 IV . TP312

中国版本图书馆 CIP 数据核字(2003)第 091772 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:蔡 岩

责任印制:路 琳

北京机工印刷厂印刷·新华书店北京发行所发行

2004 年 1 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·17.5 印张·431 千字

0001—5000 册

定价:27.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68993821、88379646

封面无防伪标均为盗版

前 言

由于 Java 在面向对象的程序设计及其在网络开发中的广泛应用,引起了广大学生、计算机应用开发者的学习兴趣。不少学校已把 Java 程序设计作为计算机专业和相关专业的必修课程,为后继课程的学习打下良好的基础。为此,我们编写了这本《Java 上机实践指导教程》,以提高 Java 学习者的上机操作能力。

本书共分 9 章,第 1 章为 Java 编程环境;第 2 章为 Java 编程基础,包括数据类型、运算符和表达式、分支和循环结构以及数组和向量等基础知识;第 3 章为系统类的应用,包括输入输出类、数据类型类、字符串处理类、数学计算类和日期时间类等系统类;第 4 章为用户自定义类,包括基本类的定义、类的继承、类的多态和重载、类的修饰、包的应用及其接口技术等;第 5 章为绘图基础和通用组件,包括文字,颜色和字体的设置方式、用户自定义绘图、图像处理、标签、按钮、输入框、选择框、鼠标键盘事件;第 6 章为容器和布局,包括“Applet”、“Panel”和“Frame”等容器、“FlowLayout”、“Borderlayout”、“Gridlayout”、“Cardlayout”等布局、“Canvas”画板、菜单设计、通用对话框及其事件容器等内容;第 7 章介绍异常及多线程,包括异常的基础知识、异常的处理、用户自定义异常、多线程基础、多线程的互斥与同步等知识;第 8 章为文件和多媒体,包括文件的基础知识、顺序文件、随机文件、标准输入和输出的重定向、多媒体技术基础等;第 9 章为网络编程基础,包括网页的打开、获取本地远程计算机的名称、IP 地址和工作组、“Socket”客户端和服务器的连接及数据传送、网络通信原理等网络知识。

本书主要由曹德胜、张玲编写,参加编写的人员还有郭海文、孙改平、曹达彬、赵爱华、王加杉、周德锋、夏永恒、杨晓红、胡炜、李雨田等。

由于作者水平有限,加之计算机技术发展十分迅速,书中必然存在一些不足和错误,敬请广大读者批评指正,必要时,可发送电子邮件至 cds528@btamail.net.cn。

编 者

目 录

前言

第 1 章 Java 编程环境	1
实践 1 Java 在 DOS 环境下的编辑、编译和运行	2
实践 2 在 Visual J++ 中编辑、编译和运行 Java 程序	6
复习题 1	12
上机练习题 1	13
第 2 章 Java 编程基础	14
实践 1 数据类型	15
实践 2 运算符	19
实践 3 流程控制结构	23
实践 4 数组和向量	29
复习题 2	33
上机练习题 2	36
第 3 章 面向对象的程序设计(I)——系统类的应用	38
实践 1 输入输出类	39
实践 2 数据类型类及其转换	43
实践 3 字符串处理类	47
实践 4 数学时间类	50
复习题 3	54
上机练习题 3	55
第 4 章 面向对象的程序设计(II)——用户自定义类	57
实践 1 类的定义	58
实践 2 类的继承	62
实践 3 类的修饰	67
实践 4 类的多态	72
实践 5 类的封装	77
实践 6 接口技术	81
复习题 4	85
上机练习题 4	86
第 5 章 图形用户界面设计(I)——用户自定义绘图和常用组件的应用	87
实践 1 绘图基础	88
实践 2 基本几何图形的绘制	92
实践 3 图像的显示、标签和按钮的应用	96

实践 4 输入框	100
实践 5 选择框	104
实践 6 鼠标和键盘事件及其事件类	110
复习题 5	114
上机练习题 5	115
第 6 章 图形用户界面设计(II)——容器和布局	117
实践 1 常用容器和布局——Applet 容器、Canvas 画布和 BorderLayout 布局	118
实践 2 面板容器和网格布局——Panel 容器、CardLayout 布局和 GridLayout 布局	122
实践 3 框架容器和菜单设计——Frame 容器、下拉菜单和弹出式菜单	126
实践 4 通用对话框——Frame 容器和 Dialog 对话框	132
实践 5 事件容器——鼠标、键盘、窗体、容器、组件和焦点事件	138
复习题 6	144
上机练习题 6	145
第 7 章 异常及多线程	147
实践 1 异常的基础知识	148
实践 2 异常处理	152
实践 3 自定义异常	159
实践 4 多线程入门	166
实践 5 多线程应用	170
实践 6 多线程的互斥与同步	173
复习题 7	179
上机练习题 7	180
第 8 章 文件和多媒体	181
实践 1 文件管理	182
实践 2 顺序文件	190
实践 3 标准输入和输出的重定向	196
实践 4 随机文件	201
实践 5 多媒体基础	207
复习题 8	214
上机练习题 8	215
第 9 章 网络编程基础	216
实践 1 打开一个网页	217
实践 2 InetAddress 类	220
实践 3 Socket 网络编程建立连接	224
实践 4 Socket 的通信	228
复习题 9	232
上机练习题 9	233
附录	234

