新版

21世纪

高职高专系列教材

# Java程序设计

◎秦 毅 主编

提供电子教案增值服务

机械工业出版社
CHINA MACHINE PRESS

21 世纪高职高专系列教材

# Java 程序设计

主编 秦 毅

参编 郑 杰 吴 蔚 汪荣斌 刘 佳

机械工业出版社

本书分为 12 章，主要内容包括 Java 的基本语法、异常处理、面向对象程序设计、字符串处理、Java 图形用户界面、Java 在 Web 上的应用、线程、Java 数据库连接等。本书各章均附有实训和习题，其中第 6～12 章的实训围绕一个项目展开，每章实现一定的功能，逐步完善项目，最终构成一个完整的应用实例。

本书可作为高职高专院校计算机软件技术课程的教材，也可作为 Java 编程爱好者的自学参考书。

**图书在版编目（CIP）数据**

Java 程序设计 / 秦毅主编. —北京：机械工业出版社，2007.6
（21 世纪高职高专系列教材）
ISBN 978-7-111-21808-1

Ⅰ. J… Ⅱ. 秦… 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（2007）第 098594 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：刘亚军
**责任印制：杨　曦**

**北京市朝阳展望印刷厂印刷**

**2007 年 9 月第 1 版·第 1 次印刷**
184mm×260mm · 17 印张 · 417 千字
0001－5000 册
标准书号：ISBN 978-7-111-21808-1
定价：24.00 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）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

# 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，并有多个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 语言于 1995 年登上程序语言的历史舞台，并在 Internet 的应用上一举成功。Java 的纯面向对象和类 C 的语法对于初学者来说，既容易上手，又非常实用。本书从把 C++程序转换成 Java 程序入手，循序渐进地让读者了解 Java 的整体结构和系统的编程知识。

在本书中，或多或少地贯穿着一些 C 或 C++语言的特性，并且与 Java 中对应的特性相比较，使对 C/C++有过接触的读者在阅读时能得到更多的方便，同时也能正确地区分不同语言之间的界限。

本书分为 12 章，前 4 章讲述了 Java 的基本语法；第 5 章提出了异常处理的概念，希望读者在后面的章节能够灵活地使用异常，掌握异常处理；第 6 章引出了面向对象的概念，并对 Java 面向对象程序设计进行了较为全面的介绍；第 7 章和第 8 章讲述了字符串和流的概念；第 9 章展示了 Java 不同于 C/C++的图形用户界面，通过 Java 提供的抽象窗口工具包中的类来构建读者所需要的图形组件，通过非可视化编程工具来实现所想即所得；第 10 章讲述了 Java 在 Web 上的应用；第 11 章提出了线程的概念，使读者能够把相关操作系统的理论运用到自己的实际应用中去；第 12 章把 Java 和数据库联系起来，介绍 JDBC。

本书各章均附有实训和习题，其中第 6～12 章的实训围绕一个项目展开，每章实现一定的功能，逐步完善项目，最终构成一个完整的应用实例。

为方便教学，本书为教师提供电子教案，其中还附有习题参考答案，需要者可在机械工业出版社网站 http:// www.cmpedu.com 下载。

由于作者水平有限，书中难免存在不妥和肤浅之处，敬请读者指正。

作　者

# 目　录

# 第1章 Java简介

## 本章要点

- Java的由来和发展历史
- Java的体系结构及开发环境的配置

## 1.1 Java的历史进程

认识Java之前，必须先了解Internet和C++。

Internet起始于20世纪60年代后期的ARPAnet（ARPA，美国国防部高级研究计划局），而直到80年代仍然是Internet发展的早期。当时，人们甚至可以用一张地图表示Internet上的路由器连接路线。1989年3月，出现了万维网（World Wide Web，WWW）的概念，这对于Internet的发展起了一个较大的推动作用。1994年，Internet的迅猛发展与WWW应用的快速增长，为浏览器的广泛使用带来了契机。

