



软件开发技术丛书



Java 2 图形设计

卷II: SWING

Graphic Java 2 Mastering the JFC Volume II:SWING,
3rd Edition

(美) David M. Geary 著 李建森 蒋欣军 龚尧堯 等译



机械工业出版社
China Machine Press

PTR
PH

1

清华大学出版社

◆ Java

Java 2 图形设计

卷II: SWING

清华大学出版社

清华大学出版社



软件开发技术丛书

Java 2 图形设计

卷 II : SWING

(美) David M. Geary 著

李建森 蒋欣军 龚尧莞 等译

万 华 审校



机械工业出版社
China Machine Press

Swing 是一流的 Java 图形用户界面开发工具。本书详细介绍了 Swing 的设计思想、体系结构、使用技巧，内容丰富、深入细致、分析透彻。本书用大量实例代码介绍了每个组件的用法，使初学者能很快入门；用大量图示分析了 Swing 组件的特点、结构及相互关系，使有经验的编程人员能高效利用 Swing 的强大功能。本书对掌握 Swing 技术提供了最全面的参考。

David M. Geary: Graphic Java 2 Mastering the JFC Volume II: Swing, 3rd Edition.
Authorized translation from the English language edition published by Prentice Hall PTR.
Copyright © 1999 by Sun Microsystems, Inc..
All rights reserved.

Chinese simplified language edition published by China Machine Press.
Copyright © 2000 by China Machine Press.

本书中文简体字版由美国 Prentice Hall PTR 公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，侵权必究。

本书版权登记号：图字：01-1999-2353

图书在版编目 (CIP) 数据

Java 2 图形设计 卷 II: SWING/ (美) 吉瑞 (Geary, D.M.) 著; 李建森等译.
—北京: 机械工业出版社, 2000.2

(软件开发技术丛书)

书名原文: Graphic Java 2 Mastering the JFC Volume II: SWING, 3rd Edition
ISBN 7-111-07774-1

I. J… II. ①吉…②李… III. JAVA 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 56459 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 吴 怡 瞿静华

北京京丰印刷厂印刷·新华书店北京发行所发行

2000 年 2 月第 1 版第 1 次印刷

787mm × 1092mm 1/16 · 62.25 印张

印数: 0 001 ~ 5 000 册

定价: 108.00 元 (附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

随着计算机技术和应用的普及，图形用户界面的开发愈加受到计算机应用开发人员的重视。Swing 是 Java 图形用户界面开发工具包的重要组成部分，它在原来的 Java 用户界面开发工具包 AWT（抽象窗口工具包）的基础上进行了重大改进，为 Java 应用程序开发人员提供了功能强大、业界一流的图形用户界面开发工具，使 Java 图形用户界面的开发更为实用、简单和方便。Swing 突出地体现了 Java 的简单、跨平台、面向对象和可移植等特性，必将进一步推动 Java 应用的普及。

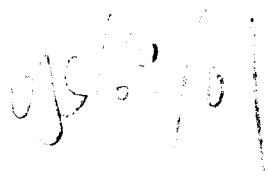
本书是关于 Swing 的权威性参考书。作者 David M. Geary 是开发 Java Soft 的 Java Management API 用户界面工具包的首席工程师。本书为 Swing 的初学者提供了大量的源代码实例，以便初学者迅速掌握 Swing 的体系结构，还为有经验的编程人员介绍了如何充分地利用 Swing 的强大功能。本书介绍了使用 Swing 的实践经验，提供了创建带有复杂图形用户界面的 Java 应用程序所必需的资源。因此，本书既适用于 Swing 的初学者，也适用于有经验的 Swing 开发人员。

本书分三大部分。第一部分介绍了 Swing 小应用程序与应用程序、Swing 与多线程、Swing 的“模型视图控制器”结构和插入式界面样式等 Swing 基础知识；第二部分详细介绍了 Swing 的所有关键组件，包括按钮、标签、进度条、滑杆、窗口与对话框、内部窗体和桌面窗格、菜单和工具条、颜色和文件选取器等，并列举了说明它们的使用方法的大量实例；第三部分附录，介绍了插入式界面样式的常量，为读者提供方便。

参加本书翻译工作的人员有：李建森、蒋欣军、龚尧尧、王彬星、卢庆龄、扬朝红、崔伟宁、李颖、周兵、张宏军、杨明立、陈华、陈豫、谢冰梅、张志弘、李玉为、李建新、杨菊梅、陈英武、邱立平、何春强、师彦彬、汪芸、赵大力、王臣太、牟志全、牛立全、王庆辰、李清、王伟、钟家杰、万志龙、吴寿虎、于云、师帅帅、宋冰清、刘一梅、钱前、黄净等。全书由李建森统稿，万华审校。

