

Java图形界面开发学习笔记——从新手到高手的跨越



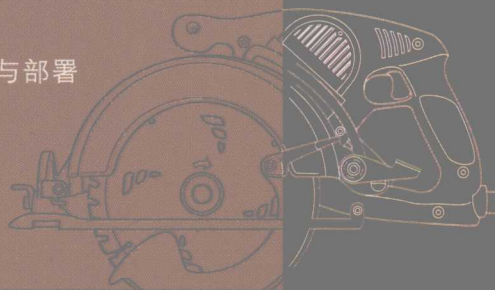
Java Swing

图形界面

开发与案例详解

王鹏 何昀峰 编著

- 涵盖Java Swing图形界面开发必须掌握的所有常用知识
- 通过**105个**具有典型性和实用价值的实例学习组件应用、开发与部署
- 通过**77个**习题巩固所学知识
- 通过1个综合实例进一步提高读者界面设计开发的能力

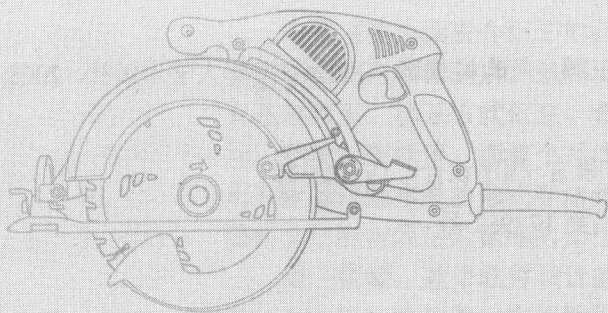


清华大学出版社

Java Swing

图形界面开发与案例详解

王鹏 何昀峰 编著



清华大学出版社
北 京

内 容 简 介

Java Swing 是目前图形界面设计的主流开发工具,本书从实用的角度出发,通过大量实例全面介绍 Java Swing 中各种组件的应用及图形界面的开发技术。

全书共 20 章,其中第 1~2 章主要介绍有关 Swing 的基础知识,包括 Swing 的基本概述、如何使用 IDE 开发 Swing 程序;第 3~15 章结合众多实例和问题介绍最常用、最有用的组件及其应用方法,包括标签和按钮组件、布局管理器组件、面板组件、列表框组件、下拉列表框组件、进度条组件、时间组件、滑块组件、分隔条组件、选取器组件、文本组件、窗口组件、对话框组件、JApplet 组件、菜单组件、工具条组件、表格组件、树组件以及组件的事件处理机制等;第 16~19 章主要讲述有关 Swing 图形开发的其他知识,包括观感器、Swing 线程与并发、模型与架构等。第 20 章通过一个综合实例使读者能够在实际开发中理解和巩固所学知识,从而提高综合应用能力。

本书几乎涵盖了目前 Java Swing 图形开发必备的所有常用知识,拥有丰富的实例,且这些实例均来自于工程实践,本书适合 Java 程序语言的初学者阅读,也可供具有一定编程经验的初级编程人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java Swing 图形界面开发与案例详解/王鹏,何昀峰编著.—北京:清华大学出版社,2008.12

ISBN 978-7-302-18904-6

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

中国版本图书馆 CIP 数据核字(2008)第 177962 号

责任编辑:夏非彼 张楠

装帧设计:图格新知

责任校对:贾淑媛

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:190×260 印 张:26.25 字 数:672 千字

版 次:2008 年 12 月第 1 版 印 次:2008 年 12 月第 1 次印刷

印 数:1~4000

定 价:49.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:030557-01

前言

Preface

Java 技术的强大生命力来自于它所蕴含的面向对象和设计模式思想，由此开发出来的程序，不仅具有非常美观的艺术性，而且具有非常高的重用性，这种重用性使开发工作变得非常轻松。