C++把程序员从结构化编程带向了面向对象编程，作为编译型语言，把效率与泛型和面向对象的特性结合起来。

注意：C++通过一个能够保存函数指针的泛型模板类，可以使代码变得简明扼要。

这里还要提到一个人——James Gosling（图1-1）。

20世纪80年代，James Gosling决定利用面向对象编程技术的优势，但是他认为，C++并不合适：实践证明C++很复杂，用户很难上手运用。到1991年8月，Gosling开发了自己设计的语言，他为之命名“Oak”（橡树）。Oak就是Java最开始的名字。

早在1990年12月，Sun公司就由Patrick Naughton、Mike Sheridan和James Gosling三人成立了一个叫做Green Team的小组。这个小组的主要目标是要发展一种分布式系统架构，使其能在消费性电子产品，如PDA、手机、信息家电等操作平台上运行。在隔年，也就是1992年的9月3日，Green Team发布了一款名叫Star 7(*7)的机器（图1-2）。它可通过动画触摸屏式的用户接口来控制其他电子设备，它有点儿像现在人们所熟悉的PDA（Personal Digital Assistant，掌上电脑）。不过，它确有着比PDA还强大的功能：具有无线通信（wireless network）、5in彩色的LCD和PCMCIA接口等。现在市面上的PDA在功能上都不及它，更不要说是早在10年前那个计算机还不普及的时代了。

Oak就是在那时诞生的，主要的目的当然是用来撰写在Star 7(*7)上的应用程序。之后，Java语言所提供的一些特性，如安全性、网络通信、垃圾回收等，其实在Oak中就已经具备了，它已是一种相当优秀的程序语言。

1994年，也就是Internet飞速发展的同一年，Sun公司投资500万美元开发一个项目，此项目的所有东西都将是围着WWW打转，不管最后的结果怎样，这个项目必须把Web的

特征作为自身最大的闪光点。Gosling 和他的同事接受了这个任务，他们要把 Oak 带入 Internet。1995 年 1 月，所有的事情都基本得到了解决，惟一的一点就是，Oak 这个名字无法成为一个注册商标，因为当时已经有了一种名为 Oak 的语言，而且已抢先注册。于是，Gosling 和他的同事们希望有一个名字能够表达他们此刻的兴奋和激动，也就是这个时候，一名从印度尼西亚度假归来的同事带来爪哇岛上的咖啡作为纪念的礼物，灵感就从这迸发出来了。“Java”被提出了，正式被注册成面向对象程序语言中的一员。

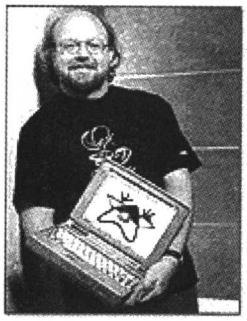


图 1-1　Java 之父——James Gosling

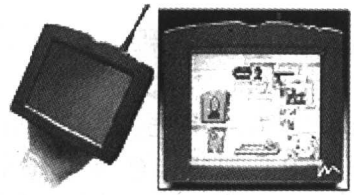


图 1-2　Star 7(*7)的展示图

同年，用 Java 语言编写的 HotJava 浏览器以及 Applet（Java 小应用程序）在 Web 上的应用，给 Java 语言带来了无限生机，使 Java 逐渐成为 Internet 上受欢迎的开发与编程语言，Microsoft、IBM、HP、Novell、Apple 等一些著名的计算机公司纷纷购买了 Java 语言的使用权。在 1995 年，Java 被美国的“PC Magazine”杂志评为年度十大优秀科技奖。