第 1 章 Java 编程环境

Java 语言是 Sun 公司在 1991 年开发的一种编程语言。由于用 Java 语言编写的程序可在不同计算机以及不同操作系统平台上运行,当 1993 年万维网出现后,Java 便随之成为 Web 编程的流行工具。

计算机程序必须被翻译成计算机处理器能够识别的机器代码后才能被执行。在 VB 和 C++ 等编程语言中,这个工作由编译器来完成。编译后形成的机器代码称为可执行的二进制映像文件。不同的处理器能够识别的机器代码是不同的。例如,用 C++ 语言编写的程序,在 IBM 兼容机、Windows 操作系统平台上编译后,就只能在 IBM 兼容机、Windows 操作系统平台上运行,在苹果机上必须重新编译才能运行。这对于网络上种类繁多的机器和操作平台来说极为不便。

Java 语言很好地解决了这个问题。它通过编译器,把 Java 源代码文件(扩展名为 .java)翻译成被称为字节码的中间层代码文件(扩展名为 .class)。与 Java 源代码一样,字节码与计算机系统无关,同一个字节码文件可以被任何计算机所使用。要在一个计算机系统中运行一个在其他计算机上编写的 Java 程序,可以下载要执行程序的字节码文件并把它交给本地的 Java 虚拟机。Java 虚拟机可以把这个字节码文件翻译成本系统的机器码,最后在本机上就可以运行这个程序了。

目前,大多数浏览器都捆绑了 Java 虚拟机,这样,当装载一个运行 Java 程序指令的 Web 页面时,浏览器会自动下载字节码文件并在虚拟机上运行它。

虚拟机是一个解释器,属于先解释后执行方式。它先翻译字节码,然后再执行翻译所生成的机器码。为了提高运行速度,Java 为每个系统都提供了可以直接把字节码文件编译成可执行的映像文件的即时编译器(JIT),并将它们捆绑在一些 Web 浏览器中。

因此,要设计 Java 程序,首先要有一个编辑程序源代码的编辑器,并将编辑完的程序代码保存为扩展名为 .java 的文件。然后通过一个名字为 javac.exe 的编译器将程序源代码文件编译为与源代码文件同名但扩展名为 .class 的字节码文件。对于网络上要运行该 Java 程序的机器,只需获得程序的字节码文件,并在机器上安装有 Java 虚拟机即可。

目前流行的 Java 开发工具包括 JDK、Visual J++ 等。它们可以帮助开发者编译、运行和调试程序。如果计算机的配置不高,可使用 JDK。JDK 可在 DOS 环境下编辑、编译程序,该软件可从 <http://www.javasoft.com> 网站下载。如果计算机的配置较高,可使用 Visual J++,使用它编译程序更加方便。



实践 1 Java 在 DOS 环境下的编辑、编译和运行

实践结果

本例介绍 Java 程序的 DOS 运行环境,运行结果如图 1-1 所示。

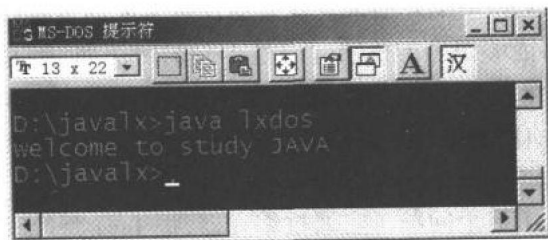


图 1-1

实践目的

学习如何在 DOS 下编辑、编译、解释 DOS 环境下的 Java 程序。

Java 程序在 DOS 下的运行过程如下:

(1) 编辑:因 Java 具有小型化、平台无关性的特点,其代码可在任何一种编辑器中编辑。在保存文件时,文件的扩展名需用 .Java。推荐用 Edit 编辑器编辑 Java 程序代码。

