

IS471/68

Java 2: In Record Time

Java 2轻松进阶

〔美〕 Steven Holzner 著

王岚波 等译

電子工業出版社

Publishing House of Electronics Industry

前 言

目前，Java越来越受到用户的表睐。这并不是为奇：Java在许多方面已经改变了Web和Web程序设计。Java使得Web页面活龙活现，它可以显示按钮、菜单、图像和动画。因为用户可以使用Java对许多不同类型的计算机进行编程，所以Java应用程序也日益变得十分流行。由于Java已经推出新的2.0版本，它比以前的版本更受用户欢迎。由于越来越多的Web浏览器具有Java解释功能，也由于越来越多的计算机支持Java，Java十分活跃。

为什么使用Java

尽管不用Java也可以创建Web页面，但是这些Web页面无法与Java页面相比较。为什么页面中没有一些动态的东西，而只有静态图像？为什么不允许用户输入他们自己的文本，而只有不能改变的文本？在各种Web页面中，交互Web页面最引人注目。

Java也是作为非Web应用而问世的一种程序设计语言。因为它是一种功能十分强大的跨平台语言，用户可以在一台计算机上开发程序，而在许多种其它计算机上运行。在当今的世界中，这是使用Java的一个重要原因。Java 2具有许多功能强大的选项，从弹出菜单到使用剪贴板，到Java文件，从Java基础类（JFC）到使用Java 2D图形生成系统。本书将探讨所有这些内容。

本书的组织

本书的重点是，通过剖析Java 2的实际程序，向读者介绍高效使用Java的技巧。如果读者想从程序设计的书中学到真正有用的内容，最佳方法是阅读实际运行的代码。许多程序设计的书籍是从抽象概念，程序设计结构和理论开始，但是本书不是这样，本书通过实际例子把Java作为建立读者非常感兴趣的程序的工具。

在本书中，读者将从不同的角度，即从完成任务的角度，观察事物。本书不是由“If语句”、“Java修改符”，或“抽象Java类”这样的各章构成，本书中的各章教授具体的技巧，诸如“复选框”、“单选按钮”、“菜单”和“图形动画”。使读者可以使用Java建立他们喜欢的程序。

读者可通过大量短小精悍的小例子程序学习Java，因为不实际运行Java程序而学习Java，就如同通过阅读飞机部件手册学习飞行一样，这些例子将比较短，但是切记要点，覆盖以下内容：

文本框和文本区

弹出计算器

Java滚动屏类

弹出菜单

JAR文件

单选按钮和复选框

打印

系统剪贴板

滚动条和滚动列表	Java事件模型
雇员数据库	图像图形
菜单	弹出窗口
弹出式对话框	布局
Java Swing (Java基础类)	Java 2D图形系统
窗格	按钮
导航到其它URL	多线程
鼠标驱动绘图程序	鼠标驱动图像调整大小
直接访问键盘	字体
图形动画	Sun公司的动画器 (Animator) 类

本书的目标是，使读者能够在编写Java程序过程中充分使用Java的潜力和丰富的类，以获得在实际中使用Java的经验。Java具有包罗万象的功能。本书向读者介绍所有这些功能，从建立Java小程序到建立Java应用程序，从建立新的对话框，到直接从程序打印。Java还提供许多工具。在本书中，当我们编写程序时，我们能够充分使用这些工具。

本书选配光盘中的内容

本书中的全部程序代码都存放在本书选配光盘中，这意味着读者在使用这些例子时，不必键入这些代码。读者将看到Java文件，这是程序的源代码，读者也将看到.class文件，这是可以运行的文件。要把这些文件安装到硬盘，见本书选配光盘根目录中readme.txt文件。

本书也具有一个相应的Web站点。读者可以通过单击本书选配光盘上的热链接 (hotlink) 访问这些站点，也可以通过Web浏览器访问www.sybex.com。在这个Web站点上，读者将看到提供共享件、演示程序和其它实用程序的公司的链接。这些共享件、演示程序和实用程序可以在Java 2中使用。

本书中使用的约定

本书中的例子程序将按照逐步增加的方式编写。当增加一个新行或若干行代码时，将使用箭头说明新行：

```
drawbutton = new Button("Draw freehand");
linebutton = new Button("Draw a line");
→ ovalbutton = new Button("Draw an oval");
→ rectbutton = new Button("Draw a rectangle");
→ roundedbutton = new Button("Draw a rounded rectangle");
```

