



新世纪高等学校计算机系列教材

Java 程序设计 实验指导及 习题解答

◎ 杨晓燕 主 编

◎ 姜遇姬 副主编



人民邮电出版社
POSTS & TELECOM PRESS

新世纪高等学校计算机系列教材

Java 程序设计实验指导 及习题解答

杨晓燕 主编

姜遇姬 副主编

人民邮电出版社

图书在版编目 (CIP) 数据

Java 程序设计实验指导及习题解答 / 杨晓燕主编. —北京: 人民邮电出版社, 2006.2
(新世纪高等学校计算机系列教材)

ISBN 7-115-13917-2

I. J... II. 杨... III. JAVA 语言—程序设计—高等学校—教学参考资料 IV. TP312
中国版本图书馆 CIP 数据核字 (2006) 第 006676 号

内 容 提 要

本书是人民邮电出版社出版的教材《Java 程序设计教程》配套用书, 也可独立作为上机用书。本书内容包括实验指导和教材习题解答两部分。实验部分内容包括: Java 工具入门及基础、Java 程序结构和类设计、Java 接口、包和线程、GUI 编程及事件驱动、图形处理、Java 数据库编程及网络编程, 并设有大作业等内容。习题解答部分列出配套教材的课后习题解答, 供读者参考。

本书力求语言简洁, 深入浅出, 注重实验设计的多样化, 实验包括验证分析、设计综合, 有些实验提供一题多解。书中完整程序在 JDK1.3/JDK1.4/JDK1.2 版本之下经过验证。

新世纪高等学校计算机系列教材

Java 程序设计实验指导及习题解答

-
- ◆ 主 编 杨晓燕
副 主 编 姜遇姬
责任编辑 邹文波
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 11.75
字数: 284 千字
印数: 1—3 000 册
 - 2006 年 2 月第 1 版
2006 年 2 月北京第 1 次印刷

ISBN 7-115-13917-2/TP • 4904

定价: 16.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223



诞生于 1995 年的 Java 语言,是目前最为流行的网络编程语言,它完全面向对象、简单高效、与平台无关、安全、支持多线程,是计算机世界的“国际语言”。

当前 Java 程序设计课程已成为高校计算机基础教学的骨干课程之一。在教育部计算机基础课程教学指导委员会制定的白皮书中,Java 课程也被列为核心课程之一。

Java 程序设计是一门实践性很强的课程,在掌握基本理论的同时,要求我们充分地利用语言提供的环境和资源,通过反复上机实验,不断积累程序设计的经验,使自己的程序设计能力不断提高。因此,Java 的实验指导书就变得特别重要。本书的编写在兼顾基本知识点的同时,根据几年来的教学实践和 Java 的主流应用,精心组织了实验项目,不仅包含大量验证实验,而且每部分都编写了若干综合实验。

本书主要内容分为实验指导和习题解答两部分。

本书倾注了资深教授姜遇姬老师 30 多年来从硬件到软件、从国内到国外从事开发和教学的经验和智慧。同时,本书也是其他几位从事 Java 一线教学和科研老师经验的结晶。全书由姜遇姬老师完成书稿的初审,由杨晓燕老师完成书稿的组织编写、最后修改、统稿和定稿。参与编写的老师还有张梁斌、陈冬亮、邓芳、刘臻。

书中所有例题及相关代码都已在 JDK1.3/JDK1.4 开发环境中经过验证。

由于编者水平所限,书中难免还存在一些缺点和错误,希望读者批评指正。联系方式 yangxy3225@163.com。

编 者

2005.12

目 录

第一部分 实验指导

第 1 章 Java 开发工具及程序设计初步	1
1.1 J2SDK 开发工具入门	1
1.1.1 J2SDK 的下载、安装	1
1.1.2 环境变量介绍和配置	5
1.1.3 JDK 开发工具简介	6
1.2 知识点	6
1.2.1 Java 程序开发步骤	6
1.2.2 Java 程序基本结构	7
1.3 本章实验	8
第 2 章 数据和数据运算	13
2.1 知识点	13
2.2 本章实验	14
第 3 章 Java 的输入/输出	18
3.1 知识点	18
3.2 本章实验	19
第 4 章 程序流程控制、算法和方法设计	27
4.1 知识点	27
4.2 本章实验	31
第 5 章 数组	46
5.1 知识点	46
5.2 本章实验	49
第 6 章 Java 程序结构和类设计	56
6.1 知识点	56
6.2 本章实验	57
第 7 章 GUI 和事件驱动	66