Java 的 Swing 技术，因 Java 强大的生命力而被赋予非常广阔的展示舞台，Swing 技术从编码时起就能承载诸多 Java 设计理念，所以，您将在相对比较“繁琐”的 Swing 代码里体会 Java 的乐趣。“Swing 组件”、“事件处理监听器”……，这些词语目前或许仅仅是一个个饶舌的词汇，但当您阅读完本书后将会发现，它们从实现角度来看是小巧的艺术品，从应用角度来看，更能让您在使用后得到无限启发。

为了使您尽快能够编写 Swing 代码、快速掌握 Java Swing 图形开发的技巧及应用方法，在本书中笔者根据自己的实践经验，从零基础讲起，回避一些高深但不实用的知识点，专门讲解实用性强的技术。“实例化讲解”是本书的一大特色，对于书中讲述的绝大多数知识点，都会给出相应难度的实例程序，这样做的目的就是让您能够理论联系实际，所以在阅读本书时，可以采用“阅读基础知识→运行代码→深入学习”的方式，从而在实际操作中熟悉 Java Swing 图形开发技术。

本书特点

与其他书籍相比，本书具有以下特点。

- 分解知识点，逐个掌握：本书几乎覆盖 Java Swing 图形开发所需掌握的所有常用知识点。
- 实例丰富，易学易用：本书的一大特色就是拥有大量的实例，能够让读者根据实例来进一步清楚地理解所介绍的知识点。
- 图例丰富，学习轻松：在具体讲述知识点时，为了让读者从开始就能理解这个知识的原理，本书提供了详细的图例，说明这些知识是如何设计、如何实现、如何应用的，并在图例中稍显复杂的地方，提供了详细的标注，让读者一看就明白整个知识点的设计原理和实现手段。
- 讲解通俗，步骤详细：在介绍大型实例时，制作步骤都以通俗易懂的语言阐述，并穿插讲解和技巧文字，在阅读时就像听课一样详细而贴切。读者只需要按照步骤操作，即可轻松完成一个实例的创建，不但掌握了开发的步骤，还能掌握开发的技巧。

本书内容

本书具有以下内容：

- 第 1 章主要介绍 Swing 的基本知识，其中包括 Swing 以及其前身 AWT 的背景知识。

- 第2章主要介绍 Java Swing 的开发工具, 通过举例来为读者详细讲解如何通过开发工具来开发 Java Swing 图形界面。
- 第3章主要介绍 Swing 中组件的基本知识。
- 第4章主要介绍有关标签组件和按钮组件的使用方法。
- 第5章主要介绍有关布局管理器的知识, 通过布局管理器来将组件排列在容器内, 从而构成一个完美的界面。
- 第6章主要介绍有关中间容器, 即面板方面的知识。
- 第7章主要介绍如何处理 Swing 事件的发生。
- 第8章主要介绍有关列表框和下拉列表框的使用方法。
- 第9章主要介绍有关进度条组件、时间组件、滑块组件和分隔条组件的知识。
- 第10章主要介绍有关文件选取器和颜色选取器的使用方法。
- 第11章主要介绍文本组件的使用以及各种文本组件的使用环境。
- 第12章主要介绍有关窗口的知识, 其中包括一般窗口、对话框和 JApplet。
- 第13章主要介绍有关工具条和菜单的使用方法。
- 第14章主要介绍有关表格的创建以及使用方法。
- 第15章主要介绍有关树组件的创建以及使用方法。
- 第16章主要介绍有关界面观感器的使用。
- 第17章主要介绍 Java 语言中如何对数据进行处理。
- 第18章主要介绍 Swing 的模型架构, 包括 MVC 和可分离的 MVC。
- 第19章主要介绍平时在开发中经常使用的细微知识点。
- 第20章主要为读者列举一个综合实例——通讯录系统的开发方法和思路。

目标读者

本书具有知识全面、实例精彩、指导性强的特点, 力求以全面的知识及丰富的实例来指导读者掌握 Java Swing 图形编程技术。本书适合以下读者阅读:

- Java 程序语言的初学者。
- 具有一定编程经验的初级编程人员。

资源下载与答疑

可以登录图格新知网站 <http://www.booksaga.com> 下载本书所有实例的源代码。在学习过程中, 遇到疑难问题, 可以通过以下方式与我们联系: booksaga@126.com。

本书由王鹏、何昀峰编著, 此外, 参与本书编写和资料整理的还有彭纯祥、毕利、左中凯、蒋勇、许丹、刘伟、吴军、刘龙、何长亮、刘大勇、陈强、王伟、方杰、潘磊、郑见、徐之祥、宋磊等, 在此, 编者对他们表示衷心的感谢。

编者

2008年9月

目 录

Contents

第 1 章 Java Swing 概述.....	1
1.1 什么是 Java Swing.....	1
1.1.1 Swing 的发展史.....	1
1.1.2 Swing 的功能.....	2
1.1.3 Swing 的特性.....	2
1.2 Java Swing 的包结构.....	3
1.3 一个 Java Swing 程序实例.....	5
1.4 本章小结.....	6
1.5 本章习题.....	6
第 2 章 如何使用 IDE 开发 Swing 程序.....	7
2.1 如何利用 Eclipse 开发 Swing 程序.....	7
2.2 如何利用 JBuilder 开发 Swing 程序.....	11
2.3 如何利用 NetBeans 开发 Swing 程序.....	16
2.4 本章小结.....	19
2.5 本章习题.....	19
第 3 章 Java Swing 组件基础.....	20
3.1 Swing 组件类的层次.....	20
3.2 Window 类.....	22
3.2.1 顶层容器类和包含层次.....	22
3.2.2 在顶层容器中添加组件.....	23
3.2.3 在顶层容器中添加菜单栏.....	25
3.3 JComponent 类.....	27
3.4 本章小结.....	31
3.5 本章习题.....	32

第4章 如何使用标签和按钮组件.....	35
4.1 如何使用标签.....	35
4.2 如何使用按钮.....	37
4.2.1 如何使用普通按钮.....	38
4.2.2 如何使用单选按钮.....	39
4.2.3 如何使用复选框.....	42
4.2.4 按钮组件的实例应用.....	43
4.3 本章小结.....	45
4.4 本章习题.....	45
第5章 如何使用布局管理器组件.....	48
5.1 布局管理器概述.....	48
5.2 布局管理器的种类.....	49
5.2.1 BorderLayout.....	49
5.2.2 FlowLayout.....	53
5.2.3 GridLayout.....	56
5.2.4 GridBagLayout.....	59
5.2.5 CardLayout.....	61
5.2.6 BoxLayout.....	65
5.2.7 SpringLayout.....	67
5.2.8 GroupLayout.....	68
5.3 自定义布局管理器的创建.....	70
5.4 本章小结.....	70
5.5 本章习题.....	71
第6章 如何使用面板组件.....	73
6.1 如何使用 JPanel.....	73
6.2 如何使用 JScrollPane.....	76
6.3 如何使用 JSplitPane.....	78
6.4 如何使用 JTabbedPane.....	81
6.5 如何使用 JInternalFrame.....	84
6.6 如何使用 JLayeredPane.....	86
6.7 如何使用 JRootPane.....	88
6.8 本章小结.....	90
6.9 本章习题.....	90