一些代码行后面有接续箭头 (➡)。这表示整个代码行在一行中放不下。当读者编写代码时，应该把续接行后面的代码放在续接箭头前面的代码行。例如，以下代码应该写为一行：

```
ptDrawTo = new Point(Math.max(ptAnchor.x, ptDrawTo.x), ➡
Math.max(ptAnchor.y, ptDrawTo.y));
```

对于由代码列表标头标识的程序，读者也能够看到完整的代码表列。这些代码位于CD盘上。

在整个本书中，说明、提示和警告向读者提供一些附加信息。

说明：说明表示一些需要特别注意的要点，诸如在某一特定的情况下，为什么使用鼠标是十分重要的，或在Java中读者能够使用一种Web浏览器完成某些功能，而用另一种浏览器不能实现这些功能。

提示：提示的目的是为了节省读者的时间或给出一些有帮助的信息，诸如程序设计捷径。

警告：警告提示读者一些应该注意的常见程序设计错误或其它潜在的问题。

对读者的要求

尽管Java是面向对象的语言，但是读者不必具有面向对象程序设计的经验。如果读者已经具有面向对象的程序设计语言的经验，读者仍然能够从本书学到很多内容和技巧。

读者需要一个编辑器或字处理器，诸如Microsoft WordPad，这样能够把Java程序输入到可以编译和运行的文件中。编辑器或字处理器必须能够以简单的文本格式保存文件。

当然，要在Web上使用读者已经完成的Java程序，读者应该具有一个地方发布你的web页面。尽管Java能够产生单机独立运行的应用程序，但是，大多数Java程序员在Java程序开发中是面对Web的。在第1章中，我们将学习如何在Web页面中嵌入Java小程序。

Internet服务提供商应该能够告诉读者如何把你的小程序上装，这样可以在Web页面中使用它们。通常，读者需要找出在何处安装Web页面和小程序（在ISP计算机中哪个目录），然后可以使用文件传输协议（FTP）程序将你的文件上装到那个目录。

在Internet上可以发现更多有关Java信息。这些资源包括Sun的Java站点：<http://www.javasoft.com>和<http://www.javasoft.com/doc>。读者也可以检查Usenet组comp.lang.java.programmer。有关对本书的更新信息，请浏览Sybex站点：<http://www.sybex.com>。

目 录

第1章 构造第一个Java程序例子	1
构造Hello例子程序	1
安装Java JDK	3
理解Java	6
面向对象的程序设计	8
连接Java和HTML	14
本章要点	17
第2章 处理Java文本框	18
说明文本框	19
使用init()方法初始化	21
使用new运算符处理内存	22
Java构造器	23
重载Java方法	23
本章要点	26
第3章 使用Java按钮	27
在Java中使用按钮	27
如何处理多按钮	36
处理Java文本区	42
本章要点	47
第4章 使用Java布局工具和复选框	48
Java布局	48
构造加法小程序	48
增加Java标题控件	51
使用Java网格布局	58
构造具有复选框的程序	64
本章要点	73
第5章 单选按钮	74
构造具有单选按钮的程序	74
构造具有面板的程序	81
组合使用复选框和单选按钮	86
本章要点	97

第6章 增加滚动条	98
向程序增加滚动条	98
使用滚动条和边界布局	107
使用ScrollPane类	115
本章要点	118
第7章 使用选择控件和滚动列表	119
使用选择控件	119
使用滚动列表	126
使用网格袋布局管理器	131
使用GridBagConstraints	133
本章要点	141
第8章 建立窗口和菜单	142
处理Java窗口	142
使用菜单	149
构造全功能菜单	165
弹出菜单	176
本章要点	182
第9章 构造Java对话框	183
使用对话框	183
构造弹出式计算器	194
本章要点	205
第10章 Java与图形	206
使用鼠标	206
Dauber小程序	219
从Java中打印	248
Java应用程序	250
本章要点	252
第11章 图像的显示与变形	253
Imagersizer小程序	253
使用图像映象	261
本章要点	266
第12章 Swing和Java 2D	267
Swing软件包	267
使用Swing按钮	267
使用组合框Swing	271
使用滑动杆Swing	276