由于本书篇幅庞大，译者水平有限，加之时间仓促，书中错漏之处在所难免，敬请广大读者批评指正。

1999 年 12 月



序

不要问我计算机用户界面设计出现多少年了。有多少人拆开并重装过电传打字机？有多少人还记得什么时候电传打字机上有一个“T”字母（由于它是一个公司及其产品的名称）？令人庆幸的是，这些都已成过眼烟云。过去计算机少，任务多，所以，用户界面不得不很简单，用户必须经过训练。现在的情况完全不同了。在许多现代应用程序中，应用程序的设计要求受到了用户界面的设计要求的挑战；或者说，用户界面的设计要求甚至超过了应用程序本身的设计要求。随着基于语音和人工智能的计算机用户界面的出现，这种情况将会更加明显。

在当今世界中，如果考察一下开发人员必须使用的各种不同的编程界面，那么就不难发现，与用户交互的界面是最重要的，且是使用最广泛的。现在，用户界面不再是强加在应用程序上的附属品，而且常常是应用程序开发的核心。正是在这种背景下，我们开发了 Swing。Swing 是复杂的用户界面开发工具，是一个功能全面的、在业界具有竞争力的开发系统。

我参与了 Swing 的早期开发工作。在建立 Swing 构件（大多数早已不存在了）的过程中，在与开发团队一起工作的过程中，我得到了很多乐趣。从那时起，作为 Swing 的一个用户，我也十分开心。希望读者们也能与我一样从中获得无尽的乐趣。

不要被本书的厚度吓倒。Swing 用户界面工具包是很复杂的，其中有很多技巧。在 Swing 的设计中，一方面要满足业界对大量功能的需求，另一方面要满足 Java 编程的简单性，因为这是 Java 最令人钦佩的特性之一；这是很难取舍的。作为开发人员，处理这个矛盾的关键是理解简单与容易之间的区别。Swing 工具包不是简单的，对 Swing 的功能需求很多。但 Swing 又很容易使用，因为其中有许多简捷的方法（不妨看一下 JOptionPane 类），并且普通开发人员可以忽略 Swing 的大多数特性。

请首先概览一下本书的内容，以便了解需要掌握 Swing 的哪些内容。然后再深入进去，以概览为导向，深入到需要细致研究的地方。

Sun Microsystem 公司副总裁
James Gosling

前 言

本书是我一年多来全部热情的结晶。我力争使本书成为最全面、最准确及最深入的 Swing 技术书籍。当然，我是否达到了目的，最终将由读者们作出自己的判断。

在开始写作之前，我花了很多时间来设计本书，希望它既适用于具有一定 Swing 经验的开发人员，又适用于 Swing 的初学者。作为一个使用过大量 GUI 对象的开发人员，我认识到，学好一个 GUI 结构的最快方法是学习那些能够说明特定概念的代码实例。因此，代码实例是本书的基础，本书详细讨论了大约 300 个代码实例。

但是，仅参考代码实例对那些已经掌握了大量 Swing 知识的开发人员来说是不够的。因此，在讨论每一个 Swing 组件后都带有一个组件总结。组件总结中包含了类图、对组件属性和事件的详细介绍以及类总结（类总结讨论由该组件实现的 public 和 protected 方法），以便开发人员对组件有深入的理解。

本书的第一部分探讨了 Swing 的基础；第二部分详细介绍了 Swing 的组件，从标签和按钮到表、树和文本包，用大量的代码实例对每一个 Swing 组件都进行了介绍。例如，对表组件的介绍用了 25 个代码实例，为了说明如何充分利用树组件使用了 20 个代码实例；第三部分附录介绍了插入式界面样式的常量。

本书的读者

本书假定读者很好地掌握了 Java 语言，包括该语言中新近增加的内部类等成分。本书还假定读者初步了解 AWT；特别是代理事件模型和 Swing 所有组件的建立基础 Component 类和 Container 类。要对 AWT 体系结构和组件的彻底研究，请参见《Java 2 图形设计，卷 I：AWT》。

如何使用本书

在深入学习 Swing 组件之前，理解 Swing 的模型视图控制器（MVC）结构和插入式界面样式等基本概念是有好处的。本书第 3 章讨论了 MVC，第 7 章介绍了插入式界面样式。理解 JComponent 类提供的服务也是很重要的。JComponent 是所有轻量（lightweight）Swing 组件的最终超类。JComponent 在第 4 章讨论。

组件总结

对每个 Swing 组件的介绍都引用了说明各种组件特性的大量代码实例。在代码实例之后还提供了组件总结。组件总结是以一个组件模型表开始的，然后是 UI 代表、绘制器和编辑器，以及由该组件激发的事件。如果该组件是用来替代一个 AWT 组件的，则还列出了替换的 AWT 组件。

类图

表示了一个组件与其他对象保持的静态关系。

属性

一个组件的属性表包括属性名、属性的数据类型，以及一个属性是否是布尔型的、关联的（当改变这个属性时，激发一个属性改变事件）、受约束的（可以否决对属性的修改）、简单的（当改变这个属性时，不激发任何事件）或索引的（这个属性可以用一个参数，通常是一个整数来访问）。