第 7 章 Swing 事件处理机制	92
7.1 Swing 事件处理机制概述	92
7.2 Swing 中的监听器	93
7.2.1 事件处理的过程与步骤	93
7.2.2 匿名类方式处理事件	94
7.2.3 适配器类	95
7.2.4 Swing 所支持的事件监听器	96
7.2.5 窗口事件的处理	96
7.2.6 动作事件的处理	99
7.2.7 焦点事件的处理	100
7.3 本章小结	102
7.4 本章习题	102
第 8 章 如何使用列表框和下拉列表框组件	105
8.1 如何使用列表框 JList	105
8.1.1 使用数组方式创建列表框	105
8.1.2 使用 Vector 方式创建列表框	106
8.1.3 使用 ListModel 方式创建列表框	108
8.1.4 列表框选取事件的处理	111
8.1.5 列表框双击事件的处理	112
8.2 如何使用下拉列表框 JComboBox	115
8.2.1 使用数组和 Vector 创建下拉列表框	115
8.2.2 使用 ComboBoxModel 创建下拉列表框	116
8.2.3 下拉列表框的事件处理	119
8.3 本章小结	121
8.4 本章习题	121
第 9 章 如何使用进度条、时间、滑块和分隔条组件	123
9.1 如何使用进度条组件 JProgressBar	123
9.2 如何使用时间组件 Timer	125
9.3 如何使用滑块组件 JSlider	127
9.4 如何使用分隔条组件 JSeparator	129
9.5 本章小结	130
9.6 本章习题	130

第 10 章 如何使用选取器组件.....	132
10.1 如何使用文件选取器 JFileChooser	132
10.1.1 如何创建 JFileChooser 组件	133
10.1.2 如何创建 JFileChooser 对话框	138
10.2 如何使用颜色选取器 JColorChooser	141
10.3 本章小结	143
10.4 本章习题	143
第 11 章 如何使用文本组件.....	145
11.1 文本组件概述	145
11.2 如何使用普通文本组件	146
11.2.1 如何使用 JTextField	146
11.2.2 如何使用 JPasswordField	151
11.2.3 如何使用 JFormattedTextField	154
11.3 如何使用文本区组件	155
11.4 如何打印文本组件	167
11.5 本章小结	168
11.6 本章习题	168
第 12 章 如何使用窗口、对话框和 JApplet 组件.....	171
12.1 如何使用窗口组件	171
12.2 如何使用对话框组件	174
12.3 如何使用 JApplet 组件.....	180
12.4 本章小结	182
12.5 本章习题	182
第 13 章 如何使用菜单和工具条组件.....	185
13.1 如何使用菜单组件	185
13.1.1 菜单组件的类层次	185
13.1.2 如何创建菜单	187
13.1.3 如何处理菜单事件	191
13.1.4 如何响应键盘操作	193
13.1.5 如何使用弹出式菜单	196
13.1.6 如何使用菜单项的启用和禁用功能	198
13.1.7 如何创建复选框菜单项	199

13.1.8	如何创建单选按钮菜单项	201
13.1.9	如何定义个性化菜单	202
13.1.10	菜单组件的常用 API	207
13.2	如何使用工具条组件	207
13.2.1	如何创建工具条	208
13.2.2	如何定义个性化工具条	209
13.2.3	工具条组件的常用 API	212
13.3	本章小结	212
13.4	本章习题	212
第 14 章	如何使用表格组件	214
14.1	如何创建一个表格	214
14.2	如何把表格加入容器	217
14.3	如何设置表格列宽	221
14.4	如何创建表格模型	222
14.5	如何监听数据变化	227
14.6	如何使用选择器	229
14.7	如何使用编辑器和渲染器	232
14.8	如何使用自定义渲染器	236
14.9	如何为单元格指定文字说明	238
14.10	如何为表头指定文字说明	239
14.11	如何使用排序和过滤	240
14.12	如何使用组合框作为编辑器	242
14.13	如何使用其他编辑器	243
14.14	如何使用编辑器验证文本	245
14.15	如何打印表格	247
14.16	本章小结	247
14.17	本章习题	247
第 15 章	如何使用树组件	249
15.1	如何创建树	249
15.2	如何创建数据模型	253
15.3	如何处理节点事件	255
15.3.1	如何处理 TreeModelEvent 事件	255
15.3.2	如何处理 TreeSelectionEvent 事件	263

