



深入学习:

JFC Swing—Java 基础类组件集

[美] Satyaraj Pantham 著

黄晓鸣 武清 译

- 内容涵盖Java 2/JDK 1.2
- 权威的内容介绍
- 商业水准的程序代码
- 适用于中高层编程人员

Pure JFC Swing

SAMS



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

内 容 提 要

Java 语言是当前最流行的编程语言。国际上,在选择编程语言时,无论是编程者还是决策者都会首选 Java——因为其程序可商品化,可以被更多人、更多系统所用。目前,介绍 Java 编程的书很多,但深层次的书较少,而编程者,特别是有经验的编程者恰恰希望阅读或购买后一类书籍。为此,我们翻译了本书。本书全面、深入地介绍了 Java 基础类中 Swing 的使用方法、技巧和编程注意事项。在介绍过程中,每节讨论一个主题,举出程序代码,并随后说明编程思路和技巧。最后的附录汇总了 Swing 的包、类、接口和构造函数,各个类和接口中的子类、方法、合法参数等等。可以说本书是 Java 编程方面的高层专业书。

Authorized translation from the English language edition published by Sams Programming, an imprint of Macmillan Computer Publishing U. S. A.

本书中文简体版专有翻译出版版权由美国 MCP 公司的子公司 Sams Programming 授予电子工业出版社。其原文版权及中文翻译出版版权受法律保护。未经许可,不得以任何形式或手段复制或抄袭本书内容。

Copyright ©1999 Sams. All rights received. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Sams.

图书在版编目(CIP)数据

深入学习: JFC Swing——Java 基础类组件集/(美)潘汉姆(Pantham S.)著;黄晓鸣等译。

-北京:电子工业出版社,2000.1

书名原文: Pure JFC Swing

ISBN 7-5053-5691-7

I. 深… II. ①潘… ②黄… III. Java 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999) 第 70662 号

书 名: 深入学习: JFC Swing——Java 基础类组件集

原 书 名: Pure JFC Swing

著 者: [美] Satyaraj Pantham

译 者: 黄晓鸣 武清

责任编辑: 杨福平

排版制作: 电子工业出版社计算机排版室监制

印 刷 者: 北京朝阳隆华印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 39.75 字数: 976 千字

版 次: 2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

书 号: ISBN 7-5053-5691-7
TP·2928

印 数: 4000 册 定价: 62.00 元

版权贸易合同登记号 图字:01-1999-3360

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;
若书店售缺,请与本社发行部联系调换。电话: 68279077

谨以此书献给

加尼莎和斯坎达

他们是 True Knowledge 的领袖，也是我永远的兄弟。

致 谢

尽管我愿意接受大家对本书的全部赞誉，但我也必须承认，本书是在重述那些创始人的工作。因此，我要首先非常地感谢那些创立了 Swing 概念和 Swing API 的创始人。

我要感谢我在 Sun Microsystem 公司的同事们，特别是卡珀诺·卡特，他在我准备程序的时候花费了很多时间与我讨论 Swing 的各种概念。还有纳瑞希·阿多尼，他帮我检查了本书的部分章节。还要感谢我的朋友戴尔·格林在关键的时候所给予我的鼓励。

我特别要感谢 Sams 小组成员对编写本书所表现出的浓厚兴趣。非常感谢蒂姆·瑞恩，他对本书也有着极大的兴趣，并对我完成这样一种类型的书提出了许多建议、修改和完善意见。感谢约翰·斯迪沃对我程序开发的帮助，感谢帕特·金永为本书编辑所做的出色工作，还要感谢杰夫·伯金斯对书中示例的检查。

最后，我必须感谢我的家人，普希帕和萨拉文，她们在我写书过程中付出了牺牲。我的妻子普希帕表现出了极大的兴趣，并且参与了本书参考部分的大量 API 方法的编辑工作。在时间紧迫，同时又还要编写附录的情况下，单凭我一个人是不可能完成那些工作的。