使用Swing工具条	281
使用Java 2D	287
本章要点	293
第13章 文本和字体	294
建立Scribbler小程序	294
使用键盘	300
使用字体	302
使用FontMetric类	307
使用来自Java的剪贴板	314
本章要点	317
第14章 图形动画	318
建立基本的动画	318
第一个真正的动画	324
使用多线程	325
消除屏幕闪动	333
Animator类	336
本章要点	338
第15章 再论图形动画	339
使用双缓冲	339
更新Dauber小程序	346
卡片布局与动画	360
本章要点	370
第16章 多线程程序和JAR文件	371
建立线程对象	371
使用Runnable接口	375
控制线程和设置优先级	376
终止或重新启动线程	377
处理多线程	382
设置线程同步	389
同步函数	392
同步代码块	394
快速下载: JAR文件	395
本章要点	397
读者购盘说明	398

第1章 构造第一个Java程序例子

- 安装JDK 2.0
- 使用Java小程序浏览器
- 创建Java小程序
- 定制小程序
- 面向对象程序设计
- 连接Java和HTML

欢迎学习和使用Java 2！在你面前呈现了一个宏伟的规划：全面掌握Java程序设计、编写功能强大的Java程序和Web页面。我们将带你全面学习Java 2。Java 2是目前可用的最令人激励程序设计软件包。正如你可能知道的那样，由于越来越多的人已经看到Java是十分通用和强大的程序设计语言，Java流行热浪日益升高。Web程序员已经发现Java是一个十分优秀的工具，因为Java程序可以在许多种不同的计算机上运行。他们开始使用Java制作实用的Web页面。

Java能够显示功能和图像、接受鼠标单击和文本、使用控件（诸如滚动条和复选框）、打印图形、支持弹出菜单、甚至支持附加窗口和菜单条。

本书的各章专门介绍一种Java技巧。在深入学习如何使用Java的过程中，你将阅读大量短小的例子程序。其它介绍Java的书籍可能仅有少量例子，或不完整的例子程序太长而不易理解。本书提供了大量例子。这些例子短小，但完整清晰说明了一个技巧。

我们将立即开始介绍Java。本书的各章不是充满抽象难懂的概念。我们将集中精力阐述例子程序，从程序员的角度介绍Java。

Java程序有两种类型：一种是独立运行的应用程序，另一种是嵌入到Web页面中的小程序（称为applets）。在这两种程序中，小程序是最受欢迎的，因此我们将主要介绍小程序。

构造Hello例子程序

第一个例子是一个简单的程序，因为目前我们刚刚开始介绍Java不想让一些额外的细节使你失去信心。我们将创建一个短小的Java程序。这个程序可以嵌入到Web页面中。它将显示“Hello from Java”！

什么是小程序

小程序是什么呢？小程序（applets）是一种特殊的程序，你可以把它嵌入到Web页面，这样小程序对Web页面的某一部分获得控制权。在页面的这部分，小程序可以显示按钮、列表框、图像等等。小程序使Web页面“充满活力”。

每个小程序都在Web页面中占据一定空间（通常以像素为单位），比如图1.1所示的空

间大小（下面介绍小程序如何“请求”空间）。这个空间是小程序用于进行显示的区域。我们在小程序中放置“Hello from Java”，如图1.2所示。



图1.1 小程序在Web页面中请求空间

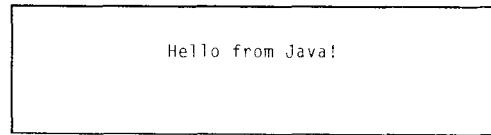


图1.2 Hello from Java!

这是这个小程序的工作原理。在建立这个小程序后，你将能够在Web页面中嵌入它。现在我们建立并且运行这个小程序。

建立Hello例子程序

我们对这个第一个小程序命名为hello。文件hello.java存储实际的Java代码（组成程序的文本行）。读者需要一个某种类型的编辑器建立这个文件（诸如Windows Wordpad或notepad）。在本书中我们将建立Java文件，因此使用你熟悉的编辑器。也要注意，如果你打算使用诸如Microsoft Word这样的字处理器，必须把Java文件存储为纯文本文件（可以在DOS提示符上显示的阅读的文件）。查看字处理器的Save As菜单选项，或字处理器的文档，看一看如何完成这个工作。Sun Java系统只能处理纯文本文件。现在将下列文本输入到文件hello.java中（这是大多数Java书籍传统的第一个程序）：

```
import java.awt.Graphics;
public class hello extends java.applet.Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hello from Java!", 60, 30 );
    }
}
```