有些属性是可以在初始化一个组件时指定。组件可以为属性提供“设置”和“获取”方法。指定一个属性的方式列在属性表的访问栏中。属性表还包含一个属性缺省值栏。属性表后面还有对表中列出的每一个属性的简短描述。

事件

本书为读者提供了一些代码实例，它们说明了组件的事件处理方法。例如，第 20 章“树”中提供了五个代码实例，它们说明了树鼠标、树编辑、树选取和树扩展事件的处理方式。

类总结

每一个组件总结都以一个类总结作为结束。类总结对该组件的构造方法和其他方法进行了描述。

Swing 中的程序错误

在质量方面，Swing 已经走过了漫漫长途。早期发布的 beta 版本含有大量的程序错误，其中的许多错误已相继被纠正了。但是，与许多软件一样，Swing 仍然含有程序错误。在整本书中，我都试图尽可能多地指出 Swing 中的错误，以便使开发人员避免调试代码带来的挫折感，因为调试的结果仅仅是发现由于 Swing 的错误引起的问题是没有必要的。

记住本书介绍的是 Swing 1.1FCS，这也是很重要的。当本书即将出版时，Swing 的 1.1.1 版本也已发布。1.1.1 版几乎就是一个纠错版本。因此，可以肯定，当本书放上书架时，本书中指出的错误已经被纠正了。

本书的 CD-ROM

本书所带的 CD-ROM 盘中包含了如下内容：

- JDK1.1.7 和 Swing1.1.1
- JDK1.2 和 Swing1.1FCS
- 本书中的代码实例

Swing 与 JDK

如前所述，Swing1.1.1 几乎是一个纠错版。Swing1.1.1 不能与 JDK1.2 一起使用，因此，本书附带 CD-ROM 中包含了 JDK1.1.7。Swing1.1FCS 可以与 JDK1.1.7 和 JDK1.2 一起使用。

本书中的代码实例

本书中的实例几乎都可以在随带的 CD-ROM 中找到。CD-ROM 中，本书的每一章都有一个目录，每一个例子对应相应章的子目录。这种组织方式使读者容易在 CD-ROM 中找到例子。每个例子都已编译好，可以直接运行。如果其中的例子展示了程序错误或与书中的相应程序清单

比较有所修改的话，该代码实例的目录中会包含 README.txt 文件。

这个 CD-ROM 盘中还带有《Java2 图形设计，卷 I：AWT》（该书已由机械工业出版社出版——编者注）所带的 GridBayLab 应用程序的两个版本。其中一个版本使用外部窗口，另一个例子则使用 Swing 内部窗体。这个应用程序是一个完整的 Swing 应用程序的例子，说明了 Swing 内部窗体的使用方法。另外，用这个应用程序可以探讨 GridBayLayout 布局管理器的复杂性。

小应用程序与应用程序

本书中讨论的大多数代码实例是小应用程序，但也有相当一部分例子是应用程序。如果某程序以应用程序形式而不以小应用程序形式实现，是由于该程序处理了文件，或者显示了对话框。小应用程序访问文件的能力有限。另外，在 JDK1.2 下，小应用程序显示的对话框中包含了一个警示字符串。有时，某程序段以小应用程序实现仅仅是为了例子的多样性。本书中讨论的几乎所有不处理文件的应用程序都能够很容易地以小应用程序的形式改写。

本书英文原书书号：ISBN 0-13-079667-0

英文原书出版社网站地址：www.phptr.com

目 录

译者序	
序	
前言	
第一部分 Swing 基础	1
第 1 章 简介	1
1.1 Swing 的历史	1
1.2 轻量组件与重量组件的比较	2
1.3 Swing 组件	2
1.3.1 AWT 的替代组件	3
1.3.2 Swing 增加的组件	3
1.4 J 组件	4
1.5 Swing 包概览	6
1.6 Swing 与 AWT	8
1.7 开始学习	9
1.8 Swing 资源	11
1.9 本章回顾	11
第 2 章 Swing 的基本知识	13
2.1 小应用程序与应用程序	13
2.1.1 小应用程序	13
2.1.2 JApplet 类	14
2.1.3 应用程序	17
2.1.4 JFrame 类	18
2.1.5 小应用程序/应用程序的组合	19
2.2 GJApp	21
2.3 混合使用 Swing 组件和 AWT 组件	23
2.3.1 层序	23
2.3.2 Swing 弹出式菜单	26
2.3.3 滚动	28
2.3.4 内部窗体	30
2.4 Swing 和线程	31
2.4.1 Swing 单线程设计的结果	32
2.4.2 SwingUtilities 类的 invokeLater 和 invokeAndWait 方法	32
2.5 本章回顾	40
第 3 章 Swing 组件的体系结构	41
3.1 典型的“模型-视图-控制器”体系结构	41
3.1.1 插入式视图和控制器	41
3.1.2 视图更新	42
3.2 Swing MVC	42
3.2.1 Swing 组件	44
3.2.2 静态认识	45
3.2.3 动态认识	46
3.2.4 模型	48
3.2.5 UI 代表	58
3.2.6 组件 UI 的案例	59
3.2.7 监听器	68
3.3 本章回顾	72
第 4 章 JComponent 类	73
4.1 JComponent 类概览	73
4.1.1 边框	73
4.1.2 可访问性	74
4.1.3 双缓存	75
4.1.4 调试图形	75
4.1.5 自动滚动	76
4.1.6 工具提示	77
4.1.7 键击处理和客户属性	77
4.2 JComponent 类结构	77
4.2.1 Swing 组件是 AWT 容器	78
4.2.2 最小尺寸、最大尺寸和首选尺寸	78
4.3 绘制 JComponent 组件	81
4.3.1 Swing 组件中的定制绘制	82
4.3.2 在 AWT 组件中重载绘制方法	82
4.3.3 在 Swing 组件中重载绘制方法	83
4.3.4 paint、repaint 和 update 方法	85
4.3.5 validate、invalidate 和 revalidate 方法	85
4.3.6 不透明组件与透明组件的比较	86
4.3.7 立即绘制 Swing 组件	88
4.4 双缓存	89
4.5 调试图形	96
4.6 自动滚动	100
4.7 工具提示	104
4.7.1 基于鼠标位置的工具提示	105
4.7.2 工具提示的首选位置	107