(2) 编译:程序编辑结束后,必须将程序转换成二进制代码,使得机器能够识别。在 Java 中,此解释转换过程称为编译。经编译后的 Java 程序,其文件的扩展名为 .class。在 DOS 下可使用 Java Development Kit(简称 JDK)的 Javac 编译器对程序进行编译。

(3) 解释:经编译后的 Java 程序,即 class 文件,还不能运行,须通过解释器对程序进行解释。在 DOS 环境下解释,可通过 JDK 的 Java 解释。

在 DOS 环境下编辑 Java 程序代码

(1) 在 DOS 提示符下键入: `c: \ > edit` (下划线部分为用户输入的命令,表示按回车键)。

(2) 在打开的 Edit 编辑器文本编辑窗口中输入下面的程序代码,如图 1-2 所示。

```
//
//1xdos
//
public class 1xdos
{
```

```
public static void main (String[] args)
{
    System.out.print ("welcome to study JAVA");
}
```

(注意:字符的大小写一定要与以上所示的代码相同。)

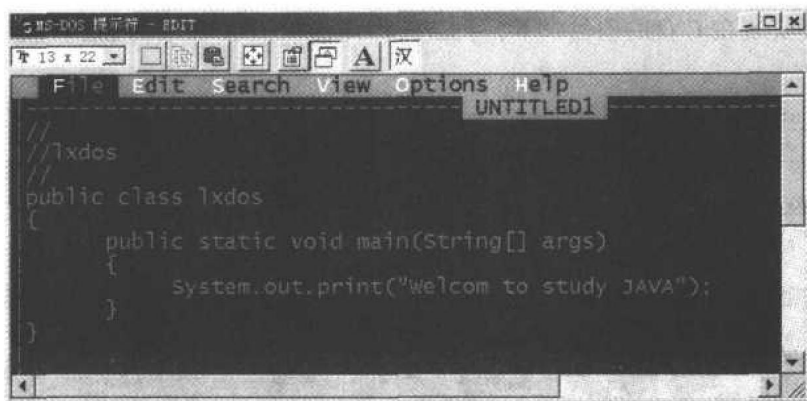


图 1-2

(3) 依次单击【File】、【Save】菜单命令,在弹出的窗口中,在 Directories 下面的列表中选择文件保存的位置(这里选择 D 盘 javalx 个人文件夹),这时,在 File Name 中会显示所选文件夹,接着输入文件名(这里为 lxdos),如图 1-3 所示。

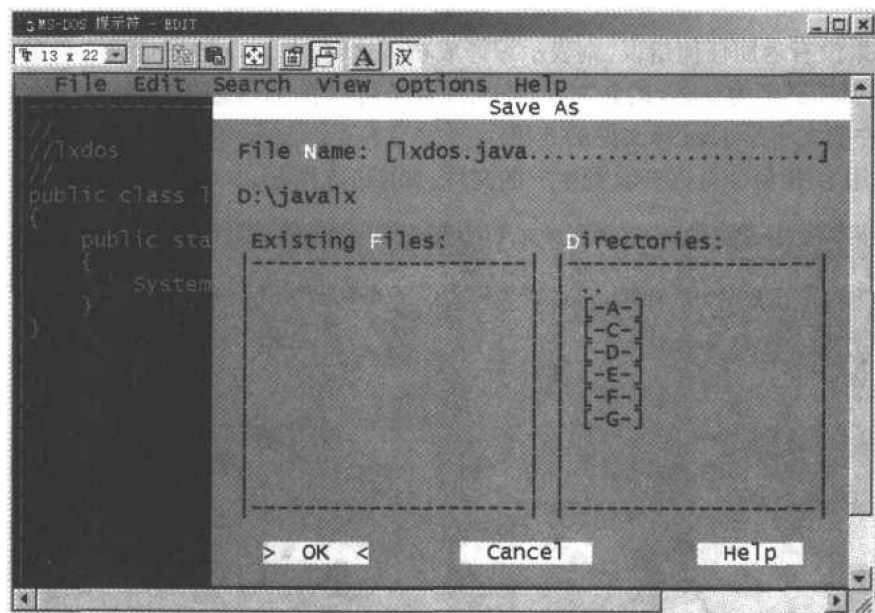


图 1-3

(4) 单击【OK】键,完成 Java 程序的编辑操作。

(5) 依次单击【File】、【Exit】菜单命令,退出 Edit 编辑器并做相应的保存。

在 DOS 环境下编译 Java 程序代码

(1) 为了使编译、解释时更加方便,首先在系统配置文件的 Path 变量中设置其路径。在 DOS 提示符状态下输入如下命令:

```
c: \ > path [JDK 安装目录] \ bin; %path%
```