15.4	如何定义个性化树	269
15.5	树组件的常用 API	269
15.6	本章小结	270
15.7	本章习题	270
第 16 章	如何使用 Swing 观感器	273
16.1	如何设置程序的观感	273
16.2	如何自定义观感器	276
16.3	本章小结	281
16.4	本章习题	281
第 17 章	Swing 与并发	282
17.1	多线程问题	282
17.2	初始线程	283
17.3	事件分派线程	284
17.4	工作线程	286
17.4.1	简单的背景任务	287
17.4.2	拥有临时结果的任务	292
17.4.3	取消背景任务	295
17.4.4	绑定属性和状态方法	297
17.5	本章小结	298
17.6	本章习题	298
第 18 章	Swing 模型架构	300
18.1	传统的 MVC 设计模式	300
18.2	可分离的模型架构	302
18.3	本章小结	303
18.4	本章习题	303
第 19 章	Swing 的其他特性	304
19.1	如何在 Swing 组件中使用 HTML	304
19.2	如何使用边框	309
19.2.1	如何使用 Swing 中的边框	309
19.2.2	如何创建自定义边框	314
19.2.3	边框组件的常用 API	315
19.3	如何使用图标	316

19.4	如何使用动作	318
19.5	如何支持辅助技术	321
19.6	如何使用焦点子系统	321
19.7	如何使用键绑定	326
19.8	如何在对话框中使用 Modality	328
19.9	如何创建 Splash Screen	332
19.10	如何使用 System Tray	332
19.11	如何使用 Swing 拖曳功能和数据传输	333
19.12	本章小结	336
19.13	本章习题	336
第 20 章	Swing 实现通讯录系统	338
20.1	通讯录系统的软件框架	338
20.2	通讯录系统的登录系统	339
20.3	通讯录系统的主菜单系统	342
20.3.1	数据库模块的设计	343
20.3.2	信息界面模块的设计	360
20.3.3	功能模块的设计	374
20.3.4	其他模块的设计	394
20.3.5	TabbedPane 容器框架的设计	397
20.3.6	主菜单的设计	398
20.4	本章小结	407

第1章 Java Swing 概述

Java Swing 的技术发展到现在,已经被许多开发人员作为图形开发的首选。相对于 AWT 技术来说,Java Swing 技术有过之而无不及。在这里将会通过对 Java Swing 发展史、功能、特性等方面的学习,使读者能够清晰地了解 Java Swing 成为图形开发的主流的原因。应用 Java Swing 进行图形开发时,最重要的就是要学会熟练地应用 Java Swing 包中提供的各种各样的 API,从而为以后的学习和开发奠定扎实的基础。

1.1 什么是 Java Swing

Swing 是一个用于开发 Java 图形界面应用程序的开发工具包,它是以抽象窗口工具包(AWT)为基础,使跨平台应用程序可以使用任何可插拔的外观风格。Swing 开发人员通过使用少量的代码,就可以利用 Swing 包中丰富、灵活的功能和模块化组件类来开发出令人满意的用户界面。

1.1.1 Swing 的发展史

由于 Swing 是以 AWT 为基础的,所以在学习 Swing 之前,首先必须要了解什么是 AWT,AWT 是 Abstract Window Toolkit (抽象窗口工具集)的全称,在 AWT 包中拥有很多组件类,这些组件类是被用来开发图形程序的基础。而 Swing 则是在 AWT 的基础上对这些组件进行了修改和升级等,从而避免了在 AWT 开发中遇到的一些问题。AWT 被 Swing 取代,主要是因为包中的组件类已经不能满足日益增长的客户要求。

其实,AWT 一开始并不是专门为用户界面(UI)工具包而设计的,其初衷是用来开发小的应用程序中的图形界面,所以针对图形界面开发时,AWT 中的功能很少,并且很多重要的功能在 AWT 中都不具备,例如剪贴板、打印支持和键盘导航等特性,AWT 包中甚至没有弹出式菜单或滚动窗格等基本特性,而这两个功能是目前图形开发中不可缺少的。