告诉我们你的想法——原文编辑的话

作为本书的读者，你是我们最重要的评论员。我们尊重你的意见，并想知道我们所做的事情哪些是正确的，哪些可以做得更好，你希望我们在哪些领域出版书籍。我们也想在今后的工作中听到你充满智慧的话语。

作为 Sams 公司 Java 小组的作稿编辑，我真诚地欢迎读者的意见。你可以发传真、电子邮件或直接写信给我，告诉我们你就此书所做过或打算做的工作——也希望你告诉我们怎样才能使本书更好。

你写信时，请在信中包括书名、作者，以及你的姓名、电话号码或传真号。我会仔细拜读你的意见，并与本书的作者和编辑们共同分知。

Fax: 317-817-7070

Email: java@mcp.com

Mail: Tim Ryan

Executive Editor, Java, Sams Publishing

201 West 103rd Street, Indianapolis, IN 46290 USA

引 言

这本带有大量代码的参考书是专门为那些需要尽快学习 Java 2 / JDK 1.2 中 JFC Swing 的概念和实际用法，并且具有编程经验的程序员编写的。由于 Swing 组件的实现依赖于诸如布局管理器和事件处理这样的 AWT 资源，因此，为完整起见，这些内容也都包含在了本书中。无论是否熟悉 AWT，都可以从本书中获得最大的价值。

《深入学习：JFC Swing》分为 3 大部分。第一部分是“概念参考”，这是一个快速导引，其中解释了 Swing 组件集的特性和结构；Swing 与 JFC、AWT、Java 2/JDK 1.2 的关系；以及在实现 Swing 组件时所需要的一些基本编程技巧。

第二部分是“技术参考”，其中的每一章说明一个 Swing 组件集中的组件，并有详细注释、具有商业质量的代码。这些代码展示了如何建立复杂的 Swing GUI。写这本书时，我特别注意了代码的创建，以便读者可以将其直接用于自己的程序之中。在每一个程序中，还用注释片段和说明语句描述了代码模块的功能。

第三部分是“语法参考”。其中包含了四个速查附录。附录 A，“JFC Swing 快速参考”，是关于 Swing 包及其类的快速参考。附录 B，“AWT 事件类快速参考”，包含了 AWT 事件类的参考资料。这些类通常用于处理由 Swing 组件所产生的事件。附录 C，“Swing 中重要的 AWT 类”，其中给出了另外一些重要的 AWT 类，例如组件、容器以及用于 Swing 程序设计的布局管理等。附录 D，“辅助选项快速参考”，这是关于 API 的参考资料。

第一次接触 Swing 而且没有 AWT 使用经验的程序员应特别留意关于事件处理和布局管理的有关章节。这两章中不仅论及了 Swing 的重要内容，而且还解释了 GUI 程序设计的基本技巧以及 Swing 中用到的大多数 AWT 的相关信息。

最后，我想指出，本书所提供的内容和程序是根据专业开发人员的需要而精心选择的。每个程序都包含了实用技巧和能够直接用于日常程序中的代码。然而，程序设计本身是一门艺术，而且开发人员还可以利用 Swing 进行各种各样的发挥。因此，我不能保证读者能够从本书中找到所有 Swing 组件的用法。

S·潘汉姆
圣荷塞，加利福尼亚

第一部分 概念参考

第 1 章 Swing 概要

第 2 章 Swing 小程序

第 3 章 事件处理

原书空白页

第 1 章 Swing 概要

JDK 1.2(Java 开发工具包,后正式命名为 Java 2)的正式版本是 1998 年 12 月发布的。JDK 1.2 包括了一个复杂的图形 API 和用户接口 API 的集合,该集合的核心软件包称为 Java 基础类(JFC)。JFC 中一个重要的技术就是 Swing,它组合了大量的可用于构建复杂用户界面的轻量级组件。图 1-1 表示出了 JFC 及 Swing 与 JDK 1.2 之间的关系。

Swing 组件是现有的抽象窗口工具包(AWT)组件的增强,但不必用它们完全取代 AWT 组件。这两种组件可以用在同一个界面上。Swing 组件包括从按钮、复选框、滚动条和滑动条这样的简单控件到诸如文本窗格和编辑器窗格那样复杂的工具。