7.1 知识点	66
7.2 本章实验	67
第 8 章 Java 的绘图技术及线程	95
8.1 知识点	95
8.2 本章实验	96
第 9 章 JDBC 数据库连接	111
9.1 知识点	111
9.2 本章实验	112
第 10 章 网络编程	120
10.1 知识点	120
10.2 本章实验	122
第 11 章 综合大作业	140
11.1 设计算术运算测试器	140
11.2 配置 Web 应用环境, 实现 JSP 留言板	145

第二部分 教材部分习题解答

第 1 章 Java 概述习题解答	149
第 2 章 Java 语言基础习题解答	150
第 3 章 Java 输入/输出习题解答	150
第 4 章 程序流程控制结构和方法习题解答	153
第 5 章 数组习题解答	156
第 6 章 Java 类和对象习题解答	159
第 7 章 Java 接口、包和异常习题解答	168
第 8 章 字符串处理习题答案	169
第 9 章 创建用户界面习题解答	170
第 10 章 图形处理习题解答	175
第 11 章 Java 数据库连接习题解答	178
第 12 章 多媒体编程习题解答	179
第 13 章 流、多线程和网络编程习题解答	180
参考文献	182

实 验 指 导

第1章 Java 开发工具及程序设计初步

1.1 J2SDK 开发工具入门

1.1.1 J2SDK 的下载、安装

1. 下载 J2SDK

J2SDK 是 Java 2 Software Development Kit 的简称,其前身是 JDK(Java Development Kit)。目前的最新版本是 JDK 5.0。使用者可以根据运行平台的不同,下载相应的版本并设置好 path 和 classpath。这个软件包提供了 Java 编译器、Java 解释器,但没有提供 Java 编辑器,因此需要使用者自己选择一个方便易用的编辑器或集成开发工具。

开发 Java 程序的 J2SDK 或 JDK 是 Sun 公司免费提供的,可以在 Sun 公司网站上下载。下载网址——<http://java.sun.com/product>。也可以从国内的一些 FTP 站点中获得。从 FTP 上下载要比国外站点上下载快一点。下载网址可在 www.google.com 中搜索到。

从初学者角度来看,采用 JDK 开发 Java 程序能够很快理解程序中各部分代码之间的关系,有利于理解 Java 面向对象的设计思想。JDK 的另一个显著特点是随着 Java (J2EE、J2SE 以及 J2ME) 版本的升级而升级。缺点是对于从事大规模企业级 Java 应用开发非常困难,不能进行复杂的 Java 软件开发,也不利于团体协同开发。

2. 安装 JDK

由于目前大多数用户使用的是 Windows 98 操作系统,所以我们以在 Windows 98 操作系统上安装 JDK1.3.1 为例,说明其安装过程。j2sdk-1_3_1_02-win.exe 是一个自解压文件,双击它就可以解压缩,同时进行安装工作。j2sdk-1_3_1_02-win.exe 中包含了 Java Runtime