4.7.3 定制工具提示的行为	108	6.7.5 状态编辑	193
4.7.4 定制工具提示的界面样式	109	6.8 本章回顾	197
4.8 键击处理	109	第7章 插入式界面样式	198
4.9 客户属性	113	7.1 界面样式结构	198
4.10 焦点管理	116	7.1.1 界面样式	199
4.10.1 JComponent 的焦点属性	116	7.1.2 界面样式缺省值	204
4.10.2 焦点管理器	119	7.1.3 UI 管理器	208
4.11 支持可访问性	120	7.1.4 UI 资源	213
4.12 本章回顾	122	7.2 Java 界面样式	217
第5章 边框、图标和动作	123	7.2.1 客户属性	217
5.1 边框	123	7.2.2 主题	222
5.1.1 边框和边衬	123	7.3 附加 UI	223
5.1.2 Swing 的边框类型	124	7.4 本章回顾	226
5.1.3 不透明与透明之间的比较	128	第二部分 Swing 组件	227
5.1.4 边框包	129	第8章 标签与按钮	227
5.1.5 边框接口	130	8.1 JLabel 与 JButton	227
5.1.6 AbstractBorder 类	130	8.2 JLabel	229
5.1.7 边框库——共享边框	131	8.2.1 内容排列	230
5.1.8 替换内置边框	132	8.2.2 文本的位置	233
5.1.9 实现定制边框	133	8.2.3 图标/文本间隙	235
5.2 图标	135	8.2.4 许可状态	236
5.2.1 把图标与组件相关联	136	8.2.5 JLabel 属性	238
5.2.2 在组件中共享图标	137	8.2.6 JLabel 事件	239
5.2.3 图像图标	140	8.2.7 JLabel 类总结	239
5.2.4 动画的图像图标	142	8.3 按钮	241
5.3 动作	143	8.4 JButton	243
5.3.1 作为控制中心点的动作	145	8.4.1 JButton 属性	245
5.3.2 动作常量	149	8.4.2 JButton 事件	248
5.4 本章回顾	150	8.4.3 JButton 类总结	251
第6章 实用工具	152	8.4.4 AWT 兼容	260
6.1 计时器	152	8.5 本章回顾	261
6.2 事件监听器列表	158	第9章 反转按钮、复选框和单选钮	262
6.3 Swing 实用工具	160	9.1 JToggleButton 类	262
6.4 Swing 常量	166	9.1.1 JToggleButton 属性	263
6.5 BorderLayout 和 Box 类	167	9.1.2 JToggleButton 事件	263
6.5.1 BorderLayout 类	167	9.1.3 JToggleButton 类总结	264
6.5.2 Box 类	169	9.1.4 AWT 兼容	266
6.6 进度监视器	172	9.2 按钮组	266
6.6.1 ProgressMonitor	172	9.3 复选框	267
6.6.2 ProgressMonitorInputStream	176	9.3.1 JCheckBox 属性	270
6.7 撤消/重复	179	9.3.2 JCheckBox 事件	270
6.7.1 一个简单的撤消/重复样例	181	9.3.3 JCheckBox 类总结	270
6.7.2 UndoableEditSupport	185	9.4 单选钮	272
6.7.3 组合编辑	187	9.4.1 JRadioButton 属性	275
6.7.4 UndoManager	192	9.4.2 JRadioButton 事件	275