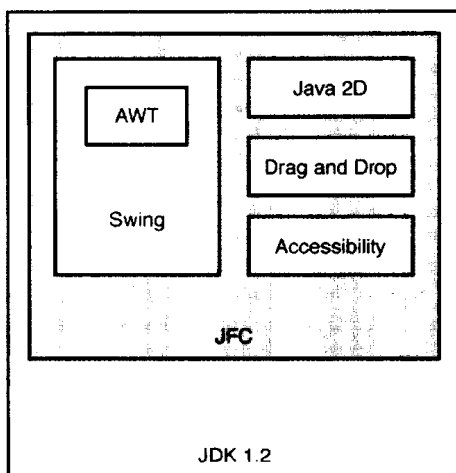


图 1-1 JDK 1.2, JFC 和 Swing

说明:

Swing 最初是作为 JDK 1.1 的扩展 API 出现的,它必须单独下载后才能使用。而在 JDK 1.2 中,JFC 和 Swing 已经被直接集成在了 JDK 中,因此就不需要另外得到 Swing API 了。

1.1 JFC 技术

JFC 包含五种主要技术,其中之一就是 Swing。JFC 也包括了抽象窗口工具包(AWT),辅助选项、拖-放以及 2D 图形处理。本书着重介绍 Swing 和辅助选项 API。以下是 JFC 的简要说明:

- 辅助选项 API——辅助技术(或系统)对于那些需要额外的帮助来使用用户界面的残疾人来说是很有用的。辅助系统包括屏幕阅读器、屏幕放大器以及语音识别系统。辅助选项 API 提供了一个接口,它使得辅助技术能够与 JFC 和 AWT 组件进行交互

和通信。本章将要进一步讨论辅助技术,并提供附录 D“辅助选项快速参考”的介绍。在 Swing 中还要考察相关的 API。

- 抽象窗口工具包(AWT)——抽象窗口工具包,亦即AWT,对Java程序员来说并不陌生。对于JFC本身来说,AWT是它的基石,也是随JDK 1.0发行的核心库之一。事实上,AWT奠定了JFC组件的基础。尽管可以用Swing组件创建用户界面,但仍需要用到AWT中所支持的布局管理和事件处理模块。正因为如此,本书包含了一些AWT中的布局管理和事件处理的内容,以使读者不必另外去查找有关的信息。本书最后也提供了相关API的快速参考。如果需要更多的信息,请参看Sams公司出版的《JFC unleashed》一书。
- 2D图形/Java 2D API——在Java程序设计中,AWT中的API在一定程度上支持图像处理。但随着Java技术逐步成熟,人们认识到,有必要将更为复杂的图像处理API引入到Java之中。因此,JDK 1.2就包含了2D图像处理,以增强现有图像处理能力。2D图像API能够支持先进的2D图形和图像。
- 拖-放——借助于可在本地平台进行拖-放操作的应用程序,一个与Java应用程序相邻的本地用户应用程序就能够在Java应用程序和本地应用程序之间进行拖-放操作。用户动作(拖动动作)的精确程度与其在本地平台上的动作是一致的。

1.2 Swing 的特点

Swing的开发源于Model-View-Controller(MVC)体系结构。基于MVC的体系结构允许Swing组件被不同的数据模型和视图所替代。可插入的外观效果(pluggable look-and-feel)就是MVC体系结构应用的结果。由于Java是不依赖于平台的语言,能够在任何客户机上运行,因此可以针对任何平台来调整软件的外观效果。下面是Swing主要特性的总结:

- 轻量级组件——从JDK1.1开始,AWT就支持轻量级组件的开发。要使一个组件成为轻量级组件,那么该组件就不能依赖任何本机系统类,这种类也叫做“对等类”(“peer” classes)。在Swing中,大多数组件都有其自己的由Java外观类所支持的视图。因此,组件的视图不依赖于任何“对等类”。
- 可插入外观组件——这一特性使得用户不必重新启动应用程序就能够看到Swing组件的外观效果。Swing库支持跨平台的外观——也称为Java外观——它使得应用程序不论在哪里运行都具有同样的外观效果。通常,本机外观效果是针对于程序所运行的特定系统的,Windows和Motif就是如此。Swing库提供了一个API,它对于确定应用程序的外观效果提供了极大的灵活性。