Environment, 只要安装了 j2sdk-1_3_1_02-win.exe, 浏览器就自然可以运行 Java Applet。

安装工作实际上分为两个步骤。安装程序首先会收集一些信息, 用于安装的选择, 然后才开始拷贝文件, 设置 Windows 注册表等具体的安装工作。

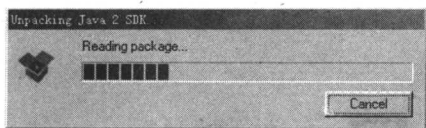


图 1-1 文件解压缩

双击 j2sdk-1_3_1_02-win.exe, 文件会自动解压缩, 如图 1-1 所示。

解压缩工作完成之后, 出现 JDK1.3.1 的安装欢迎界面, 如图 1-2 所示。

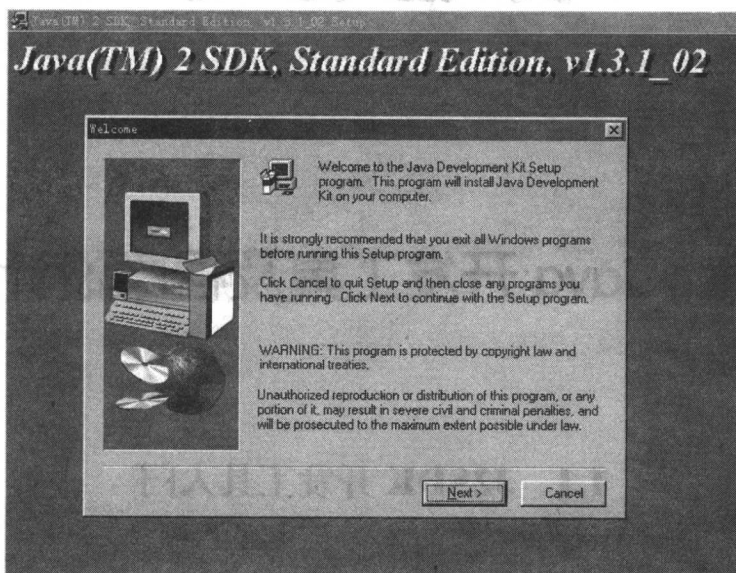


图 1-2 欢迎安装界面

单击“Next”按钮, 出现 JDK1.3.1 的许可协议, 如图 1-3 所示。

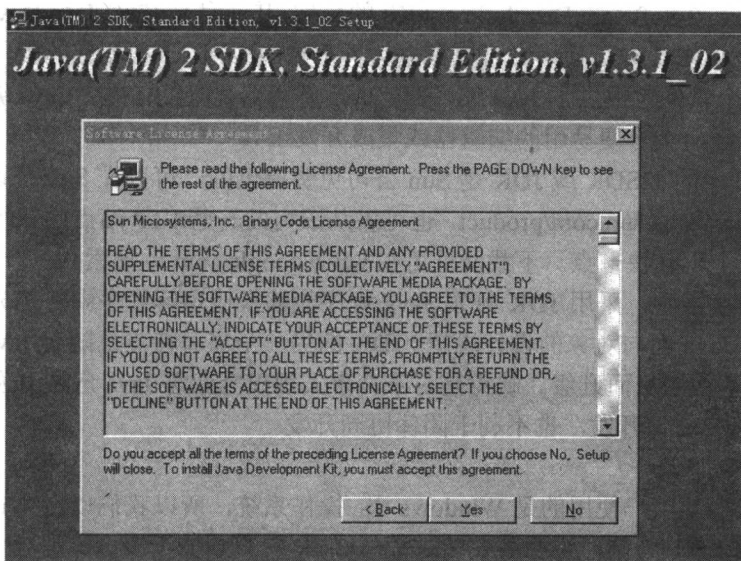


图 1-3 许可协议

单击“**Yes**”按钮，接受许可协议，安装程序会出现让用户选择安装目标路径的对话框，如图 1-4 所示。

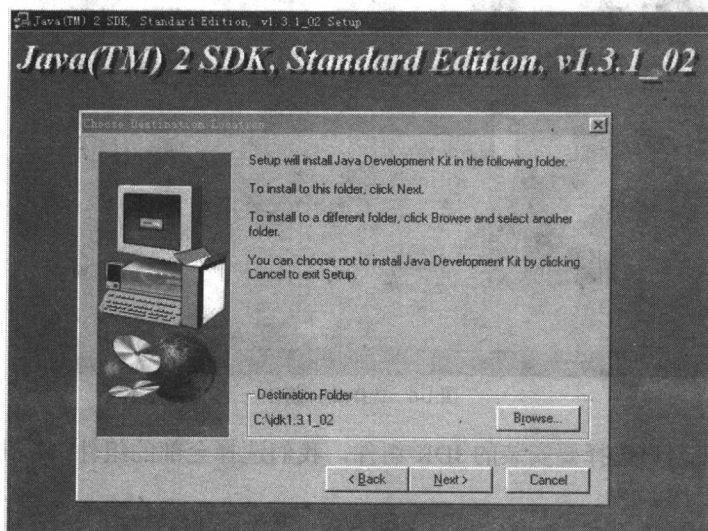


图 1-4 选择安装目标路径

在这个对话框中，系统让用户选择 JDK 程序的安装路径，系统默认的路径是 C:\jdk1.3.1_02，单击“**Next**”按钮，JDK 的所有程序就会被安装到 C:\jdk1.3.1_02 目录下。用户也可以在本对话框中单击“**Browse**”按钮，修改 JDK 程序的目标安装路径。

下一个对话框让用户选择所使用的浏览器，如图 1-5 所示。

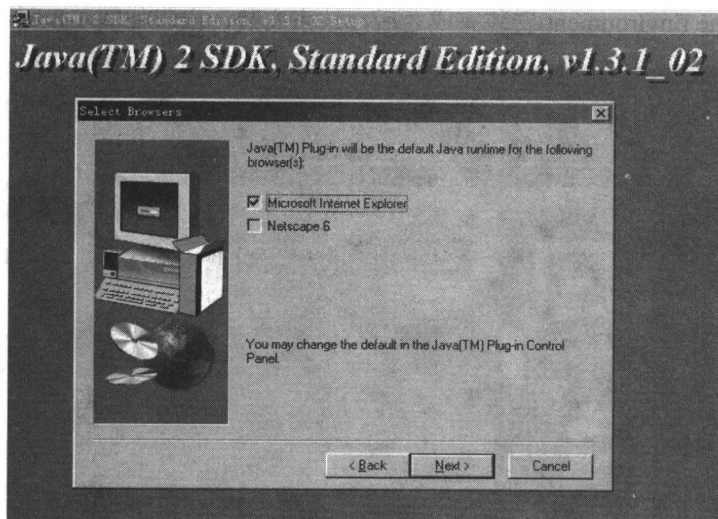


图 1-5 选择所使用的浏览器

Windows 用户常用的浏览器是 IE，所以选择“**Microsoft Internet Explorer**”复选框。单击“**Next**”按钮，进入下一个对话框，如图 1-6 所示。