9.4.3	JRadioButton 类总结	275	10.9.5	AWT 兼容	345
9.4.4	AWT 兼容	276	10.10	JToolBar	346
9.5	本章回顾	276	10.10.1	滚过式工具条	349
第 10 章	菜单和工具条	278	10.10.2	在工具条中使用动作	350
10.1	菜单、菜单栏和工具条	278	10.10.3	浮动工具条	352
10.2	菜单和弹出式菜单	282	10.10.4	位置固定的工具提示	353
10.3	JMenuItem	283	10.10.5	JToolBar 属性	355
10.3.1	菜单项快捷键和助记符键	286	10.10.6	JToolBar 事件	356
10.3.2	JMenuItem 属性	289	10.10.7	JToolBar 类总结	356
10.3.3	JMenuItem 事件	289	10.10.8	AWT 兼容	357
10.3.4	JMenuItem 类总结	292	10.11	本章回顾	357
10.3.5	AWT 兼容	294	第 11 章	进度条、滑杆和分隔条	358
10.4	JCheckBoxMenuItem	295	11.1	JProgressBar	358
10.4.1	JCheckBoxMenuItem 属性	296	11.1.1	进度条与线程	359
10.4.2	JCheckBoxMenuItem 事件	296	11.1.2	JProgressBar 属性	363
10.4.3	JCheckBoxMenuItem 类总结	297	11.1.3	JProgressBar 事件	364
10.4.4	AWT 兼容	298	11.1.4	JProgressBar 类总结	366
10.5	JRadioButtonMenuItem	298	11.1.5	AWT 兼容	368
10.5.1	JRadioButtonMenuItem 属性	299	11.2	JSlider	368
10.5.2	JRadioButtonMenuItem 事件	299	11.2.1	填充的滑杆	368
10.5.3	JRadioButtonMenuItem 类 总结	303	11.2.2	滑杆间隔标记	369
10.5.4	AWT 兼容	304	11.2.3	滑杆标签	373
10.6	JMenu	304	11.2.4	反转滑杆值	375
10.6.1	动态修改菜单	305	11.2.5	滑杆的外延值	376
10.6.2	右拉式菜单	309	11.2.6	JSlider 属性	378
10.6.3	JMenu 属性	311	11.2.7	JSlider 事件	379
10.6.4	JMenu 事件	313	11.2.8	JSlider 类总结	380
10.6.5	JMenu 类总结	314	11.2.9	AWT 兼容	382
10.6.6	AWT 兼容	319	11.3	JSeparator	382
10.7	菜单元素	320	11.3.1	分隔条与框	385
10.8	JPopupMenu	324	11.3.2	JSeparator 属性	387
10.8.1	弹出式菜单触发器	326	11.3.3	JSeparator 事件	387
10.8.2	轻量/中量/重量弹出式 菜单	327	11.3.4	AWT 兼容	388
10.8.3	弹出式菜单调用者	328	11.4	本章回顾	388
10.8.4	JPopupMenu 属性	331	第 12 章	轻量容器	389
10.8.5	JPopupMenu 事件	333	12.1	JPanel	389
10.8.6	JPopupMenu 类总结	334	12.1.1	JPanel 的属性	391
10.8.7	AWT 兼容	338	12.1.2	JPanel 的事件	391
10.9	JMenuBar	338	12.1.3	JPanel 类总结	391
10.9.1	菜单栏菜单和组件	339	12.1.4	AWT 兼容	392
10.9.2	JMenuBar 属性	343	12.2	JRootPane	392
10.9.3	JMenuBar 事件	344	12.2.1	RootPaneContainer 接口	392
10.9.4	JMenuBar 类总结	344	12.2.2	玻璃窗格	393
			12.2.3	内容窗格	397
			12.2.4	JRootPane 属性	401