下面回顾一下 Java 的发展历程：

1995 年 3 月，Sun 公司公布了 Java 的 Alpha 1.0a2 版；

1995 年 5 月，Java 的第一个办公版发布；

1996 年 1 月，Java 的第一个开发包 JDK v1.0 版发布；

1997 年 2 月，Java 的开发包 JDK v1.1 版发布；

1998 年 12 月，Java 的开发包 JDK v1.2 版发布，Sun 公司发布 Java2 平台；

2004 年 6 月，Java 的开发包 JDK v1.5 版发布，Sun 公司将其更名为 JDK5。

## 1.2　Java 体系结构

Java 作为一种程序语言，其发展借鉴了 C++的某些面向对象的理论。原本对于面向对象的程序开发，C++也是可圈可点的，而 Java 的异军突起也就借用了 C++的这一点。这里，Java 的作者在 Java“白皮书”中总结了 11 个特征来描述 Java：简单、面向对象、分布性、健壮性、安全性、结构中立性、可移植性、解释性、高效性、多线程和动态性。

对于以上的这些特征，读者可通过后面章节的学习有一个循序渐进的理解。

现在，读者应该对 Java 有一个大概的印象了。下面再提出一个新概念——Java Virtual Machine，中文译作 Java 虚拟机，简称 JVM。其实，对于 JVM 的构想，James Gosling 的团队在开发 Star 7 时就提出来了，为的是不依赖于任意一种 CPU 结构。JVM 是在一台计算机上由软件模拟也可用硬件来实现的假想计算机。它定义了指令集（相当于 CPU）、寄存器、

类文件结构栈、垃圾回收堆、内存区域。JVM 接收 Java 编译器获取的 Java 源代码解释成的字节码，即 class 文件，class 文件是 JVM 中可执行文件的格式。JVM 规范为不同的硬件平台提供了不同的解释代码规范，使得 Java 能独立于平台。

总体来说，Java 的体系结构包括两个大的环境，一个是 Java 的开发编译环境；另一个是 Java 的运行环境。开发编译环境包括编译器、帮助文档、应用程序接口（API）和一些与编译相关的工具；运行环境由 Java 虚拟机和 Java 运行时解释器组成。

### 1.2.1 安装 Java 开发环境和编译工具

#### 1. 安装 Java 编译器

本书采用 Java 开发包 v1.4 版为例来说明设置 JDK 的开发环境，操作系统为 Microsoft Windows XP。读者可以到 Sun 公司的官方网站 java.sun.com 去免费下载 JDK 的最新版本。

下面仅以 Microsoft 的操作系统为例，介绍 Java 编译器，即 JDK 开发环境的安装设置。

（1）操作系统为 Windows 9X/Me

在系统根目录下找到 Autoexec.bat 文件，通过记事本或某些文档编辑器打开，对其做以下修改：

```
set Path =…;C:\JDK1.4\bin
set ClassPath = .;C:\JDK1.4\lib
```

这里，Path 后的省略号代表原来已经存在的路径；ClassPath 后面的“.”代表在任意当前目录下都可以执行 JDK，“.”一定不能省略！

完成以上两项设置后，对 Autoexec.bat 文件进行保存，然后重新启动操作系统。

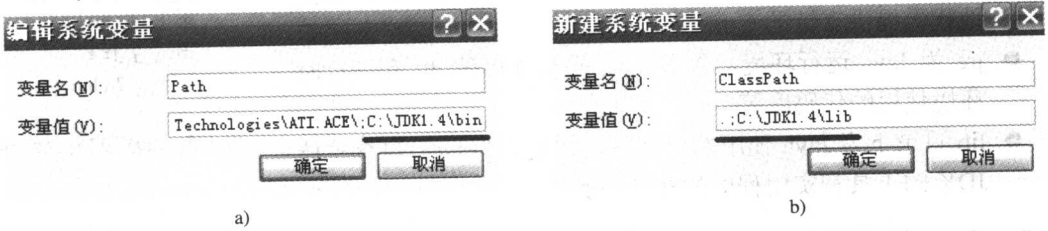
（2）操作系统为 Windows 2000/XP/2003

1）在桌面上右键单击“我的电脑”图标，进入属性设置，选择高级选项卡，对环境变量进行修改设置，或者依次通过“开始”→“控制面板”→“系统”→“高级”→“环境变量”，进入环境变量的设置。