例如 JDK 安装在 C 盘的根目录下,而 Javac 编译器在 JDK 目录的 bin 目录下。这里输入如下命令:

```
c: \ > path c: \ JDK \ bin; %path% ✓
```

(2) 为编译上面的 lxdos.java 程序代码,执行如下操作:

```
c: \ > javac d: \ javalex \ lxdos.java ✓ (必须输入扩展名)
```

(3) 编译完成后,如果程序有错误,则显示错误提示,如图 1-4 所示。

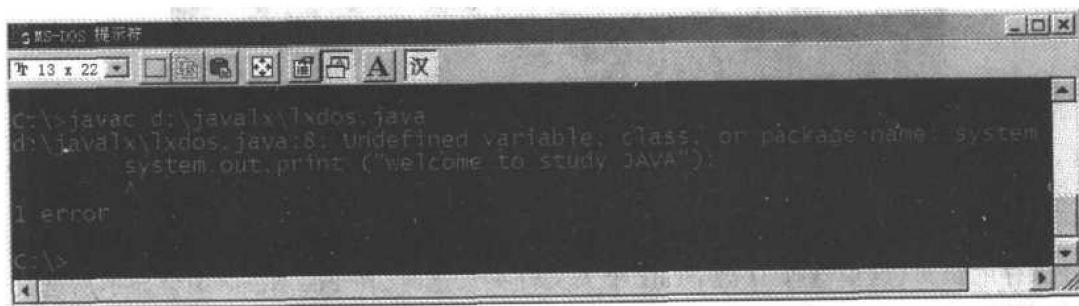


图 1-4

(4) 如果没有返回任何错误,则表示编译成功。这时,在 Java 程序代码文件的同一目录下会生成同名但扩展名为 .class 的新文件,该文件就是可以在 Java 虚拟机中运行的 Java 字节码文件,该文件的大小是 Java 源文件的 3 倍。

(5) 使用 DIR 命令可以查看到生成的文件,如图 1-5 所示。

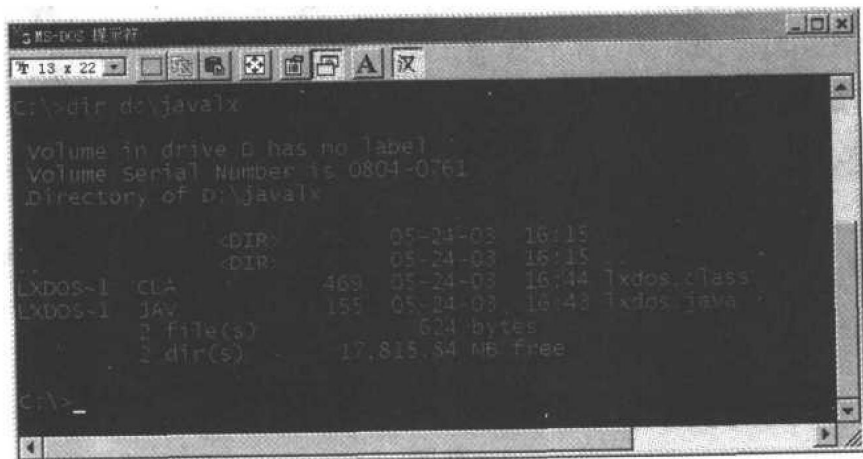


图 1-5

在 DOS 环境下运行 Java 程序

编译成功后,进入用户工作目录下后,执行下面的命令运行 Java 程序:

D: \ java\ > java lxdos (不输入扩展名)

这时在屏幕上会显示一条“welcome to study JAVA”,如图 1-1 所示。

程序分析与注意事项

(1) 下载并安装 JDK 后,默认方式下它会被安装在 C 盘的 JDK 目录下。其中的 bin 目录包括编译 Java 程序的 javac 编译器和运行字节码文件的 java 运行器。

(2) JDK 编辑器下载并安装到用户计算机后,需要使用“Path”命令设置其路径,使得编译、解释时更加方便。其格式为:“path [JDK 安装目录] \ bin; %path%”。

(3) 在编译 Java 程序时,如果显示错误信息,可考虑下面的因素:

- 根据错误提示的位置,修改程序代码。
- 由于 Java 区分字母的大小写,在编辑 Java 程序代码时,字母的大小写是否正确。
- 源文件中语句“public class lxdos”中的 lxdos 必须与主文件名相同。
- 保存代码源文件时一定要使用扩展名 .java。
- 如果系统不识别 javac 命令,可能是没有正确安装 JDK,或者没有正确设置系统路径。查找 JDK 所在的安装目录,找到后,用带路径名的编译命令试一试。如果成功,则修改系统变量 Path。否则重新安装或下载 JDK。
- 修改的系统变量 Path,如果不加到 autoexec.bat 文件中,每次启动 DOS 后,都要重新设置一次。
- 在编译 Java 源程序时,一定要输入文件的扩展名 .Java。否则将不能编译。
- 运行 Java 程序时,要将当前的工作目录设置为 Java 程序所在的目录。在输入“Class 文件名”时,不能输入文件的扩展名 .class,否则将不能解释。