另外,AWT 有很高的错误发生率。正是因为这一点,第三方开始提供他们自己的工具包,这些工具包的可靠性高,且比 AWT 具有更多功能的组件构架。这些工具包之一就是 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,Java 和 Netscape 达成了一个共识,即共同实现 Java 基础类。Netscape 开发人员与 Swing 工程师一起合作,以便把大部分的 IFC 功能嵌入到 Swing 组件中。

起初打算让 Swing 类似于 Netscape 的 IFC。但是随着时间的推移,在增加了插入式界面

样式等特性并修改设计后, Swing 大大偏离了原来的目标。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 类似的大部分已消失。目前, Swing 可以被称为一个功能全面的用户界面工具包, 它拥有 AWT 和 IFC 中最优秀的部分, 并且非常注重个性化的设计。

1.1.2 Swing 的功能

本节将通过与 AWT 进行比较来详细描述有关 Swing 的一些与众不同的功能。

- AWT 是 Abstract Window Toolkit (抽象窗口工具包) 的缩写, 这个工具包提供了一套与本地图形界面进行交互的接口。AWT 中的图形函数与操作系统所提供的图形函数之间有着一一对应的关系, 也就是说, 当使用 AWT 包来创建图形用户界面时, 实际上就是在使用本地操作系统所提供的图形库来绘制图形界面。
- 由于不同操作系统的图形库所提供的功能是不一样的, 在一个平台上存在的功能在另外一个平台上则可能不存在, 为了实现 Java 语言的“一次编译, 到处运行”的概念, AWT 只能通过牺牲功能来实现其平台无关性, 也就是说, AWT 所提供的图形功能是各种不同操作系统所提供的图形功能的交集。由于 AWT 是依靠本地操作系统的方法来实现其功能的, 所以通常把 AWT 控件称为重量级控件。
- Swing 是在 AWT 的基础上发展起来的一套新的图形界面开发工具, 它提供了 AWT 所能够提供的所有功能, 并且用纯粹的 Java 代码对 AWT 的功能进行扩充。在实际开发中, 并不是所有的操作系统都能够支持树型控件, 而 Swing 则是利用了 AWT 中所提供的基本作图方法对树型控件进行模拟。由于 Swing 控件是用纯粹的 Java 代码实现的, 因此在一个平台上设计的树型控件可以在其他平台上使用。由于在 Swing 中没有使用本地操作系统的内在方法来实现图形功能, 所以通常把 Swing 控件称为轻量级控件。

正是由于 Swing 的以上功能, 它自然而然地代替了 AWT 成为 Java 图形开发的首选。

1.1.3 Swing 的特性

本节将通过表格的形式为读者一一列出有关 Swing 的宏观特性, 如表 1.1 所示。

表 1.1 Swing 特性一览表

特性	描述
Swing GUI 组件	包括从按钮、分隔窗格到表格的所有组件
可插式外观感觉	允许任何使用 Swing 组件的程序选择其外观感觉
Accessibility (无障碍功能) API	支持辅助技术, 例如屏幕阅读器和点字显示器, 帮助用户获得信息
Java 2D API	允许开发人员在应用程序和 Applet 中方便地使用 2D 图形、文本以及图像
拖放支持	为 Java 应用程序和本机应用程序之间提供了拖放功能
国际化	允许创建与全世界使用不同语言 and 不同文化习俗的用户进行交互的应用程序。通过输入法的架构, 开发人员可以创建能够接收各种不同字符的语言文字的应用程序