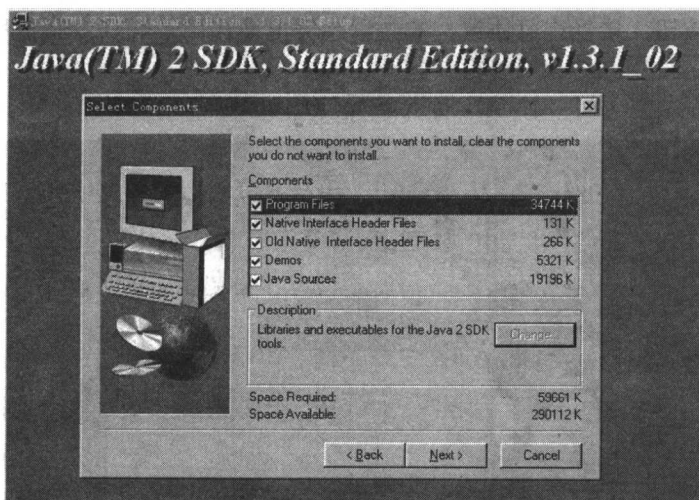


图 1-6 选择 JDK 组件

这个对话框让用户选择要安装的 JDK 组件，我们选择全部的组件，单击“Next”按钮，开始正式的安装工作。

JDK1.3.1 安装结束之后，会继续安装 Java 2 Runtime Environment，用于 Java 程序的执行。安装程序给出如图 1-7 所示的提示。

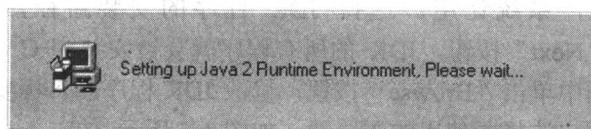


图 1-7 JRE 的安装

Java 2 Runtime Environment 安装完成之后，结束安装，出现如图 1-8 所示的对话框。

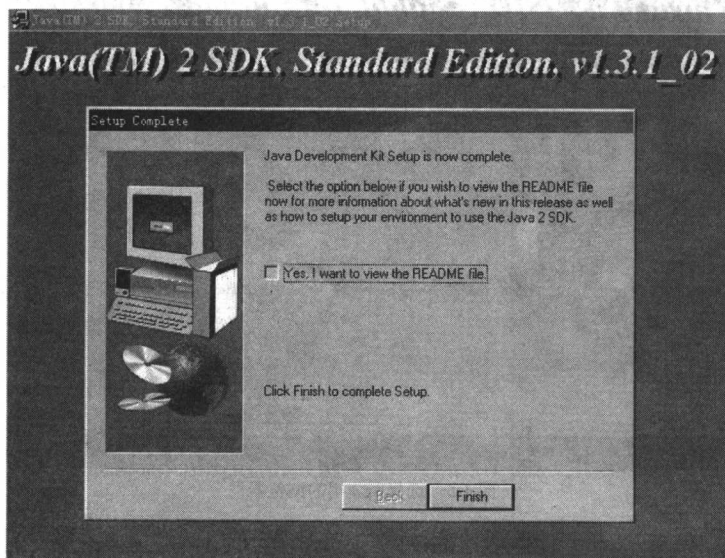


图 1-8 安装结束对话框

在这里，单击“Finish”按钮，结束安装。

1.1.2 环境变量介绍和配置

当 Java 的运行平台为 Windows 98 时，需要修改系统根目录下的 autoexec.bat 文件。使用记事本或任何文本编辑器打开文件 autoexec.bat，在该文件的最后增加如下两行：

```
set path=c:\jdk1.3.1_02\bin;%path%
```

```
set classpath=.;c:\jdk1.3.1_02\lib
```