这是第一个Java程序的文本。后面你将看到每行的含意。在输入完程序后，将它存储到磁盘上的hello.java文件中。

说明：注意这里大小写是敏感的。一定要键入hello.java，不是Hello.java或hello.Java。

一般说来，文件的名称将准确地与文件中“类”语句匹配（包括大小写）。在本例中，这是hello:

```
import java.awt.Graphics;
→ public class hello extends java.applet.Applet
{
    public void paint( Graphics g )
    {
        g.drawString("Hello from Java!", 60, 30 );
    }
}
```

在本书中，你将把程序放到java1-2目录的各个子目录中（这是可选的，你可以选择任何名字）。这意味着你将把hello.java文件存储为C:\java1-2\hello.java。

现在已经建立了hello.java——这是小程序的源代码。它包含你编写的Java代码。下一步是将这个Java代码编译成为可以运行的小程序，并且看一下运行中的小程序。小程序具有扩展名.class，因此实际的小程序名字为hello.class。后面我将说明为什么小程序具有扩展名.class。

安装Java JDK

现在要使用Java本身，从Java源代码hello.java建立小程序hello.class。如果你尚未完成这个任务，你应该安装Java开发工具（JDK）。它在本书选配光盘上。

在Java的以前版本中，常常要经历冗长和复杂的安装过程，但是这已经一去不复返了。你只需运行.EXE文件。你可以在线从<http://developer.javasoft.com>上获得这个.EXE文件。只要下载它并且按照安装说明即可。

下一步是在计算机上的任何位置运行JDK（包括C:\java1-2目录和基本目录，它是存放Java程序的目录）。为了能够这样做，务必保证AUTOEXEC.BAT文件（位于C:盘的主目录）中的PATH语句包含JDK BIN和LIB目录（我已经在C:\jdk12安装了JDK——使用任何已经在其中安装JDK的路径）：

```
PATH=C:\WINDOWS;C:\JDK12\JAVA\BIN;C:\JDK12\JAVA\LIB JDK
```

提示：如果你需要详细有关安装JDK的帮助，查看<http://www.javasoft.com>站点上的调试排错Web页面。

你可以从JavaSoft把Java文档拷贝到相同目录，例如，C:\JDK12。解压缩.zip文件文档，建立docs子目录（解压缩程序必须能够处理长文件名）。

说明：你需要Web浏览器查看Java文档，因为它是HTML格式。现在已经安装了Java 2，我们看一下这个Java版本中的新特性。

Java 2中的新特性

如果你熟悉Java 1.0或Java 1.1，那么你大概期望在Java 2中有一些改变。下面让我们看一下这个新Java版本中的变化。如果你不熟悉Java，则可先跳过下一节，以后再看这个材料。如果以前没有使用Java编程，这个内容很难理解。

从Java 1.0到Java 1.1

许多读者可能熟悉Java 1.0，但是不熟悉Java 1.1，因此我们首先看一下Java 1.0到Java 1.1的变化。

抽象窗口工具增强 Java 1.1支持打印、快速滚动、弹出菜单、剪贴板、事件模型、图像和图形增强等等。此外，它的执行速度比Java 1.0更快（这是Java程序员最受益的特性）！

.jar文件 .jar（Java文档）文件出现在Java 1.1中。它允许将若干文件作为一个包，然后压缩这些文件，这样用户一次可以下载许多文件。可以将许多小程序和数据放在一

个.jar文件中,这使得下载更快。除了浏览器下载.jar文件,然后随时解压它们之外,这些文件类似于.zip文件。

国际化 Java 1.1允许程序员开发本地小程序,包括使用UNICODE字符、本地机制、它支持对消息本地化、本地日期、时间、时区和数字系统。

带签名的小程序和数字签字 Java 1.1能够建立数字签名的Java应用程序。在出现错误的情况下,数字签字向用户提供返回的“路径”。这是Word Wide Web上最流行的新安全性机制。

远程方法调用 在Java 1.1中,远程方法调用(RMI)允许Java对象从运行在其它Java对话中的Java代码中调用方法。这类似于本地远程过程的调用(LRPC)。

对象串行化 串行化(Serialization)在Java 1.1中是新特性。它允许用户利用二进制输入/输出流存储和处理对象。除了允许存储你串行化的对象拷贝之外,串行化也是RMI中对象通信的基础。对于熟悉Microsoft基础类的读者说,对象串行化类似于MFC串行化。