正是因为 Swing 拥有以上显著的特性，使得开发人员越来越热衷于使用 Swing 进行图形开发。

1.2 Java Swing 的包结构

Java Swing 由许多包组成，下面将通过表的形式列出这些包及其功能，说明如表 1.2 所示。

表 1.2 Java Swing 包结构及其功能

包名称	功能简介
javax.swing	Swing 组件和实用工具
javax.swing.border	Swing 轻量组件的边框
javax.swing.colorchooser	JColorChooser 的支持类（接口）
javax.swing.event	事件和侦听器类
javax.swing.filechooser	JFileChooser 的支持类（接口）
javax.swing.pending	未完全实现的 Swing 组件
javax.swing.plaf	抽象类，用于定义 UI 代表的行为
javax.swing.plaf.basic	实现所有标准界面样式公共功能的基类
javax.swing.plaf.metal	用户界面代表类，用于实现 Metal 界面样式
javax.swing.table JTable	组件的支持类
javax.swing.text	支持文档的显示和编辑
javax.swing.text.html	支持显示和编辑 HTML 文件
javax.swing.text.html.parser html	文件的分析器类
javax.swing.text.rtf	支持显示和编辑 RTF 文件
javax.swing.tree JTree	组件的支持类
javax.swing.undo	支持取消操作

下面将针对常用包给予详细说明。

- swing 包是 Swing 提供的最大包，包含将近 100 个类和 25 个接口。几乎所有的 Swing 组件都在 swing 包中，只有 JTableHeader 和 JTextComponent 例外，它们分别存放在 swing.table 包和 swing.text 包中。
- swing.border 包中含有多个在轻量 Swing 组件的边衬中用于绘制边框的类。border 包由一个 Border 接口、一个 AbstractBorder 类和 AbstractBorder 的许多具体扩展组成。
- swing.event 包中定义了事件和事件监听器类，swing.event 包与 AWT 的 event 包类似。awt.event 和 swing.event 都包含事件类和监听器接口，它们分别响应由 AWT 组件和 Swing 组件激发的事件，例如当在树组件中需要节点扩展（或折叠）的通知时，则要实现 Swing 的 TreeExpansionListener 接口，并把一个 TreeExpansionEvent 实例传送给 TreeExpansionListener 接口中定义的方法。TreeExpansionListener 和 TreeExpansionEvent 都是在 swing.event 包中定义的。

- swing.pending 包包括仍没有完全实现的 Swing 组件。在 Swing 1.1 FCS 中，pending 包包含选择器（日期、货币选择器）、计算器、弹出式按钮等。pending 包中的组件最终会放置到 swing 包中。
- 虽然 Swing 的表格组件 (JTable) 存放在 swing 包中，但它的支持类却在 swing.table 包中。表格模型、单元绘制器和编辑器等都在 swing.table 包中。
- 与 JTable 类相似，Swing 的 JTree 类（用于按层次组织数据的结构组件）也在 swing 包中，而它的支持类却存放在 swing.tree 包中。swing.tree 包提供树模型、树节点、树单元编辑器和绘制器等支持类。
- Swing 中包含 4 个用于显示和编辑文档的包，即 swing.text、swing.text.html、swing.text.html.parser 和 swing.text.rtf。swing.text 包为 Swing 的文档模型提供了所有必须的下层构件，包括用于文档、元素、加字符、增亮、编辑工具包等的类和接口；swing.text.html 和 swing.text.rtf 包是 Swing 最小的两个包，它们分别提供用于实现 HTML（超文本标记语言）和 RTF（多文本格式）文档编辑器的编辑器工具包。swing.text.html.parser 包中含有分析 HTML 文件的支持类。swing.undo 包为实现取消操作提供支持。
- swing.plaf 包中的类形成了 Swing 插入式界面样式的 UI 代表部分的基础。UI 代表为它们的相关组件实现界面模式。
- swing.plaf 包中的大多数类定义 UI 资源或扩展 swing.ComponentUI 类。swing.ComponentUI 类定义所有 UI 代表的公共行为。在 swing.plaf 包中的 UI 代表类（即以 UI 结尾的类）通常为特定的组件定义附加的抽象方法，例如 swing.plaf.ButtonUI 类扩展 swing.ComponentUI 类并添加了抽象方法 getDefaultMargin()，该方法返回按钮边框与按钮内容之间的间距。
- swing.plaf.basic 包扩展在 swing.plaf 包中定义的类，并且实现所有标准 Swing 界面样式共有的特性，例如 swing.plaf.basic.BasicButtonUI 类提供了 swing.ComponentUI 类和 swing.plaf.ButtonUI 类定义的一些方法的默认实现。BasicButtonUI 类还为 paint 等方法提供了多个不同的实现（paint 方法用于绘制按钮的文本和图标）。BasicButtonUI 以无操作形式实现其他与界面样式相关的方法，如 paintButtonPressed() 方法。这些方法将被特定的界面样式扩展所重载。
- metal 和 motif 包是用于实现相应界面样式的 UI 代表类。通常，UI 代表类扩展自 swing.plaf.basic 包中的类，例如 metal.ButtonUI 类重载 swing.plaf.basic.ButtonUI 类中的 paintButtonPressed 方法的无操作实现。
- Swing.plaf.multi 包支持界面样式复用。界面样式复用允许多个 UI 代表与一个组件相关联，例如一个按钮 UI 代表可能同时与一个视觉 UI 代表和一个音频 UI 代表相关联，这样，当这个按钮被激活时，不仅能产生可视的反馈信息，还能播放一个声音。UI 复用的主要用途是使组件更具可访问性。
- Swing 还提供了其他两种界面样式实现，即 Macintosh 和 Organic 的界面样式。