(4) 如果用户计算机中安装了 Windows 操作系统,可使用其中的记事本编辑 Java 程序。

(5) 本例程序代码中:

- “//”开始行是程序注释行。编译时对此部分内容忽略。
- “public class lxdos”声明公有类 lxdos。
- “{”是类和方法开始的标记。
- 在“public static void main(String[] args)”中,main 为程序执行入口。String[] args 为定义字符串类型的数组。注意:其中的 String 中的第一个字母一定要大写。
- “System.out.print(“welcom to study JAVA”)”将字符串“welcom to study JAVA”输出到标准输出流中。
- “}”是类和方法结束的标记。



实践 2 在 Visual J++ 中编辑、编译和运行 Java 程序

实践结果

本例介绍在 Microsoft Visual J++ 6.0 中如何编辑、编译和运行 Java 程序。运行结果如图 1-6 所示。

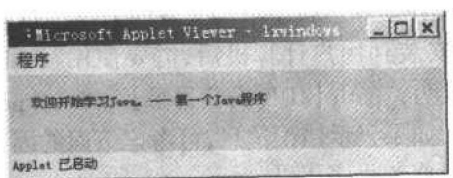


图 1-6

实践目的

了解如何用 Microsoft Visual J++ 6.0 编辑器编辑、编译、解释 Windows 环境下的 Java 程序。

在 Microsoft Visual J++ 6.0 中创建 Java 程序

(1) 依次单击【开始】、【程序】、【Microsoft Visual J++ 6.0】、【Microsoft Visual J++ 6.0】菜单命令,打开如图 1-7 所示的启动界面。