反射(Reflection) 在Java 1.1中,反射允许Java代码检查有关方法和装载类的构造器,并且利用这些反射的方法和构造器。

内部类(Inner classes) Java 1.1使得建立适配器类更加容易。适配器类是实现API(应用程序接口)所需要的接口的类。适配器类将控制“递交”回给结束的主对象。

新Java方法接口 固有(native)代码是为专为某个特定机器编写的代码。在Java 1.1中,这个接口用于为编写Java固有方法提供标准编程接口。主要目标是在给定平台上跨越全部Java虚拟机固有方法库的二进制兼容性。编写和调用固有代码能够极大地提高执行速度。Java 1.1包含一个功能十分强大的新Java固有方法接口。

Byte, Short和Void类 在Java 1.1中,当使用新的Java类Byte和Short时,可以将字节(byte)和短(short)值作为“环绕式”数字处理。新的Void类是一个位置标识符。我们能够从Void类派生类,而不是直接使用它。

废弃的方法 在Java 1.1中,许多Java 1.0的方法被认为是过时的。在Java 1.1文档中这些方法被标记为被废弃的(当Java编辑器遇到使用被废弃特性的代码时,它显示一个警告信息)。

网络增强 Java 1.1中的网络增强包括在java.net基类中支持有选择的BSD套接字功能。在Java 1.1中,Socket和ServerSocket是非最终可扩展的类。为了更准确地报告和处理网络错误,增加了SocketException的新子类。

I/O增强 在Java 1.1中,字符流扩展了I/O包。字符流除了包含16位Unicode字符而不是8位字节之外,它与字节流相同。字符流使得编写独立于特定字符编码的程序更加容易,因而更易于国际化。几乎字节流的全部可用功能也都对字符流可用。

以上是Java 1.1中新特性的概述。如果你对这些内容一无所知,不要着急。通过学习你将清晰地掌握它们。

从Java 1.1到Java 2

现在我们看一下Java 2中的新特性。

安全性增强 现在当装入代码时,根据目前有效地安全性策略对代码分配权限。每个权限指定对某个资源的访问权限(诸如,对某个文件或目录的“读”和“写”权限,

对某个主机和端口的“连接”权限等等)。外部可配置的策略文件可以初始化策略(它指定从各个签字人/位置,代码的可用权限)。除非明确地对代码分配权限,否则不能访问被权限保护的资源。

Swing (JFC) Swing是Java基础类(JFC)的一部分,它使用“可插入的”视觉组件(Look and feel)实现一组新的GUI部件。Swing用纯Java实现,并且基于JDK 1.1轻量级UI框架。可插入的视觉允许你设计一组GUI部件,这组GUI部件能够自动地具有任何平台(例如,Windows, Solaris, Macintosh)的视觉(look and feel)。

Java 2D (JFC) Java 2D API是为高级2D图形和图像准备的一组类。它在一个完整的模型中包括线形图,文本和图像。

易用性 (JFC) 通过Java易用性API,开发人员能够建立Java应用程序,这些应用程序能够使用有益的技术,比如屏幕阅读器、语音识别系统和盲文终端。

拖放 (JFC) 拖放能够使数据在Java和应用程序之间、在Java应用程序之间和一个Java应用程序内传输。

聚集 (Collections) Java聚集API是为演示和操纵Java聚集而设计的统一框架(后面将详细介绍有关聚集的内容),聚集可以独立于它们代表的细节。

Java扩展 框架 (Framework) 扩展是Java类(和任何相关代码)的软件包。应用程序开发人员可以使用它们扩展核心平台。扩展机制允许Java虚拟机(VM)以它使用系统类的方式使用扩展类。

JavaBean增强 Java 2向开发人员提供了建立复杂JavaBean部件和应用程序所需的标准机制。这些JavaBean部件和应用程序向客户提供了与延时环境的无缝集成,诸如基本操作系统或浏览器。

输入方法框架 输入方法框架使得所有文本编辑部件能够通过标准输入方法接收日文、中文或朝鲜文的文本输入。

软件包版本标识 “版本标识”能够确定软件包的版本,应用程序和小程序能够在运行时标识某个Java延时环境、VM和类软件包的版本。

RMI增强 远程方法调用(RMI)具有若干新增强,包括远程对象激活(它支持远程对象和自动对象激活)和客户套接字类型(它允许远程对象指定RMI对那个对象远程调用使用的客户套接字类型(使用客户套接字类型,支持通过安全传输的RMI(比如SSL))。