1.3 Swing 包

Swing API被组织成了若干个包,以支持诸如组件、可插入外观组件、事件、对象边界等不同类别的API。附录A“JFC Swing快速参考”中给出了包的成员(类和接口)及其API方法。下面列出了Swing中各个包的名称:

```
javax.swing
```



```
javax.swing.border  
javax.swing.colorchooser  
javax.swing.event  
javax.swing.filechooser  
javax.swing.plaf  
javax.swing.plaf.basic  
javax.swing.plaf.metal  
javax.swing.plaf.multi  
javax.swing.table  
javax.swing.text  
javax.swing.text.html  
javax.swing.tree  
javax.swing.undo
```

另外,还有两个包中存放了 Motif 和 Windows 操作系统的外观类。这两个类包含在 JDK 1.2 中,但由于它们所处理的操作系统属于 Sun Microsystem 公司之外的产品,因此这两个包采用了不同的名称:

```
com.sun.swing.plaf.motif.*  
com.sun.swing.plaf.windows.*
```

1.4 JComponent 类

JComponent 类引入时是作为 Swing 组件的父类。这是一个抽象类,它封装了特定于 Swing 组件的基本特性和操作。AWT 的 Container 类是 JComponent 的父类,如图 1-2 所示。因此,AWT 是建立 Swing 组件集的基础。

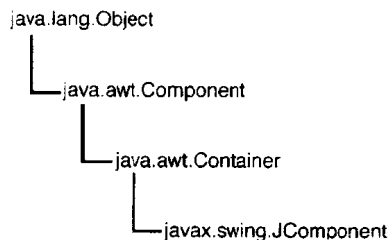


图 1-2 JComponent 的类层次结构图

说明:

由于 Swing 组件是从 AWT 容器类扩展得到的,因此,可以期望 Swing 组件具有容器的特性。但是也必须根据具体的组件而小心地使用这个特性。

JComponent 类是 Swing 组件各种特性通常的存放位置。可以引用附录 A 中列出的代表这些特性的各个方法。下面所列出的就是这里所述的各种特性:

- 辅助选项支持
- 设定组件边界

- 设定组件界限
- 从一个点控制多个组件的动作
- 击键处理
- 附加更改监听程序
- 激活性质更改事件
- 将组件在其显示区域内对齐
- 自动滚动
- 设置双缓冲区
- 为改变大小的动作设置标注属性
- 赋予工具提示
- 允许操作需要的焦点

1.4.1 Swing 组件

本节给出 Swing 组件的概览。有关 Swing 标签和按钮的内容将通过详细的代码进行讨论,以便演示前面所介绍的诸如事件处理、容器、布局以及边界等概念。表 1-1 列出了 Swing 组件及其定义。

表 1-1 Swing 组件

组件	定义
JApplet	这个类代表 Swing 小程序,它从层次上扩展了 AWT 小程序
JButton	这个类代表 Swing 按钮。这些按钮可以在适当的位置(绝对的和相对的)带有图标和文本。可以用它所支持的任何一个构造函数创建按钮对象(请参看第 8 章,“按钮和复选框”)。做法如下: <pre>JButton buttonRef = new JButton(text, icon);</pre> 这里 text 和 icon 是用在按钮上的文本标签和图标
JCheckBox	这个类代表 Swing 的复选框(请参看第 8 章)
JColorChooser	这个类代表 Swing 的颜色选择框。颜色选择框是现成的对话框,其上有一个调色板。这一部分内容将在第 15 章“文件选择框和颜色选择框”中介绍
JComboBox	Swing 中的组合框是对 AWT 组件 Choice 的一个增强。组合框是一个下拉列表,它能够在显示区域内显示出多个选项。也可以在显示区域里输入一项并将它编入列表中(请参看第 9 章,“列表框和组合框”)
JDesktopPane	这是一种附在 Swing 的内部框架上的容器。在第 4 章“框架、窗格和面板”中可以找到有关该组件更多的内容(也可以在本表中找到 JInternalFrame 的定义)
JDialog	这个类代表了 Swing 版本的对话框。JDialog 扩展了 AWT 的对话框,它要求其组件添加到它的内容窗格上(请参看第 14 章,“对话框和选项窗格”)
JFileChooser	JFileChooser 类是一个对话框,它允许打开或保存信息内容。Swing 的文件选择框将在第 15 章中讨论