12.2.5	JRootPane 事件	402	13.4.5	JScrollBar 类总结	487
12.2.6	JRootPane 类总结	405	13.4.6	AWT 兼容	488
12.2.7	AWT 兼容	406	13.5	本章回顾	488
12.3	JLayeredPane	406	第 14 章	窗口和对话框	490
12.3.1	回顾轻量组件的层序	407	14.1	JWindow	490
12.3.2	为组件分配层	409	14.1.1	JWindow 属性	494
12.3.3	指定同一层中组件的位置	411	14.1.2	JWindow 类总结	495
12.3.4	使用拖动层	415	14.1.3	AWT 兼容	496
12.3.5	JLayeredPane 属性	420	14.2	JDialog	496
12.3.6	JLayeredPane 类总结	420	14.2.1	JDialog 属性	501
12.3.7	AWT 兼容	422	14.2.2	JDialog 类总结	501
12.4	JTabbedPane	422	14.2.3	AWT 兼容	503
12.4.1	选项卡的位置	424	14.3	JOptionPane	503
12.4.2	JTabbedPane 的属性	429	14.3.1	内部窗体	506
12.4.3	JTabbedPane 事件	430	14.3.2	用 JOptionPane 静态方法创建对话框	507
12.4.4	JTabbedPane 类总结	431	14.3.3	消息对话框	509
12.5	JSplitPane 类	433	14.3.4	确认对话框	514
12.5.1	JSplitPane 属性	439	14.3.5	输入对话框	517
12.5.2	JSplitPane 事件	439	14.3.6	选项对话框	521
12.5.3	JSplitPane 类总结	440	14.3.7	JOptionPane 属性	525
12.5.4	AWT 兼容	442	14.3.8	JOptionPane 事件	527
12.6	本章回顾	442	14.3.9	JOptionPane 类总结	530
第 13 章	滚动	443	14.3.10	AWT 兼容	538
13.1	JViewport	443	14.4	本章回顾	538
13.1.1	拖动视口中的视图	446	第 15 章	内部窗体和桌面窗格	539
13.1.2	使用 scrollRectToVisible 方法	448	15.1	JInternalFrame	539
13.1.3	JViewport 属性	452	15.1.1	JInternalFrame 属性	541
13.1.4	JViewport 事件	452	15.1.2	JInternalFrame 事件	543
13.1.5	JViewport 类总结	455	15.1.3	AWT 兼容	552
13.1.6	AWT 兼容	457	15.2	JDesktopPane	553
13.2	JScrollPane	457	15.2.1	JDesktopPane 属性	558
13.2.1	滚动窗格的头部	458	15.2.2	JDesktopPane 事件	558
13.2.2	滚动窗格的角部	464	15.2.3	JDesktopPane 类总结	558
13.2.3	JScrollPane 属性	467	15.2.4	AWT 兼容	559
13.2.4	JScrollPane 事件	468	15.3	DesktopManager	559
13.2.5	JScrollPane 类总结	468	15.4	本章回顾	567
13.2.6	AWT 兼容	472	第 16 章	选取器	568
13.3	Scrollable 接口	473	16.1	JFileChooser	568
13.4	JScrollBar	476	16.1.1	文件选取器类型	572
13.4.1	使用 Swing 的 JScrollBar 类进行手动滚动	476	16.1.2	可访问组件	576
13.4.2	块增量和单元增量	479	16.1.3	过滤文件类型	580
13.4.3	JScrollBar 属性	484	16.1.4	文件视图	589
13.4.4	JScrollBar 事件	485	16.1.5	多文件选取	593
			16.1.6	JFileChooser 属性	596

16.1.7	JFileChooser 事件	599	19.2.2	TableModel 接口	710
16.1.8	JFileChooser 类总结	603	19.2.3	AbstractTableModel	710
16.1.9	AWT 兼容	608	19.2.4	DefaultTableModel	713
16.2	JColorChooser	608	19.2.5	表格模型、缺省绘制器 和缺省编辑器	718
16.2.1	在对话框中显示颜色 选取器	610	19.3	表格列	721
16.2.2	定制颜色选取器	614	19.3.1	列调整大小模式	722
16.2.3	JColorChooser 属性	624	19.3.2	列宽度	725
16.2.4	JColorChooser 事件	624	19.4	表格列模型	731
16.2.5	JColorChooser 类总结	624	19.4.1	DefaultTableColumnModel 类	732
16.2.6	AWT 兼容	626	19.4.2	列边距	733
16.3	本章回顾	626	19.4.3	隐藏列	736
第 17 章	列表	627	19.4.4	锁定左边列	738
17.1	列表模型	628	19.5	表格选取	741
17.1.1	AbstractListModel	629	19.6	绘制和编辑	746
17.1.2	DefaultListModel	630	19.6.1	使用表格单元绘制器和编 辑器	746
17.2	列表选取	635	19.6.2	表格单元绘制器	752
17.3	列表单元绘制器	637	19.6.3	Default Table Cell Renderer 类	754
17.3.1	JList 属性	643	19.6.4	表格格式化绘制器	755
17.3.2	JList 事件	646	19.6.5	单元编辑器	756
17.3.3	JList 类总结	661	19.6.6	表格单元编辑器	757
17.3.4	AWT 兼容	665	19.6.7	实现 TableCellEditor 接口	759
17.4	本章回顾	666	19.7	表格行	767
第 18 章	组合框	667	19.7.1	行高	767
18.1	JComboBox 与 JList 的比较	667	19.7.2	绘制行	769
18.2	JComboBox 组件	667	19.8	表格装饰器	771
18.3	组合框模型	668	19.9	表格头部	779
18.3.1	ComboBoxModel	670	19.9.1	JTableHeader	779
18.3.2	MutableComboBoxModel	670	19.9.2	列头部绘制器和头部工具 提示	780
18.3.3	DefaultComboBoxModel	670	19.9.3	JTable 属性	785
18.4	组合框单元绘制器	671	19.9.4	表格事件	787
18.5	组合框键选取管理器	674	19.9.5	表格模型事件	788
18.5.1	使用缺省键选取管理器	675	19.9.6	TableColumnModel 事件	791
18.5.2	定制键选取管理器	677	19.9.7	列表选取事件	793
18.5.3	程序式的键选取	682	19.9.8	JTable 类总结	794
18.6	组合框编辑器	686	19.9.9	AWT 兼容	799
18.6.1	JComboBox 属性	696	19.10	本章回顾	799
18.6.2	JComboBox 事件	697	第 20 章	树	800
18.6.3	JComboBox 类总结	700	20.1	创建树	800
18.6.4	AWT 兼容	704	20.2	树节点	803
18.7	本章回顾	704	20.2.1	TreeNode 接口	803
第 19 章	表格	705	20.2.2	MutableTreeNode 接口	804
19.1	表格和滚动	705			
19.2	表格模型	707			
19.2.1	表格数据模型	708			