串行化增强 目前串行化包括允许指定对象的串行化数据,而与类的域无关。使用现在的技术,这允许串行化数据域写向流(stream)或从流读入(这保证与缺省读和写机制兼容)。

引用对象 引用对象封装对某个其它对象的引用,这样可以像任何其它对象那样检查和操作引用本身。引用对象允许程序保持对一个对象的引用,而不阻止对象被Java“垃圾聚集器”收回,垃圾聚集器(garbage collector)管理内存。

音频增强 音频增强包括新的声音引擎,并且支持小程序和应用程序中的音频。

Java IDL Java IDL向Java增加了COBRA(Common Object Request Broker Architecture)功能,它提供基于标准的互操作性和连接性。Java IDL使分布的具有Web能力的Java应用程序,能够使用工业标准OMG IDL(Object Management Group Interface Definition

Language) 和IIOP (由对象管理小组定义的Internet Inter-ORB Protocol) 透明地在远程网络服务上调用操作。

JAR增强 增强包括用于建立和更新JAR文件的命令行JAR工具所增加的功能。对于读和写JAR文件也有新的标准API。

JNI增强 Java固有接口 (Native Interface) 是编写Java固有方法和将Java虚拟机嵌入到应用程序中的标准程序设计接口。主要目标是在某个平台上跨越所有Java虚拟机固有方法库的二进制兼容。Java 2扩展了Java固有接口 (JNI), 融合了Java平台中的新特性。

JVMDI 新调试器接口, Java虚拟机, 对调试提供低级服务。这些服务的接口是Java虚拟机调试器接口 (JVMDI)。

JDBC增强 Java数据库互联 (JDBC) 是标准SQL数据库访问接口, 它提供对广泛的关系数据库的统一访问。JDBC也提供一个公共基础。较高级工具和接口可以构造在这个公共基础之上Java 2软件, 包括JABC和JDBC-ODBC桥。

随着学习的深入, 这些概念将越来越清晰。现在我们准备编译hello小程序, 看一下它如何运行。

编译Hello小程序

由于我们已经安装了JDK并且建立了hello.java文件, 因此我们可以建立实际的小程序, 看一下它如何运行。要完成这个工作, 现在转到C:\java1-2\hello目录 (或存放hello.java的任何目录)。这是在DOS提示符下应该看到的情况:

```
C:\java1-2\hello>
```

接下去, 键入以下命令, 创建小程序:

```
C:\java1-2\hello>javac hello.java
```

接受Java代码并且把它转换为可以运行的.class文件, Java程序名是javac.exe, 它是Java编译器 (即, 它把.java文件编译为.class文件)。如果键入DOS命令Dir看一下当前目录的内容, 应该看到hello.java和hello.class文件。因为你已经创建了hello.class, 小程序可以运行。但是, 这意味着什么? 实际上做了什么工作?

理解Java

现在我们花费一点时间对Java做一个概述。与大多数程序设计语言相同, 我们使用字和数字编写Java代码, 然后把它们翻译 (编译) 为计算机能够理解的二进制文件。Hello.java程序是一个例子。编写它, 这样就能够理解它。但是, 当你想要实际地运行程序时, 必须把它编译成为计算机能够使用的代码。在本例中, 这意味着使用Java编译器产生hello.class文件。hello.class是字节码的二进制文件。Java兼容的Web浏览器能够运行它, 产生所期望的结果。用这种方式, 若干行Java程序代码可以被编译成为几个字节。当Web浏览器读取嵌入了Java小程序的Web页时, 实际地下载这些字节。也就是说, 实际的小程序是.class文件, 像hello.class。你把这些文件放在Internet服务提供者的服务器上, 这样其他人的Web浏览器能

够下载它们，正如你不久将看到的那样。

提示：后面将介绍如何使用.jar文件（Java文档文件），把.class压缩得更小，将多个.jar文件打包成为一个可下载的文件。

经验丰富的程序员可能不禁要问：为什么Java不把代码编译成为每台计算机实际运行的机器码？因为Java代码是专门作为独立于机器的，这样它们可以在各种机器上运行。这也是为什么它们在Internet上如此受欢迎的原因。无论你把它们下载到何种机器类型，只要用户的Web浏览器能够支持Java即可。下载的代码由Java虚拟机（JVM）运行。JVM负责把代码转换到其中的JVM。它使用JVM的类装入器（class loader）装入组成小程序的.class文件，然后再运行这些小程序。