表 1-1 Swing 组件

(续表)

组件	定义
JFrame	JFrame 代表 Swing 的框架类,它比 AWT 的框架类更为复杂。你可以分层向其中添加组件、菜单条,或者[通过叫做玻璃窗格(glass pane)的组件]在上面绘制组件。有关框架类的内容将在第 15 章中讨论
JInternalFrame	内部框架是附在应用程序主框架上的。在应用程序中,利用内部框架可以启动一次会话(session)过程。内部框架类的内容将在第 4 章中讨论
JLabel	这个类代表 Swing 中的标签,它能够在某个适当位置显示文本串或图标。标签对象可以用下面的语句来创建: <pre>JLabel labelRef = new JLabel(text,icon,JLabel.CENTER);</pre> 这一条语句使用指定的 text,icon 以及按照水平方向对齐的要求创建了一个标签(请参看第 7 章,“图标、标签和边界”)
JLayeredPane	这个类代表 Swing 中的分层窗格。分层窗格建立了一个沿组件深度方向的维(dimension),可以将组件放在分层窗格的每一层上。Swing 小程序和框架都缺省地包含了分层窗格。分层窗格的内容将在第 4 章中讨论
JList	JList 类代表能够在用户界面中显示一系列条目的组件。该组件用到了 M-UI(改进的 MVC)体系结构并允许针对复杂程序设计中的列表模型进行处理(参看第 9 章)
JMenu	该类代表 Swing 的菜单(参看第 13 章,“菜单、工具条和工具提示”)
JMenuBar	该类代表 Swing 小程序或者框架上的菜单条,第 13 章详细讨论了有关的内容
JMenuItem	该类代表 Swing 的菜单项。菜单项是类似于按钮的对象,它能够被程序的特定动作所触发。Swing 的菜单项将在第 13 章中讨论
JOptionPane	该类代表 Swing 中的选项窗格。选项窗格是一些对话框,这些对话框为用户显示了一些信息,或者从用户那里获取信息,以便程序根据用户所做的选择执行进一步的动作。选项窗格在第 14 章中进行讨论
JPanel	该类代表 Swing 的面板,它除了能够组合其他的组件外,还具有双缓冲(double buffering)的功能。窗格类将在第 4 章中进行讨论
JPasswordField	这个类代表一个文本字段,它不显示输入的文本,而是在输入过程中显示特殊的字符(比如 *)。口令字段的讨论放在第 11 章中进行
JPopupMenu	这个类代表 Swing 的弹出式菜单,它包含了接受鼠标单击的功能(参看第 13 章)
JProgressBar	该类代表 Swing 的进度条类,可以用进度条表示某些操作的状态,第 17 章“定时器和进度指示器”中将讨论有关进度条的内容
JRadioButton	该类代表 Swing 的单选按钮。利用单选按钮可以从给定的一系列互斥选项中选择。单选按钮的讨论放在第 8 章中进行
JRadioButtonMenuItem	单选按钮菜单项为表示互斥的菜单项,有关的讨论见第 13 章
JScrollBar	滚动条用于水平或垂直滚动一个容器内的子组件。滚动条见第 12 章“滚动条和滑动条”中的讨论

表 1-1 Swing 组件

(续表)