其中，环境变量 path 和 classpath 分别指定了 JDK 命令搜索路径，和 Java 类路径。在这里我们假设 JDK 安装在 C:\jdk1.3.1_02 目录下，JDK 的所有命令都放在 c:\jdk1.3.1_02\bin 目录下。设置环境变量 path 的作用是使 DOS 操作系统可以找到 JDK 命令。%path% 表示 path 环境变量的已有的当前的字符串取值。设置环境变量 classpath 的作用是告诉 Java 类装载器到哪里去寻找第三方提供的类和用户定义的类。在 classpath 环境变量中添加的 (.) 代表 Java 虚拟机运行时的当前工作目录。JVM 和其他 JDK 工具通过依次搜索平台库、库扩展和类路径来查找类。

path 环境变量的作用是设置供操作系统去寻找和执行的应用程序的路径，也就是说，如果操作系统在当前目录下没有找到我们想要执行的程序和命令时，操作系统就会按照 path 环境变量指定的目录依次去查找，以最先找到为准。path 环境变量可以存放多个路径，Windows 下路径和路径之间用分号 (;) 隔开。

当 Java 的运行平台为 Windows 2000 时，右键单击桌面上的“我的电脑”，打开菜单中的“属性”，在出现的属性面板中选择“高级”标签，如图 1-9 所示。然后单击“环境变量”按钮，打开环境变量面板，在这里可以看到上下两个窗口，上面窗口为“某用户的环境变量”，下面窗口为“系统变量”，如图 1-10 所示。

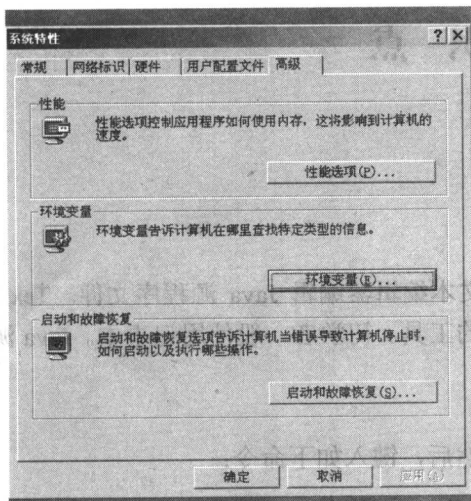


图 1-9 属性面板

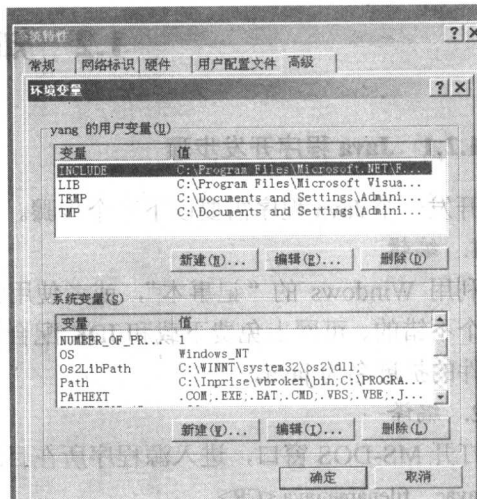


图 1-10 环境变量面板

可以在图 1-10 中的任意一个窗口进行设置，区别在于上面的窗口设置用于个人环境变量，只有以该用户身份登录时才有效，而下面窗口中的设置则对所有用户都有效。以设置系统变量为例，单击名为“path”的变量（如果没有你想设置的环境变量选项，在“用户变量”

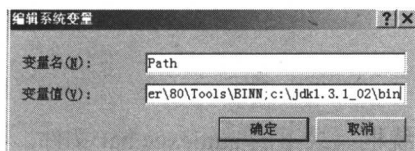


图 1-11 环境变量编辑窗口

或“系统变量”中选择“新建”来添加), 选择“编辑”, 然后在如图 1-11 所示的“编辑系统变量”窗口的“变量值”输入框中加入想设置的环境变量。注意应当在 path 原有值的末尾加上分号 (;) 然后加上 Java 编译器和解释器所在的路径 (这里是 c:\jdk1.3.1_02\bin.),

然后单击“确定”按钮。

path 环境变量设置好之后, 在 DOS 下用 cd 进入工作目录, 程序就可以编译运行。如果偶尔有问题, 可以进一步在 DOS 下, 使用设置语句: set classpath=c:\myjava (myjava 为自己的源代码存放文件夹), 再运行自己的源文件就可以了。

如果用户在安装 jdk1.3.1 时, 选择了另外的 JDK 安装路径, 则环境变量 path 和 classpath 要做相应的调整。

环境变量设置完成后, 在 DOS 窗口下, 敲入 javac 并回车后, 如果出现 javac 的用法参数提示信息, 则安装正确。也可以在 DOS 下使用 set 命令查看添加的系统路径。否则要检查环境变量设置是否正确。