运行Hello小程序

要看到第一个小程序hello.class运行，你需要一个Web页面，把这个小程序放到其中。使用编辑程序建立一个新文件hello.htm。它是用Web页面语言，超文本标记语言（HTML），编写的Web页面。将下列文本输入到hello.htm中，然后保存它（过一会我们将介绍HTML）：

```
<html>
  <!-- Web page written for the Sun Applet Viewer>

  <head>
    <title>hello</title>
  </head>

  <body>
    <hr>

    <applet
      code=hello.class
      width=200
      height=200>

    </applet>

    <hr>
  </body>
</html>
```

现在通过简单地浏览这个新的Web页面hello.htm，我们能够运行hello小程序。要完成这个任务，使用JDK 1.1提供的Applet Viewer。要使用Applet Viewer，转回到hello子目录，输入以下命令：

```
C:\java1-2\hello>appletviewer hello.htm
```

注意字母的大小写是十分重要的。务必保证大小写与Web页面名字准确匹配。当你完成这个工作后，Applet Viewer运行，如下图所示。你看到消息“Hello from Java!”。第一个小程序是成功的。



提示：可以使用任何具有Java能力的Web浏览器浏览这个Web页面。对于本书中的大多数小程序，必须使用支持Java 2（不是支持Java 1.0或Java 1.1）的Web浏览器，或使用Sun Applet Viewer

第一个小程序hello.class成功地运行，但是准确地说，你做了什么呢？现在我们看一下hello.java文件中的Java代码，逐步地分析，以掌握Java程序设计的工作原理（尽管Java将为你处理许多细节）。

理解Hello例子程序

现在我们开始剖析第一个小程序。首先从这一行开始：

```
import java.awt.Graphics;  
.  
.  
.
```

这意味着什么？这行实际上说明了Java程序设计的一个巨大的优点。当你向Java小程序增加菜单和不同窗口时，可以想象到：如果一切从头开始，需要大量工作。也就是说需要为菜单处理，不同窗口创建等等编写全部代码。Java不要求你这样做，取而代之，Java具有许多完整的预先定义好的库。本书的大多数章节将详细介绍这些库中的例程（routines）。在本书后面，我们将详细学习这些内容，但是现在要做的是，从Java图形库例程向小程序增加支持功能。用这种方式，我们将能够在小程序的窗口中显示字符串“Hello from Java!”。

说明： C/C++程序员会注意到，import语句与C/C++的# include语句非常类同。

接下去，向hello.java增加这一行：

```
import java.awt.Graphics;  
  
→ public class hello extends java.applet.Applet  
  
→ {  
.  
.  
.
```

现在已经建立了一个名字为hello的Java类。这意味着什么呢？

面向对象的程序设计

在面向对象的语言中（诸如Java），对象（objects）和类（classes）是两个基本的概念。人们对面向对象的程序设计（OOP）已经做了大肆宣传。这使得这个内容似乎很神秘并且高

不可攀。事实上，面向对象程序设计的出现，使得较长的程序更容易编制。首先通过讨论对象，我们简要介绍一下面向对象的程序设计。

理解Java对象

在冗长、复杂的程序中，可能有大量的变量和函数，有时甚至有几百个变量和函数。建立和维护程序代码是一项极其困难的任务，因为你必须在大脑中记住如此之多的变量和函数。如果各种函数使用相同名字的变量，也可能发生一些非意料之中的交互。利用面向对象程序设计的方法，可以将这样程序分解为若干个小程序。

对象背后的想法十分简单——把程序分解为各个部分，你可以容易地将这些部分的每一个想象为完成一个具体的任务——这些就是对象。例如，你可以把一个程序的全部屏幕处理部分放在名字为screen的对象中。与简单地函数或一组变量相比，对象是十分强大的，因为对象可以按照一种易于使用的方式，把函数和变量组合在一起。对象screen不仅拥有在屏幕上显示的全部数据，它也拥有处理这些数据所需的函数，比如drawString()或drawLine()。这意味着全部屏幕处理工作都以一种方便的方式隐藏于程序的其它部分，从而使得程序的其它部分更易于处理。