组件	定义
JScrollPane	Swing 的滚动窗格是一个具有滚动功能的容器,滚动窗格的讨论见第 12 章
JSlider	该类创建一个可滑动对象,这个滑动对象允许从一个指定区间内选取特定的值。关于 Swing 的滑动条的讨论见第 12 章
JSplitPane	该类代表 Swing 的拆分窗格类。拆分窗格可以将位于其中的两个组件分割开来,有关的讨论见第 4 章
JTabbedPane	该类代表 Swing 所使用的卡片式窗格类。这个窗格类能够显示子组件,并且可以节省空间。子组件可以在多个卡片中一次一个地进行显示。卡片式窗格类将在第 4 章中讨论
JTable	JTable 类代表 Swing 中的表格,第 10 章“表格”中将讨论有关内容
JTextArea	该类代表 Swing 的文本区组件。文本区组件允许编辑多纯文本信息。可以在第 11 章中找到有关内容
JTextPane	该类可用来在 Swing 应用程序中创建复杂的编辑器。文本窗格既可以作为图像编辑器,也可作为文本编辑器。文本窗格组件将在第 11 章中详细讨论
JToggleButton	该类代表 Swing 的切换按钮。初次操作切换按钮,它会保持被按下的状态;再次操作后它便保持弹起状态。切换按钮将在第 11 章讨论
JToolBar	该类代表 Swing 的工具条对象。工具条是一个条形区域,可以为其添加一系列重要的按钮或者其他组件,以便更容易操作它们。工具条可以被做成浮动的。第 13 章中详细讨论了 Swing 的工具条
JToolTip	该类代表 Swing 的工具提示对象。“工具提示”可以显示一小段文字,用来指示一个组件的名称或说明该组件的目的。工具提示的内容将在第 13 章中讨论
JTree	该类用来显示一个用户界面中分层存储的数据,它代表了 Swing 的树对象。有关的内容将在第 16 章“树”中进行讨论。

1.5 辅助选项支持

辅助技术(也称为辅助选项技术)帮助那些身有残疾的用户更容易地使用计算机。辅助技术的例子包括为视力受损的用户所提供的屏幕放大功能,为盲人提供的能够将屏幕内容同语音进行合成的屏幕阅读器,以及对为那些不能正常使用键盘的人所提供的基于屏幕的键盘装置,等等。

Java 辅助技术支持提供了一个紧凑、简洁并且易于使用的可扩展 API 集合。利用辅助选项的界面和类,可以把 Java 小程序或 Java 应用程序中的用户界面组件精简成为可访问的或辅助技术(系统),这一技术提供了使用小程序和应用程序的途径。附录 D“辅助选项快速参考”中可以找到关于辅助选项 API 的快速参考。实现辅助选项支持的应用程序结构如图 1-3 所示。

除了辅助选项 API 之外,辅助技术还需必要的辅助选项工具。这些工具可以寻找实现辅助选项 API 的对象,支持将它们装入 Java 虚拟机,监视 AWT 事件和 JFC 事件,计数已打开的顶层 Java 窗口,获取鼠标的光标位置以及获取焦点的组件等等。

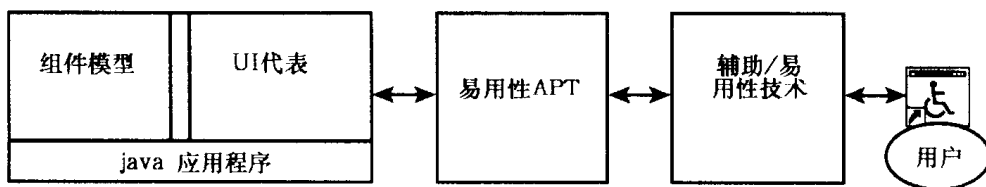


图 1-3 使用易用性支持的应用程序结构

说明:

作为应用程序开发者,通常不会用到辅助选项工具;这些工具是给辅助技术系统的开发人员所使用的。

第 2 章 Swing 小程序

这一章是有关使用 Swing 小程序的快速介绍。其中讨论了创建 Swing 小程序过程中所需的基本内容,以及 Swing 小程序是如何区别于标准小程序的。首先,会看到一个 Swing 小程序,“Hello! Swing World”;然后再研究这个 Swing 小程序的技术细节。