20.2.3 DefaultMutableTreeNode 类	804	22.1 JTextField	904
20.3 树路径	811	22.1.1 水平可视性和滚动偏移	907
20.4 树模型	814	22.1.2 布局单行文本域	910
20.5 树选取	822	22.1.3 使单行文本域有效	913
20.6 树单元绘制	827	22.1.4 JTextField 组件总结	916
20.6.1 DefaultTreeCellRenderer	827	22.1.5 JTextField 属性	917
20.6.2 Metal 界面样式	835	22.1.6 JTextField 事件	918
20.6.3 根节点和根句柄	835	22.1.7 JTextField 类总结	920
20.7 树单元编辑	836	22.1.8 AWT 兼容	923
20.7.1 扩展 DefaultCellEditor	837	22.2 JPasswordField	923
20.7.2 DefaultTreeCellEditor	838	22.2.1 JPasswordField 组件总结	924
20.8 绘制和编辑: 学习一个样例	842	22.2.2 JPasswordField 属性	924
20.8.1 Test 类	843	22.2.3 JPasswordField 类总结	925
20.8.2 SelectableFile 类和 FileNode 类	845	22.3 JTextArea	926
20.8.3 绘制器	846	22.3.1 JTextArea 组件总结	929
20.8.4 编辑器	848	22.3.2 JTextArea 属性	929
20.8.5 JTree 属性	851	22.3.3 JTextArea 类总结	930
20.8.6 树事件	852	22.3.4 AWT 兼容	932
20.8.7 JTree 类总结	863	22.4 JEditorPane	932
20.8.8 AWT 兼容	868	22.4.1 JEditorPane 属性	934
20.9 本章回顾	868	22.4.2 JEditorPane 事件	935
第 21 章 文本基础	869	22.4.3 JEditorPane 类总结	936
21.1 Swing 文本组件	869	22.5 JTextPane	939
21.2 动作	871	22.5.1 嵌入图标和组件	939
21.2.1 文本动作	871	22.5.2 用属性标记内容	941
21.2.2 动作和编辑工具包	875	22.5.3 JTextPane 属性	947
21.3 键映射	877	22.5.4 JTextPane 类总结	947
21.4 文档	880	22.6 AWT 兼容	949
21.4.1 定制文档	882	22.7 本章回顾	949
21.4.2 文档监听器	883	第 23 章 定制文本组件	950
21.5 加字符与加重器	887	23.1 概览	950
21.5.1 加字符	887	23.2 属性集和风格常量	952
21.5.2 加字符监听器	888	23.3 定制动作	954
21.5.3 定制加字符	889	23.4 视图	958
21.5.4 加重器	891	23.5 风格和风格的相关内容	962
21.6 撤销/恢复	893	23.6 元素	968
21.7 JTextComponent	897	23.7 本章回顾	971
21.8 本章回顾	903	第三部分 附录	972
第 22 章 文本组件	904	附录 A 类图	972
		附录 B 插入式界面样式常量	975

第一部分 Swing 基础

第 1 章 简介

Java 的基础类 (JFC) 是开发图形用户界面的 API 集。Java 的基础类包括以下 API:

- 抽象窗口工具包 (版本 1.1 及以后的版本)。
- 2D API。
- Swing 组件。
- 可访问性 API。

抽象窗口工具包 (Abstract Window Toolkit, AWT) 是 Java 开发用户界面最初的工具包。AWT 是建立 JFC 的主要基础,《Java 2 图形设计, 卷 I: AWT》对 AWT 有详细的介绍。

2D API 提供了 AWT 所缺乏的附加图形功能。例如, AWT 对图形操作只提供了一种规格的笔——即一个像素大小的正方形。而 2D API 除提供了不同大小的笔外, 还提供了丰富的二维着色能力。《Graphic Java 2, Volume IV: 2D API》一书中对 2D API 有详细的介绍。

Swing 是建立在 AWT 之上的[○]、包括大多数轻量组件的组件集。除提供了 AWT 所缺少的、大量的附加组件外, Swing 还提供了替代 AWT 重量组件的轻量组件。Swing 还包括了一个使人印象深刻的、用于实现包含插入式界面样式等特性的图形用户界面的下层构件。因此, 在不同的平台上, Swing 组件都能保持组件的界面样式特性, 如双缓冲、调试图形和文本编辑包等。

可访问性 (Accessibility) API 是一个类集, 其中的类使 Swing 组件能够与用于残疾用户的援助技术交互。JFC 还包括许多可访问性工具, 这些工具与可访问性 API 联合使用。