1.1.3 JDK 开发工具简介

J2SDK 工具是以行命令方式应用的, 即在 Windows 操作系统的 DOS 命令行提示符窗口中执行 J2SDK 命令。在 JDK 的 bin 目录下, 存放着 Java 2 提供的一些可执行程序, 为我们开发和测试 Java 程序提供了工具。在我们的学习中, 常用的 JDK 开发工具有以下三种。

- javac.exe: Java 语言的编译器。
- java.exe: Java 程序执行引擎。
- appletviewer: JDK 自带的小应用程序浏览器。

1.2 知 识 点

1.2.1 Java 程序开发步骤

开发一个 Java 程序需要以下三个步骤。

1. 编辑

利用 Windows 的“记事本”, 或者使用其他文本编辑器编辑 Java 源程序文件。TextPad 是一个不错的、可网上免费下载和 JDK 配合使用的工具。初学者一般使用记事本。Java 源程序文件的扩展名为 java。

2. 编译

打开 MS-DOS 窗口, 进入源程序所在目录。然后, 键入如下命令:

```
javac filename.java <CR>
```

其中, filename.java 表示要编译的源程序文件名, <CR>代表回车。将源程序编译成字节码文件, 字节码文件的扩展名为 class。

3. 运行

Java 程序有两种类型: Java 应用程序 (又名 Java Application 程序) 和 Java 小应用程序

(又名 Java Applet)。

(1) 对于 Application, 打开 MS-DOS 窗口, 进入 Java 字节码文件所在目录, 在命令行状态下执行 `java filename <CR>`

可将 Application 字节码文件解释为本地计算机能够执行的指令并予以执行, 其中 filename 为以编译的 Java 字节码文件名。

(2) 对于 Applet, 可以将 Applet 字节码文件加入到一个网页 (HTML 文件), 然后用浏览器打开该网页, Applet 的运行结果就会在浏览器窗口中显示出来。我们也可以在命令行状态下执行 `appletviewer` 来查看网页, 同样会看到 Applet 的运行结果。

1.2.2 Java 程序基本结构

Java 应用程序必须得到 Java 虚拟机的支持才能够运行。Java 小应用程序则需要客户端浏览器的支持。Java 小应用程序运行之前必须先将其嵌入 HTML 文件的 `<applet>` 和 `</applet>` 标记中。当用户浏览该 HTML 页面时, Java 小应用程序将从服务器端下载到客户端, 进而被执行。

Application 的基本编程模式为:

```
class 用户自定义的类名 // 定义类
{
    public static void main(String args[] ) //调用 main()方法
    {
        方法体
    }
}
```

Applet 的基本编程模式为:

```
import java.awt.Graphics; //引入 java.awt 软件包中的 Graphics 类
import java.applet.Applet; //引入 java.applet 软件包中的 Applet 类
class 用户自定义的类名 extends Applet //定义类
{
    public void paint(Graphics g) //调用 Applet 类的 paint() 方法
    {
        方法体
    }
}
```

注意: 在关键字 `class` 前面亦可冠以关键字 `public`。

Applet 需要的 HTML 文件的最小集的格式为:

```
<HTML>
    <applet code=类名.class width= 宽度 height=高度>
</THML>
```

注意: HTML 标记包含在尖括号内, 并且总是成对出现, 前面加斜杠表明标记结束。

`<HTML>`和`</THML>`来标记 HTML 文件的开始和结束, 用`<applet>`和`</applet>`来标记 Applet 的开始和结束。必须把以.class 结尾的字节码文件名嵌入到 HTML 文件中, HTML 文件应和

字节码文件放在同一目录下。另外，HTML 对字符大小写是不敏感的，参数值可加引号也可不加。

综上所述，Applet 和 Application 是 Java 程序的两种基本类型，从源代码的角度来看，Applet 和 Application 有两个基本的不同点：

(1) 一个 Applet 类必须定义一个从 Applet 类派生的类，Application 则没有这个必要。

(2) 一个 Application 必须定义一个包含 main 的方法，以控制它的执行，即程序的入口。而 Applet 不会用到 main 方法，它的执行是由 Applet 类中的几个系统方法来控制的。

两者共同之处是：编程语法是完全一样的。

1.3 本章实验

一、实验目的

- (1) 熟悉 JDK 平台 Java 程序的开发步骤。
- (2) 通过程序验证，了解 Java Application 结构。
- (3) 通过程序验证，了解 Java Applet 结构。