2.1 Hello! Swing World

清单 2.1 包含了一个简单的 Swing 程序 THello.java。这个程序只是简单地输出问候语“Hello! Swing World”(不是以前的问候语“Hello! World”)。“Hello! Swing World”按小程序和应用程序两种方式编写了相应的函数,所以你会在程序中发现一些额外的代码语句,事实上,该程序只有一条核心语句,具体如下:

```
getContentPane().add(new JLabel(  
    "Hello! Swing World",  
    new ImageIcon("AnimSpeaker.gif"),  
    JLabel.CENTER));
```

读过后几章以后,这个语句的含义就会逐渐清晰的。现在可以看见动画图标和问候标语“Hello! Swing World”显示在小程序或者应用程序框架中。另外,你可能已经注意到了分别代表 Swing 小程序和 Swing 框架的类标识符 JApplet 和 JFrame。这两个类分别从 AWT 类 Applet 和 Frame 继承而来。图 2-1 和图 2-2 显示了该程序分别作为小程序和应用程序的运行结果。注意到小程序的运行结果显示的标题栏指明了该程序是在小程序查看器中运行的。还可以看到状态栏中的状态信息“Applet Started”。

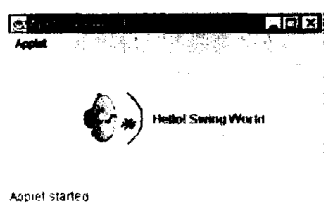


图 2-1 在小程序查看器中运行的 THello 小程序

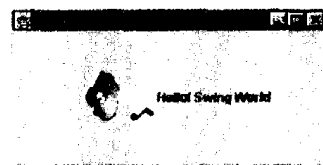


图 2-2 THello 作为一个应用程序运行

清单 2.1 演示 JApplet 的一个简单的 Swing 小程序/应用程序(THello.java)

```
(THello.java)  
// Your first Swing program.  
  
/*  
 * <Applet code = THello width = 300 height = 150 >  
 * </Applet >
```

```
*/
import javax.swing.*; // Import the Swing classes
import java.awt.event.*; // for the WindowAdapter and WindowEvent classes

// This is your Swing applet (by extending JApplet).
public class THello extends JApplet {
    public void init() {
        // Here is the only program statement.
        // Create a string label to display the text as shown here
        // with an icon made from the animated GIF file.
        // The text and icon are centered by using JLabel.CENTER
        // The label is added to the content pane
        // of the applet (see Code Details).
        getContentPane().add(new JLabel(
            "Hello! Swing World",
            new ImageIcon("AnimSpeaker.gif"),
            JLabel.CENTER));
    }

    // When you run this program as an application.
    // The main method to instantiates the application frame.
    public static void main(String[] args) {
        // Create a Swing frame.
        JFrame f = new JFrame("THello");

        // Create an instance of the THello applet.
        THello helloApplet = new THello();

        // Initialize the applet instance.
        helloApplet.init();

        // Add the applet to the JFrame (to it's content pane).
        f.getContentPane().add(helloApplet);

        // Add the listener to close the frame.
        f.addWindowListener(new WindowEventHandler());

        // Assign the frame size and display it.
        f.setSize(300, 150); // frame: width = 300, height = 150
        f.show(); // Display the frame
    }
}

// Class to close the frame and exit the application.
class WindowEventHandler extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
}
```

代码详析

如前所述,该程序既可以用做小程序,也可以作为应用程序。这样,就可以在同一程序中同时使用 `init()` 和 `main()` 这两种方法。其中 `init()` 方法包含创建一个文本标签“Hello! Swing World”和一个动画图标(正发送声波的扬声器)的语句。然后该标签被加入到小程序中。`init()` 方法在此程序中作为小程序方式运行时被调用。

在分析 `init()` 方法内的代码之前,应该注意到程序中出现的两个更重要的特征。如你所见,输入语句包含了 Swing 包 `javjx.swing.*`。这是包含 Swing 类 `JApplet` 和 `JFrame` 的关键程序包。在整个 Swing 编程过程中,编程者会不断地在所有程序的顶端使用这个输入语句。