2）在弹出的“环境变量”菜单中的“系统变量”处，选择“Path 选项”，单击“编辑”按钮，在新弹出的“编辑系统变量”菜单中的“变量值”编辑框中，把 C:\JDK1.4\bin 加在原先存在的最后一个路径后面，并且与原先存在的路径之间要用“;”隔开。

3）单击“新建”按钮，在弹出的“编辑系统变量”菜单中的“变量名”编辑框中添加新的环境变量“ClassPath”，在“变量值”编辑框中添加.;C:\JDK1.4\lib，别忘记分号前面的“.”。

4）单击“确定”按钮设置完成，效果如图 1-3 所示。




a) b)

图 1-3 设置完成的效果图

a) Java Path 设置效果图 b) Java ClassPath 设置效果图

设置完成之后，进入 Windows 9X 的 MSDOS 或是 Windows 2000 以上版本的命令提示符（CMD）窗口，输入 Javac 或 Java，按〈Enter〉键。如果出现的是其对应的参数提示信息（图 1-4），那么，Congratulation 祝贺你，Java 编译平台已经安装设置成功。

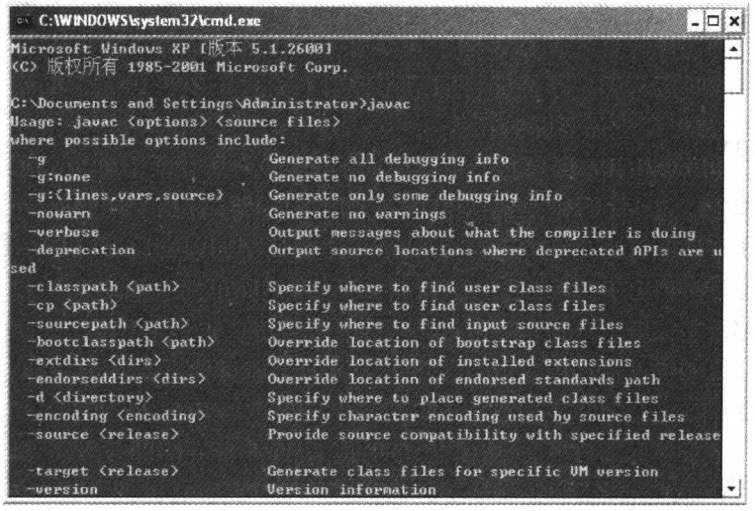



图 1-4　Java 开发环境设置成功对应的参数

### 2. Java 开发工具

本书采用的 Java 开发工具建议使用 UltraEdit，版本不限，读者同样可以通过上网下载试用版或是购买正版光盘得到该编译工具的使用权。略提一下，UltraEdit 是一款文本编辑工具，本身并没有附带 Java JDK 平台，所以必须要先安装 JDK 平台之后再使用此类的开发工具。其实，除了 UltraEdit 之外，使用 Windows 平台下的 Notepad 记事本，DOS 平台下的 Edit 都可以进行编辑。这里也推荐读者去尝试一些集成开发环境 IDE，如 Borland 公司的 Jbuilder 和 IBM 公司的 Eclipse 等。

## 1.2.2　JDK 目录结构

JDK 平台安装设置成功之后，在安装目录下的目录结构如图 1-5 所示。

- bin 目录包含 JDK 平台下的可执行文件。
- demo 目录下是含有源代码的程序示例。
- include 目录下是对 Java 本地接口和虚拟机调试接口的本地代码提供支持的 C 语言头文件。
- jre 为 Java 运行环境，包含 Java 平台使用的 dll 和运行时环境使用的代码库等。
- lib 目录下为 Java 使用的包文件，扩展名为.jar，包含支持 JDK 的工具和实用程序的非核心类等。