二、实验练习

实验 1 创建第一个 Java Application 程序

为了编写一个 Java Application 程序，需要进行如下操作。

1. 用户需要下载和安装 J2SDK (JDK)。以 JDK1.3.1_01 版本为例，暂且把源文件放置在 JDK 的 bin 目录之下的 code 文件夹中。

2. 文本编辑器，在本例中使用记事本。从“开始”菜单项中选择“程序”→“附件”→“记事本”。当然，用户也可以选择其他文本编辑器。

3. 实验题目：实现第一个简单的应用程序：打印一行文字。

(1) 在“记事本”中编写如下源程序。

// 文件名: Welcome.java

```
public class Welcome {  
    public static void main( String args[] )  
    {  
        System.out.println( "Welcome to Java Programming!" );  
    } //结束 main 方法的定义  
} //结束类 Welcome 的定义
```

(2) 语法说明。

程序中的“//”是单行注释符，只对当前行有效，表示该行后边文字是注释行。程序员在程序中加入注释，用于提高程序的可读性，使程序便于阅读和理解。在执行程序时不会执行注释行。注释会被 Java 编译器所忽略。多行注释用“/*”开始，以“*/”结束。

Java 程序是由类或类的定义组成。类构成了 Java 程序的基本程序单位，创建一个类是 Java

程序的首要工作，而且类名要和所使用的文件名完全一样，包括字母的大小写。Java 用关键字 `class` 标志一个类定义的开始，`class` 前面的 `public` 关键字代表该类的访问属性是公共的，表示这个类在所有场合中可使用。一个程序文件中可以声明多个类，但仅允许有一个公共的类，程序文件名要与公共类的名称相同。`class` 后面是该类的类名，在本例中是 `Welcome`。

Application 中有一个显著标记就是必须定义一个 `main()` 主方法，而且应该按照例 1-1 中所示的那样来定义其修饰符和命令行参数，用关键字说明它是 `public`，静态的 `static`，无返回值的 `void`，主方法的参数是字符串类型 `String` 的数组 `args[]`。一个类中可以声明多个方法，Java 应用程序自动从 `main` 主方法开始运行，通过主方法再调用其他的方法。Java 语言的每条语句都必须用分号结束。

`System.out` 是标准输出对象，它用于在 Java 应用程序执行的过程中向命令窗口显示字符串和其他类型的信息。方法 `System.out.println` 在命令窗口中显示一行文字后，会自动将光标位置移到下一行（与在文本编辑器中按 `Enter` 键类似）。

（3）编译运行程序。

源程序编写好之后，保存为 `Welcome.java`。接下来，我们准备执行此程序。为此我们打开一个命令窗口，其界面如图 1-12 所示。

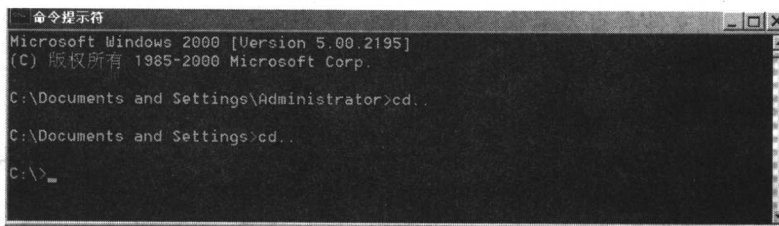


图 1-12 “命令提示符”窗口

接着，进入程序所存储的目录（假定应用程序存放在 `c:\jdk1.3.1_01\bin\code` 之下），在“命令提示窗口”中键入 `dir Welcome.java` 命令，显示文件，如图 1-13 所示。



图 1-13 显示 `Welcome.java` 文件

在“命令提示符”窗口中键入：

```
javac Welcome.java
```

执行后界面如图 1-14 所示。

```

命令提示符
2005-02-02 12:29          203 Welcome.java
          1 个文件      203 字节
          0 个目录      160,407,552 可用字节

C:\jdk1.3.1_01\bin\code>javac Welcome.java

C:\jdk1.3.1_01\bin\code>

```

图 1-14 编译 Welcome.java 文件

在“命令提示符”中键入 `dir` 命令，就可以看到 `.class` 文件，如图 1-15 所示，表明程序编译成功。

```

命令提示符
C:\jdk1.3.1_01\bin\code 的目录
2005-02-02 12:29          203 Welcome.java
2005-02-02 17:19          436 Welcome.class
          2 个文件      639 字节
          0 个目录      159,858,688 可用字节

C:\jdk1.3.1_01\bin\code>

```

图 1-15 显示编译后的文件变化情况

如果此程序不含语法错误提示，那么，此命令将创建一个 `Welcome.class` 文件，自动保存在源文件目录下，此文件含有表示该程序的 Java 字节代码。

运行字节码文件程序，键入：`java Welcome`

程序运行结果如图 1-16 所示。

```

命令提示符
2005-02-02 12:29          203 Welcome.java
2005-02-02 17:19          436 Welcome.class
          2 个文件      639 字节
          0 个目录      159,858,688 可用字节

C:\jdk1.3.1_01\bin\code>java Welcome
Welcome to Java Programming!

C:\jdk1.3.1_01\bin\code>