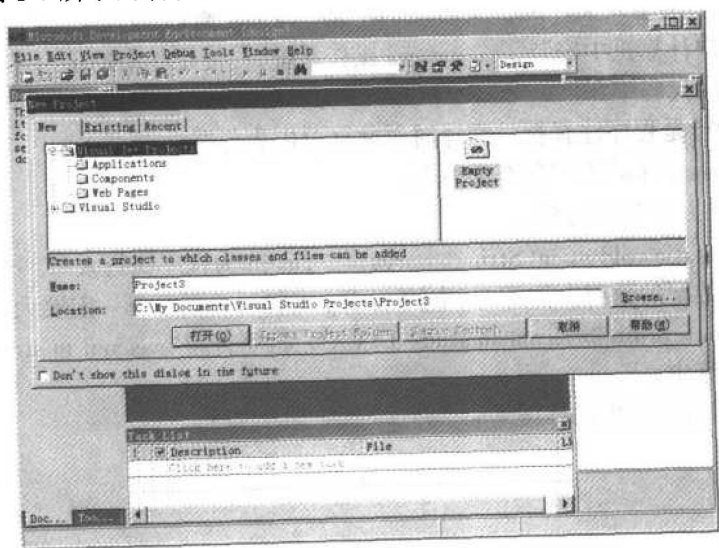


图 1-7

(2) 在“New Project”对话框中选择创建 Java 程序项目所在的文件夹和文件名,如图 1-8 所示,单击【打开】按钮。

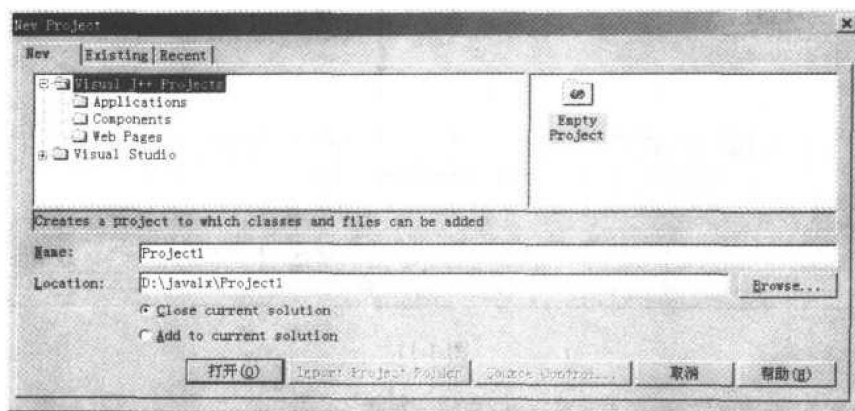


图 1-8

(3) 这时,进入 Microsoft Visual J++ 6.0 窗口,如图 1-9 所示。

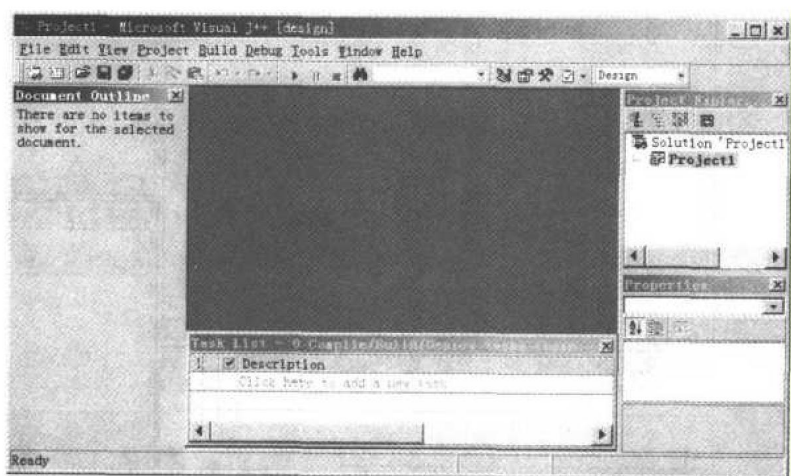


图 1-9

(4) 在窗口右侧的“Project Explorer”小窗口中,用鼠标右击新创建的 Java 程序项目,如“Project1”,在弹出的快捷菜单中依次选择【Add】、【Add Class】选项,如图 1-10 所示。

(5) 在打开的如图 1-11 所示的“Add Item”对话框中,在“Name”文本框中为要编写的程序代码文件命名,如 lxwindows,然后单击【打开】按钮。