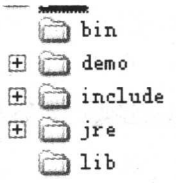



图 1-5　JDK 的目录结构

## 1.2.3　Java 程序结构

Java 虽同 C++有些相似，但是也有不同之处。就程序的结构而言，在整个 Java 程序最

开始的位置已不是 C++中的预编译，而是通过 package 关键字引导的类归档语句，其后则是由 import 引导的类导入语句，因为程序中要使用到类，所以必须放在所有自定义的类之前。

下面来看一个由 C++转换为 Java 的例子。

用 C++实现的例子：

```
#include <iostream>
#include <string>
using namespace std;
class HelloJam      //程序名称随意
{
      private :
      char name[20];
      public :
      HelloJam(){...}
      void GoodMorning(){...}
};
class HelloCplusplus
{
      public :
      HelloCplusplus(){...}
      void GoodIdea(){...}
};
class HelloJava
{
      public :
      HelloJave(){...}
      void GoodIdea(){...}
};
int main()
{
      ...
      return 0;
}
```

把上面的例子用 Java 实现：

```
package javaproject.lesson1;        //程序文件存放在 javaproject\lesson1 目录下
import java.io.*;
public class HiJames                //程序名称必须是 HiJames
{
      public static void main(String[] args)
      {
            char name[];
            name=new char[20];
            ...
      }
}
```

```
class HiCplusplus
{
    HiCplusplus(){...}
    void GoodIdea(){...}
}
class HiJava
{
    HiJava(){...}
    void GoodIdea(){...}
}
```

对于学习过 C++的读者，可以对上面给出的两个例子做个比较。没有学过的读者，请直接看下面用 Java 实现的代码，了解 Java 的程序结构以及程序文件的命名规则，这是掌握 Java 的首要条件。

### 1.2.4 编译和运行第一个 Java 程序

通过上一节的内容，读者们对 Java 的程序结构已有了一个基本的认识，为了进一步掌握 Java 的程序结构和相关格式，来看下面这个完整的例子。

所有语言都使用过 HelloWorld 程序，并以此作为其入门的经典用例，这里用 Java 来实现。其步骤如下：

1）打开 UltraEdit 编辑器，新建文档，另存为 HelloWorld.java。

2）动手写程序之前先写好文件名 HelloWorld.java。

3）第一行代码，定义公共类：

```
public class HelloWorld
```

这里，注意类名 HellowWorld 一定要和刚才的文件名相同。

4）定义主方法（C 和 C++中叫主函数，Java 里称为主方法）：

```
public static void main(String args[])
```

5）实现主方法（通过 System 类的标准输出实现）：

```
System.out.print("Hello " + args[0] + " and everyone in the world!");
```

6）保存程序文件。其最终程序如下：

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello " + args[0] + " and everyone in the world!");
    }
}
```

7）在 JDK 平台已经安装配置好的前提下，在 UltraEdit 菜单栏选择“高级”→“运行 DOS 命令”，在弹出的对话框中先设置工作目录（W），选择程序文件保存的文件目录，在命令（C）

对话框中键入 javac HelloWorld.java（如图 1-6）。单击“确定”按钮完成设置。

注意：Java 不但在文件中区分大小写，在程序运行时也严格区分文件名的大小写。

8）在自动生成的命令输出页面没有报错的前提下，重新选择“运行 DOS 命令”，在弹出的对话框的命令行直接键入 java HelloWorld James（如图 1-7）

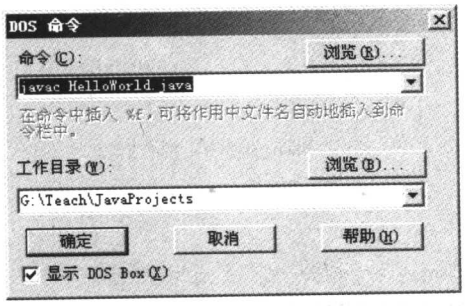



图 1-6　通过 javac 解释 HelloWorld.java 文件

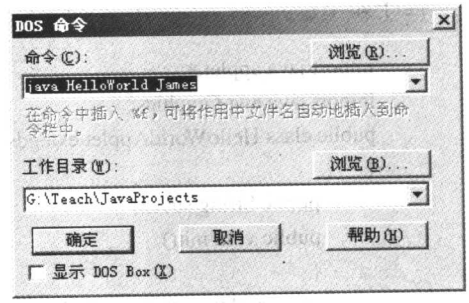



图 1-7　通过 java 执行字节码文件

程序的输出结果如图 1-8 所示。

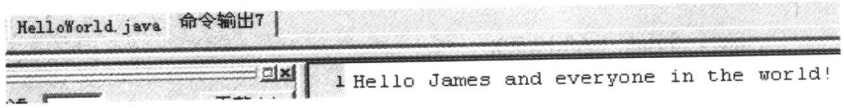



图 1-8　UltraEdit 中的结果显示

以上的步骤 7）和步骤 8）可以直接在 DOS 或 CMD 提示符下，进入程序文件保存的目录，键入编译命令