1.1 Swing 的历史

要了解 Swing, 首先必须了解 AWT, AWT 是 Swing 的基础。

Java 的发展速度超出了人们的想象, Java API 中最可视的部分——AWT 突然成为了人们关注的焦点。遗憾的是, 原来的 AWT 不能满足发展的需要。

原来的 AWT 不是为许多开发人员使用的、功能强大的用户界面 (UI) 工具包而设计的, 其设计目的是支持开发小应用程序中的简单用户界面。例如, 原来的 AWT 缺少许多面向对象 UI 工具包中所能见到的特性, 例如, 剪贴板、打印支持和键盘导航等特性在 AWT 中都不存在。原来的 AWT 甚至不包括弹出式菜单或滚动窗格等基本特性, 而弹出式菜单和滚动窗格是开发现代用户界面的两个基本元素。

此外, AWT 的下层构件还有严重的缺陷。人们使 AWT 适应基于继承的、具有很大伸缩性的事件模型。甚至更糟, 基于对等组件 (peer) 的体系结构也被用于 AWT, 该体系结构注定要成为 AWT 的致命弱点。

为了尽快推向市场和保持本地的界面样式, 于是产生了基于对等组件的体系结构, 而该体

○ 参见 1.2 节“轻量组件与重量组件的比较”中对轻量组件与重量组件比较的讨论。

系结构注定是要失败的。对等组件是完成薄弱的 AWT 对象所委托任务的本地用户界面组件。对等组件负责完成所有的具体工作，包括绘制自己、对事件做出反应等，这使得 AWT 组件除了在适当的时间与其对等组件交互外无事可做。由于 AWT 类只是较复杂的本地对等组件的外壳，所以，AWT 的早期开发人员能在最快的时间^①内创建组件。例如，`java.awt.Panel` 类只包含十二行代码。

另外，对等组件的设计也有严重的缺点。首先，在大多数平台上，对等组件都是在本地窗口中绘制的。每个组件一个本地窗口实在不能得到高性能，为此，含有大量 AWT 组件的小应用程序付出了很高的性能代价。

把不同平台上的本地对等组件硬塞进 Java 框架中也是一个问题，使这些 AWT 组件跨平台的表现一致是完全不可能的。结果，不但没有实现急需的新组件，而且开发时间都浪费在修补对等组件的错误上和不兼容问题上了。

更糟的是，AWT 有很高的错误发生率。于是，第三方开始提供他们自己的工具包，这些工具包提供了更可靠的下层构件并提供了比 AWT 更多的功能。这些工具包之一是 Netscape 的 Internet 基础类 (IFC)，IFC 是一组建立在 NEXTSTEP 中的用户界面工具包概念基础上的一组轻量类。IFC 组件不是对等的，在许多方面胜过了 AWT 组件。IFC 还吸引了更多的开发人员加盟。

由于认识到 Java 领域很可能在标准用户界面工具包问题上出现分裂局面，JavaSoft 和 Netscape 达成了一个交易，共同实现 Java 基础类 (Apple 公司和 IBM 公司也参加了 IFC 的开发)。Netscape 开发人员与 Swing 工程师一起合作，以便把大部分的 IFC 的功能嵌入到 Swing 组件中。

起初打算让 Swing 类似于 Netscape 的 IFC。然而，随着时间的推移，在增加了插入式界面样式等特性并修改了设计之后，Swing 大大地偏离了它原来的目标。随着 Swing 1.1 版本的推出，虽然大量的 IFC 技术仍然嵌在 Swing 中，但是，Swing 与 IFC 相似的部分已大部分消失了。今天，在一个功能全面的用户界面工具包中，Swing 提供了 AWT 和 IFC 中最优秀的成份。

1.2 轻量组件与重量组件的比较

轻量组件首次出现在 AWT 1.1 版本中。AWT 最初只包括与本地对等组件相关联的重量组件，这些组件在它们自己的本地不透明窗口中绘制。

相反，轻量组件没有本地对等组件，而且在它们的重量容器的窗口中绘制。

由于轻量组件不在本地不透明的窗口中绘制，因此，它们可以有透明的背景。透明的背景使显示的轻量组件可以是非矩形的，虽然所有组件（重量的或轻量的）都有一个矩形边框。

Swing 组件几乎都是轻量组件，那些顶层容器：窗体、小应用程序、窗口和对话框除外。因为轻量组件是在其容器的窗口中绘制的，而不是在自己的窗口中绘制的，所以轻量组件最终必须包含在一个重量容器中。因此，Swing 的窗体、小应用程序、窗口和对话框都必须是重量组件，以便提供一个可以在其中绘制 Swing 轻量组件的窗口。

1.3 Swing 组件

Swing 包含 250 多个类，是组件和支持类的集合。Swing 提供了 40 多个组件，是 AWT 组件

^① 原来的 AWT 是在不足六个星期的时间内开发出来的。