(6) 这时,在 Microsoft Visual J++ 6.0 窗口中会出现一个“lxwindows.java”的编辑窗口,如图 1-12 所示。

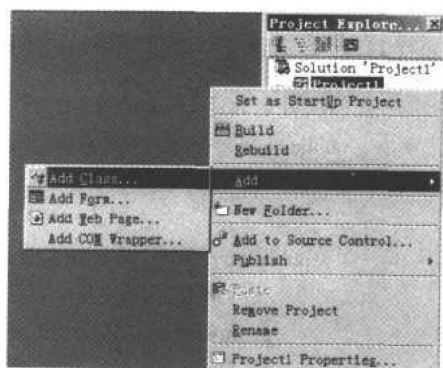


图 1-10

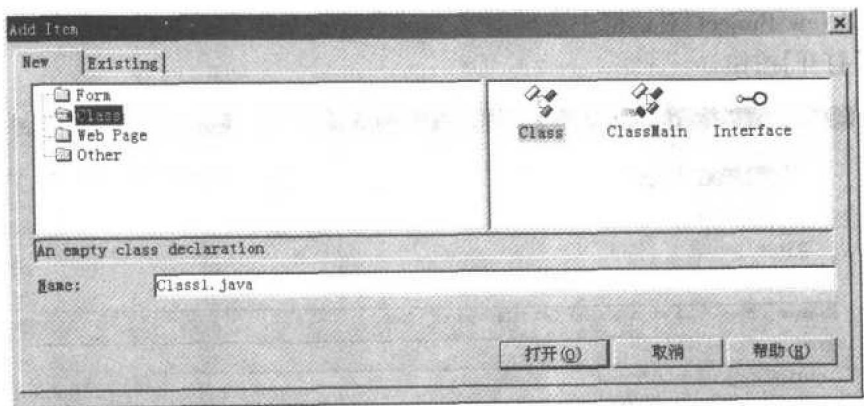


图 1-11

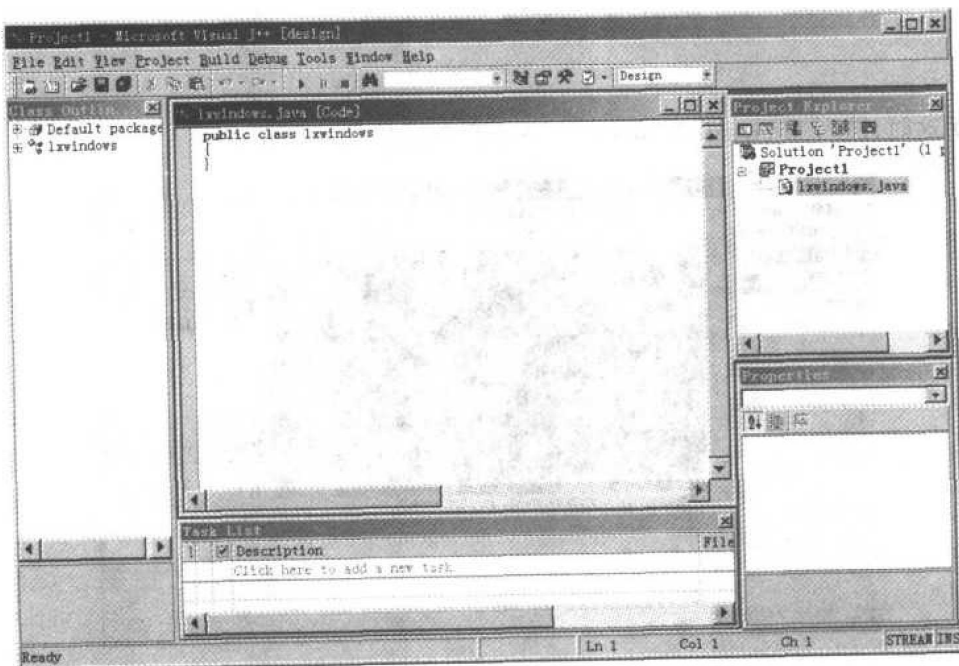


图 1-12

(7) 在编辑窗口中删除默认的代码,然后输入如下的程序代码,如图 1-13 所示。

//文件名为:lxwindows.java

```
public class lxwindows extends java.applet.Applet
```

```
{    //“Applet”小程序,可用“JavaAppletViwer”解释器解释。
```

```
    public void paint(java.awt.Graphics g)
```

```
    {    //调用“piant( )”方法,向浏览器中绘制文字或图形
```

```
        g.drawString("欢迎开始学习 Java。——第一个 Java 程序",2,0);
```

```
        //“drawString( )”方法,向浏览器中绘制文字
```

```
    }
```

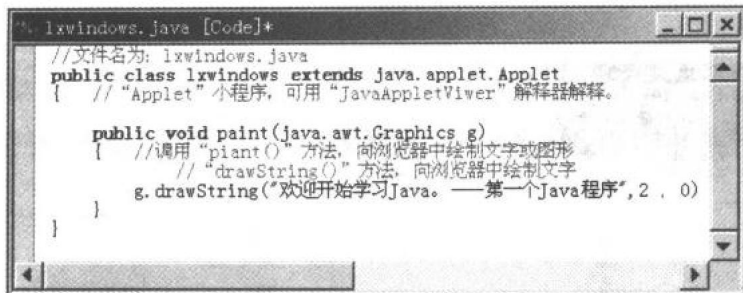


图 1-13

(8) 编辑完成后,依次单击【Build】、【Build】菜单命令,编译当前的代码。如果程序有错,会在“Task List”窗口显示错误的信息。

(9) 如果编译正确,按【F5】键,或单击工具栏上的【Start】按钮,会打开如图 1-14 所示的“Project1 Properties”对话框。

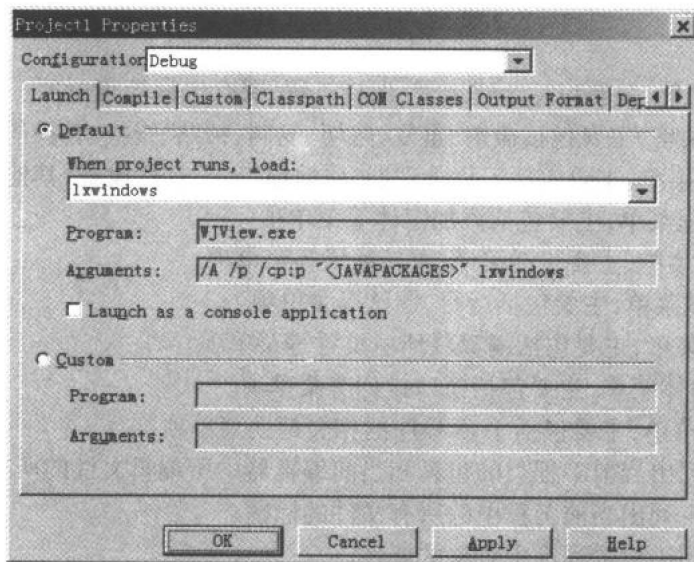


图 1-14

(10) 使用默认设置,单击【OK】按钮后,开始运行程序,这时,在屏幕左上角显示如图 1-15 所示的只有一个标题栏的 Java 程序运行窗口。

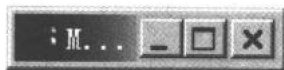


图 1-15

(11) 调整窗口大小,就会显示如图 1-6 所示的运行结果。



程序分析与注意事项

(1) Microsoft Visual J++ 6.0 是一个很好的 Java 程序的编辑环境,如图 1-16 所示。其窗

口包括如下内容:

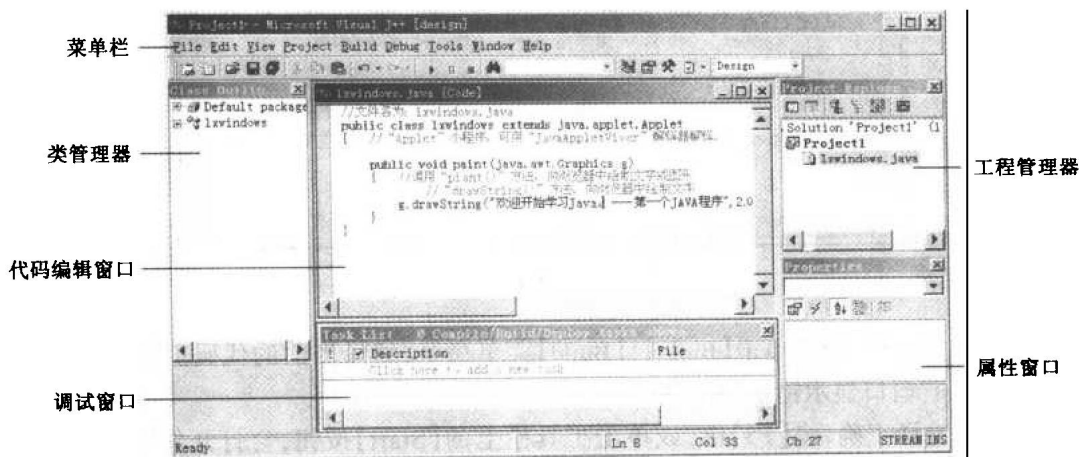


图 1-16

1) 菜单栏:

- 【File】文件菜单, 主要包括新建工程、保持工程、保存工程等菜单项。
- 【Edit】编辑菜单, 主要包括撤销、重复、剪切、复制、粘贴、查找、替换等菜单项。
- 【View】视图菜单, 主要包括工程管理器、属性窗口、工具栏设置、其他窗口等菜单项。
- 【Project】工程菜单, 主要包括添加窗体等菜单项。
- 【Build】编译菜单, 主要包括编译、重新编译等菜单项。
- 【Debug】调试菜单, 主要包括运行、调试等菜单项。
- 【Tool】工具菜单, 主要包括编辑器环境设置等菜单项。
- 【Window】窗体菜单, 主要包括窗口排列等菜单项。
- 【Help】帮助菜单, 主要包括 Java 程序的相关帮助菜单项。

2) 类管理器: 列出当前工程类的包裹和当前编辑器中所编辑文件的所有类的属性。

3) 工程管理器: 列出当前工程中的所有文件的标题。

4) 属性窗口: 用于设置程序中对象的属性。

5) 代码编辑窗口: 用于在程序中编写具体的程序代码。

6) 调试窗口: 用于调试程序时的相关信息。

(2) 本例程序代码中:

- “public class lxwindows extends java.applet.Applet”表示声明公有类 lxwindows。由于所有的 Java 小应用程序都是 Applet 的派生类, 所以可使用 extends 关键字来显式声明。
- “public void paint(java.awt.Graphics g)”表示调用 paint 方法, 作用是在屏幕上显示其中的内容。
- “g.drawString(“欢迎开始学习 Java。——第一个 JAVA 程序”, 2, 0);”表示在屏幕(2, 0)处输出字符串“欢迎开始学习 Java。——第一个 JAVA 程序”。

(3) 注意: 在 DOS 环境下的 JDK 中, 要在屏幕输出字符等内容所使用的方法, 与在 Windows 中的 Visual J++ 中所使用的方法不同。

- 在 DOS 环境下的 JDK 中, 要在屏幕输出字符等内容, 使用的基本框架是:

```
public static void main (String[] args)
{
    System.out.print ("字符串");
}
```

- 在 Windows 中的 Visual J++ 中,要在屏幕输出字符等内容,首先在声明公用类的同时,要使用 `extends java.applet.Applet` 调用 Applet 类,并且使用下面的基本框架:

```
public void paint(java.awt.Graphics g)
{
    g.drawString("字符串", x 坐标, y 坐标);
}
```

(4) 后续的实例中,如无特殊声明,调试和运行的 Java 程序均是在 DOS 环境下的 JDK 中进行的。