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 组件的 4 倍。除提供替代 AWT 重量组件的轻量组件外，Swing 还提供了大量有助于开发图形用户界面的附加组件。

1.3 一个 Java Swing 程序实例

上节为读者详细介绍了有关 Java Swing 包的结构，并且针对每个包讲述了其功能，本节将通过一个实例进行讲解，从而使读者能够更加轻松地学习本书后面的知识。

以下实例用于显示一个带有 `helloswing` 标题的窗口，示例代码如下：

```
import javax.swing.JFrame;
public class Swingtest
{
    static final int WIDTH=300;
    static final int HEIGHT=200;
    public static void main(String[] args)
    {
        JFrame jf=new JFrame("helloSwing");
        jf.setSize(WIDTH,HEIGHT);
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jf.setVisible(true);
    }
}
```

上述代码的运行结果如图 1.1 所示。

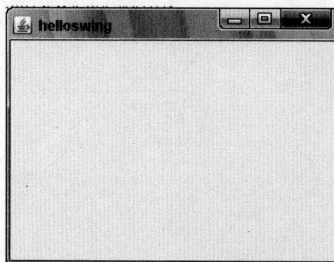


图 1.1 `helloswing` 程序代码的运行结果

为了能够让读者更加清晰地了解此实例，下面将给予详细分析。

- `import javax.swing.JFrame`: 是一个包的导入代码，因为在程序代码中需要使用到这个包中的类以及其中的方法。其实在大多数的程序代码中还需要引入两个 AWT 的包，一个是“`Java.awt.*`”，另一个是“`Java.awt.event.*`”，这些包都是必须的，因为 Swing 组件使用了 AWT 的基本架构，这两个包中包括了 AWT 的事件模型。
- `jf.setSize(WIDTH,HEIGHT)`: 代码的含义是设置窗口的大小。
- `jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)`: 代码的含义是使窗口上的最大化、最小化以及关闭键发挥作用。
- `jf.setVisible(true)`: 代码的含义主要是让前面创建的窗口显示出来。

上面编写的窗口也被称为顶层窗口，编写好顶层窗口后就可以在顶层窗口中添加各种各样的组件，从而实现各种各样的功能。有关窗口方面的知识，本书在后面的章节中会有详细讲解，这里不再赘述。