作为另一个例子，想象一台电冰箱。如果你必须总是用手调节温度和压缩机的话，电冰箱可能用处不大。将所有这些功能做在内部并且使之自动化，这使它成为易于使用和十分有用的对象：电冰箱。按照这种方式，将代码和数据封装在对象中；这是面向对象程序设计的基础。

Java类

但是如何建立对象呢？这是产生类（class）的根源。类对于对象来说，如同饼干模具对于饼干一样，类是模板（template）。用程序设计的术语讲，你可以想像在数据类型（诸如一个整数）和实际变量本身之间的关系。以下代码行建立了名字为the_data整数：

```
int the_data;
```

在Java中，这是建立整数变量的方法。在这里，int是变量类型，而the_data是int类型的变量。类与对象之间关系和这种关系完全相同。非严格地讲，可以把类认为是对象的类型。

提示：Java支持C和C++中全部标准的原始数据类型，比如int、double、long、float等等。

例如，如果已经建立了类，比如说graphicsClass，可以使用以下方式，建立名为screen的这个类的对象：

```
graphicsClass screen;
```

不久，你将看到如何实际地建立类（建立像graphicsClass这样的类并不难——当在代码中建立类时，要在类定义中分别定义函数和数据）。然后，看到如何建立这个类的对象。重要的是要记住：对象本身存放要使用的数据，类本身不保存数据，但它只描述对象的组成。

面向对象程序设计的精髓是，它将函数使用的数据放在一起，这样使程序变得更清晰。在整个本书中，我们将详细介绍面向对象的程序设计，包括如何建立类，如何建立那个类的对象和如何使用对象中的函数以及数据。

现在我们完成了对类和对象的概要介绍。正如你看到的那样，类是一种程序设计结构，它把函数和数据组合（封装）在一起。对象可以想象为是那个类的类型的一个变量，就如同对象screen与类screenclass的关系一样。

Java提供了若干完整的预先定义类库。这些类库可以节省我们大量的时间。在本书中，我们将介绍这些预先定义的、十分有用的Java类。使用这些预先定义的类，我们能够建立很多对象，诸如按钮、文本域、滚动条等等。

Java包

在Java中，这些类库集合称为包（packages），包中单个的库为java.awt（awtAbstract window Toolkit（抽象窗口工具））。这个库具有Graphics类，这个类处理图形工作。因此在第一个小程序中有这一行：

```
import java.awt.Graphics;
```

这一行实际上意味着，你想要包括Java Graphics类并且在程序中使用它。过一会，我们将为图形输出使用Graphics类的对象。

通过包括java.awt.Graphics类，增加了图形处理的能力（在Java中，显示文本字符串“Hello from Java!”认为是图形处理）。接下去，应该编写hello小程序了。要完成这个工作，定义一个名字为hello的新类。在Java中，这是建立小程序的标准方式。事实上，小程序本身具有文件扩展名.class。这是因为java文件中定义的每个类最终都被转换到.class文件。我们将详细介绍这个主题。

要从零开始编写小程序类需要全部代码是十分困难的。例如，我们需要与Web浏览器交互、保留屏幕的一部分，初始化正确的Java软件包等等。实际上，所有这些功能已经构造在Java Applet类中，它是java.applet包的一部分。但是，如何使用Applet类呢？我们可以生成小程序，让它显示文本字符串。但是java.applet.Applet类本身不知道你要做什么。

Java继承性

通过从java.applet.Applet类派生出hello类，便可自定义java.applet.Applet类。这使得java.applet.Applet成为hello类的基类，使得hello成为从java.applet.Applet类派生出来的类。这显示了java.applet.Applet类的巨大威力，你不必担心自己编写它。通过向派生类hello增加代码，你可以向这个类中随心所欲地增加任何功能。

这是面向对象程序设计的一个重要方面，它被称为继承性。派生类继承其基类的功能并且可以在其上增加更多功能。例如，你可能具有一个称为chassis的基类。从这个基类，你可以派生出各种类，比如说car类和truck类，两个派生类能够共享相同的基类，节省大量编程时间和精力。尽管car和truck类共享相同的基类chassis，但它们向基类增加了不同项（items），最终作为两个不同的类car和truck。

因此，使用继承性，通过建立自己的类hello并且加到基类上，你可以扩展基类java.applet.Applet。使用以下方法，说明hello类是从java.applet.Applet类派生出来的（注意使用关键字class，说明定义新类）：

```
import java.awt.Graphics;
```