另一个有趣的地方是,类 `THello` 扩展了 `JApplet`。这是因为该程序是一个 Swing 小程序,所以它扩展了 Swing 小程序类 `JApplet`。在程序顶端的小程序标志 `<Applet>...</Applet>` 允许使用小程序查看器来运行程序 `THello.java`。这就避免了在测试程序时创建另一个名为 `THello.html` 的文件。注意,本书中的所有小程序都是按此惯例来测试的。

接下来,我们考察一下程序内部的 `main()` 方法,该方法一般用在应用程序的代码中。为了创建一个 Swing 应用程序,需要使用一个 `JFrame` 对象来包含这条语句。然而,在这个程序中,含有“Hello! Swing World”标题的小程序被加到了框架中。请看下面的代码语句:

```
// Create a Swing frame.
JFrame f = new JFrame("THello");

// Create an instance of THello applet.
THello helloApplet = new THello();

// Initialize the applet instance.
helloApplet.init();

// Add the applet to the JFrame (to its content pane).
f.getContentPane().add(helloApplet);
```

其余的代码语句显示了具有指定大小的框架。该框架也具有窗口监听程序的特性,亦即在关闭框架时使得程序完全退出。最后一点,像 `lable` 这样的组件被加到所谓的“内容窗格”里,而不是直接加到小程序或框架里。对于内容窗格的引用可以使用 `getContentPane()` 方法获得。在通读本书后就能学到更多有关内容窗格的知识了。

技巧:

可以将本程序(允许程序作为小程序或应用程序运行)中的外包装(`Wrapper`)代码用于其他需要类似功能的程序之中。

2.2 Swing 小程序集

Swing 集引入了它自己的小程序版本,其前身是 AWT 小程序。Swing 小程序的设计更加复杂,它能支持菜单条和在深度维数上的组件分层,并且允许在已经位于小程序内部的组件上

描绘。

由于 Swing 小程序支持菜单条,所以它可以拥有一个菜单条。组件的分层法本质上就是将组件重叠放置在多个层面上。这样就可指定组件放在某一层上,从而使其具有纵深感。Swing 小程序包含一个叫做分层窗格(layered pane)的专用组件来达到这一目的。

允许在已经位于小程序内部的组件上绘制的另一特性可以通过使用一个叫做玻璃窗格(glass pane)的组件来完成。玻璃窗格是一个透明组件,它使得背景组件是可见的。编程者可以在玻璃窗格上绘制文字或图像,并使得前景上的图像仍然是可见的。

Swing 小程序由类 JApplet 表示,该类存储在 javax.swing 程序包中。可以通过创建 JApplet 的扩展类来建立 Swing 小程序。当在其中增加子类时,需要将它们加入到所谓的内容窗格中,该窗格接受小程序的子类。Swing 小程序引入了这样一个中间容器,以便处理在小程序中混合轻量级组件和重量级组件所带来的复杂性。下面的代码语句显示了一个组件是怎样加到 Swing 小程序里的内容窗格上的:

```

/ Get a handle on the applet's content pane.
Container contentPane = this.getContentPane();

// You may want to assign a new layout.
contentPane.setLayout(someLayoutObject);

// Add the component to the applet.
contentPane.add(someComponent);

```

通过调用小程序实例的以下几个方法,可以获取对于小程序分层窗格、玻璃窗格和菜单条的引用。

```

Public void getLayeredPane()
Public void getGlassPane()
Public void getJMenuBar()

```

2.2.1 JApplet 程序示例

清单 2.2 TJApplet.java 示例演示了实现一个 Swing 小程序的过程,该程序显示了一个带边界的简单标签,这个程序产生的小程序结果见图 2-3。

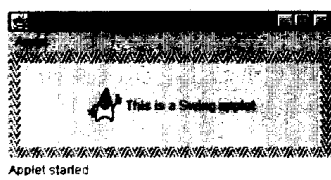


图 2-3 Swing 小程序,显示一个带简单边界的标签

清单 2.2 带简单边界的 Swing 小程序(TJApplet.java)

```

// Demonstrates the Swing applets
/*

```