```

图 1-16 运行程序并显示运行结果

此命令启动 Java 解释器，载入“`.class`”文件，字节代码被 Java 解释器解释执行。解释命令省略了 `.class` 文件扩展名；否则解释器不执行。解释器自动调用方法 `main`，然后通过 `System.out.println` 方法显示 “Welcome to Java Programming!”。

如果你在安装时没有另外指定 JDK 安装目录，则 `javac.exe` 和 `java.exe` 被存放在 `c:\jdk1.3.1_01\bin` 目录之下（以 JDK 1.3.1_01 版本为例）。如果想在任何目录下都能使用编译器和解释器，应在 DOS 提示符下运行命令：`C:\>path c:\jdk1.3.1_01` 或将 `path c:\jdk1.3.1_01\bin` 放到 `autoexe.bat` 文件中。

实验 2 创建第一个 Java Applet 程序

Java 程序分为 `Application` 和 `Applet` 两大类。Java `Applet` 作为实现动态的、交互式网页功能的编程工具，在当今网络世界中扮演着重要的角色。`Applet` 是一种嵌入到 HTML 文件当中的 Java 程序，可以通过网络下载来运行。HTML 是网页设计语言，它采用一整套标记来定

义 Web 页。一个 HTML 文件可定义一个 Web 页，文件扩展名为.html 或 .htm。与从命令窗口执行 Java 应用程序不同，Applet 通过 JDK 的查看器 appletviewer 或 WWW 浏览器运行。当一个包含 Applet 的 HTML 文档在 appletviewer 中打开时，该客户程序将执行 Applet。

1. 实验题目：显示一行字符串的简单 Java Applet。在记事本中编写源程序。

```
// 文件名: WelcomeApplet.java
// A first applet in Java

import javax.swing.JApplet; // 加载系统类 JApplet
import java.awt.Graphics;   // 加载系统类 Graphics
```

```
public class WelcomeApplet extends JApplet {
    public void paint( Graphics g )
    {
        g.drawString( "Welcome to Java Programming!", 25, 25 );
    } //结束 paint 方法的定义
} //结束类 WelcomeApplet 的定义
```

2. 语法说明。

“//”表示单行注释。Java 含有许多预定义的类或数据类型，这些类被归入 Java API (Java 应用程序编程接口, Java 类库)的各个包中。程序中使用 import 语句告诉编译器编译此 Applet 所需的系统预定义类的位置。程序中两行加载语句告诉编译器 JApplet 类的位置在 javax.swing 包中, Graphics 类的位置在 java.awt 包中。当在 Java 中创建一个 Applet 时,通常要加载 JApplet 类或 Applet 类;加载 Graphics 类为的是使程序能够画图(如线、矩形、椭圆和字符串)。

注意: java.applet 中有一个传统的 Applet 类,它没有包括在 Java 最新的 GUI (图形用户界面)构件 javax.swing 包中。

Java API 中的所有包存放在 java 目录或 javax 目录中,这两个目录下还有许多子目录,包括 awt 目录和 swing 目录。注意:在磁盘上找不到这些目录,因为他们都存储在一个称为 JAR 的特殊的压缩文件中。在 J2SDK 安装结构中有一个名为 rt.jar 的文件,该文件包括了 Java API 里所有.class 文件。

与应用程序一样,每一个 Java Applet 至少由一个类定义组成。但是,用户几乎不必“从头开始”定义一个类。因为 Java 提供继承机制,使用户可以在已存在的类的基础上创建一个新类,程序中通过关键字 extends 实现。Extends 前面为用户自定义的类,作为派生类或子类,extends 后边的类名为被继承的类,称为基类或父类,本程序中为系统类 JApplet。通过继承建立的新类具有其父类的属性(数据)和行为(方法),同时增加了新功能(尤其是在屏幕上显示 Welcome to Java Programming!的能力)。

实际上,一个 Applet 需要定义 200 多个不同的方法,而在上边的程序中,我们只定义了一个 paint 方法。如果非得定义 200 多个方法,仅仅为了显示一句话,我们可能永远无法完成一个 Applet。使用 extends 继承 JApplet 类,这样 JApplet 的所有方法就已成为 WelcomeApplet 的一部分,使用继承机制,程序员不必知道所继承的基类的每一个细节,只需知道 JApplet 类具有创建一个 Applet 的能力。

学习 Java 语言,一方面是学习用 Java 语言编写自己所需的类和方法,另一方面是学习