```
javac HelloWorld.java
```

产生字节码文件 HelloWorld.class。

执行命令

```
java HelloWorld "James"
```

该操作将 James 作为参数从命令行传入，引号可以省略。输出结果如图 1-9 所示。

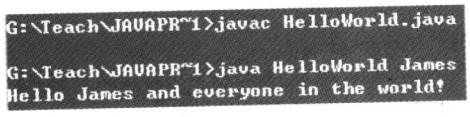



图 1-9　在 DOS 或 CMD 提示符下执行

注意：javac 命令表示对源文件进行编译，即把源文件编译成字节码文件；java 命令是 Java 为了实现跨平台机制而对字节码进行解释。所以，一般情况下可以把 Java 看成一种解释语言。Java 程序的执行过程如图 1-10 所示。

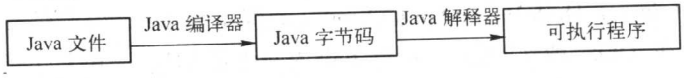



图 1-10　Java 程序的执行过程

### 1.2.5 小应用程序 Applet

Java 语言的核心并不是上一节提到的应用程序 Application，而是它的小应用程序 Applet。Applet 是怎样实现的？它和 Application 的区别到底有多大？先来看一个 Applet 的例子。按照上一节提到的步骤来实现 HelloWorld 小应用程序。

程序如下：

```
import java.applet.*;
import java.awt.Graphics;
public class HelloWorldApplet extends Applet
{
    public String strAlt;
    public void init()
    {
        strAlt=new String("Hello James and everyone in the World!");
    }

    public void paint(Graphics g)
    {
        g.drawString(strAlt,10,60);
    }
}
```

用与编译应用程序 Application 相同的方法编译小应用程序 Applet，产生了字节码文件 HelloWorldApplet.class。而之后要注意的是，以往在 C 或 C++中要实现程序必须要有主函数，即 Java 应用程序中的主方法，但是在 Java Applet 中没有 main()，所以不能用 Application 的方法来实现 Applet。

Java 是借着 WWW 迅速崛起的，所以 Java Applet 需要通过在 Internet 上使用最广泛的 HTML（Hypertext Markup Language，超文本标记语言）来实现，即把 Applet 嵌入到 HTML 中，在浏览器中实现。

新建 HTML 文件为 HelloWorldApplet.html，代码如下：

```
<html>
<applet code="HelloWorldApplet.class"
    width=200
    height=200>
</applet>
</html>
```

将写好的代码进行保存，其实现方法有两种。

- 在 UltraEdit 中打开运行“DOS 命令”窗口；在命令行键入 appletviewer HelloWorldApplet.html（如图 1-11 所示）。这与在 DOS 或 CMD 提示符下键入命令的结果是一样的。

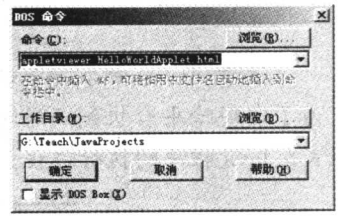



图 1-11　执行 appletviewer 命令