

高职高专计算机系列规划教材

• JAVA技术与实训系列教程

JAVA

丛书主编 孔 敏

程序设计与实训 (高级)

朱寅非 主 编

朱寅非 田明君 井 辉 编 著



南京大学出版社

高职高专计算机系列规划教材

• JAVA技术与实训系列教程

JAVA

丛书主编 孔 敏

程序设计与实训 (高级)

朱寅非 主 编
朱寅非 田明君 井 辉 编 著



南京大学出版社

内容简介

Java 语言因其面向对象、平台无关性等特征,成为当前通用的网络编程语言。本书主要围绕 Java 语言的高级编程进行讲解,编著者根据丰富的教学经验,在每个知识点的讲解过程中,通过配套的案例和实训来讲解如何使用 Java 语言进行编程。本书一共有 11 个章节,主要包含了 AWT 编程、GUI 的界面元素之间的事件处理、Java Swing 的编程、Applet 小程序、Java 数据结构、Java 文件输入/输出、Java 数据库连接(JDBC)、Java 网络编程、图形与图像、Java 与多媒体以及 Java 多线程等内容。

本书可作为高职高专学生的教材,也可作为其他本科院校计算机软件开发专业学生的自学教材,或作为计算机软件开发人员的参考书。

图书在版编目(CIP)数据

JAVA 程序设计与实训:高级 / 朱寅非主编. — 南京:南京大学出版社, 2013. 12
高职高专计算机系列规划教材
ISBN 978-7-305-12475-4

I. ①J… II. ①朱… III. ①JAVA 语言—程序设计—
高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 275015 号

出版发行 南京大学出版社
社 址 南京市汉口路 22 号 邮 编 210093
网 址 <http://www.NjupCo.com>
出 版 人 左 健
丛 书 名 高职高专计算机系列规划教材
书 名 JAVA 程序设计及实训(高级)
主 编 朱寅非
责任编辑 王秉华 吴 汀 编辑热线 025-83595860
照 排 南京南琳图文制作有限公司
印 刷 南京京新印刷厂
开 本 787×1092 1/16 印张 11.5 字数 322 千
版 次 2013 年 12 月第 1 版 2013 年 12 月第 1 次印刷
ISBN 978-7-305-12475-4
定 价 28.00 元
发行热线 025-83594756 83686452
电子邮箱 Press@NjupCo.com
Sales@NjupCo.com(市场部)

* 版权所有,侵权必究

* 凡购买南大版图书,如有印装质量问题,请与所购
图书销售部门联系调换

前言

Java 语言自 1990 年 12 月诞生以来,已经风靡全世界。因其具有平台无关性、安全性、健壮性、可移植性等特点,到目前为止,已经被公认为是最优秀的编程语言。Java API(Application Programming Interface,应用程序编程接口)是一些预先定义的函数,面向应用程序与开发人员提供调用具有特定功能函数的能力,而又无需访问源码,或理解内部工作机制的细节。高效地使用 Java 意味着不仅仅是对 Java 的语法的了解,而是要发挥 Java 语言面向对象编程之优势,充分利用 Java 预先定义的函数。本书假定你已经在某种程度上使用过 Java,在对 Java 程序设计方法和语言了解的基础上,主要关注 Java 语言核心 API 的使用,内容主要涉及图形编程、Java 数据结构、如何画图、如何播放数字音频、网络编程、流以及多线程等相关内容。读完本书后,读者能够熟练使用常用的 API 进行编程。

本书是孔敏主编的、介绍 Java 语言核心概念的《JAVA 程序设计与实训指导——程序设计方法基础》的进阶篇。作者根据多年的教学经验,在内容的编排上体现了“理论素质培养”和“职业能力训练”的计算机教学思想和方法。通过大量、详尽的实用案例、循序渐进地介绍全书内容,使读者可以很快地理解基本概念和掌握基本应用 API 的编程方法,并能得心应手地编写出合格的程序。本书也注重培养读者思考问题、分析问题和解决问题的能力,是高职高专院校及大专院校的首选教学用书,并适合具有一定 Java 编程经验者进行自我学习。

本书主要针对桌面操作系统环境下,开发独立应用程序类型的 Java 程序。全书的内容编排如下:

第 1 章:介绍图形化用户界面设计的基本知识。在简要介绍 AWT 的概念基础上,对 java.awt 中包含的基本包结构进行了详细的描述,重点对 java.awt 包中定义的图形化用户界面组件对象的定义和应用进行了讲解。

第 2 章:首先介绍了 Java 的事件处理机制及模型,然后重点介绍了 JDK API 中常用的事件对象的定义及应用,同时对事件监听器接口及事件适配器对象进行了详细描述。读者通过本章的学习,真正能体会到图形界面交互操作的实现是因为对某些组件发生的事件进行了处理。

第 3 章:本章选择了 Swing 包中部分常用的组件对象,结合程序代码进行讲解,主要要求读者能掌握 Swing 组件开发图形界面的设计思想和编程方法,做到举一反三的效果。

第 4 章:从介绍 Applet 的工作原理开始,重点介绍了 Applet 类的创建及其生命周期中的 4 个重要方法的调用,随后通过示例介绍网页和 Applet 之间是如何进行参数的传递,使得 Applet 小程序能在 HTML 页面中正常显示。

第 5 章:主要结合相关示例介绍了 Java 中常用的可重用类,包括 Collection、Vector、

Stack、LinkedList、ArrayList 等类。希望读者深刻体会到使用数据结构对象实现数据资源的有效管理和维护。

第 6 章:首先介绍了流的概念及分类,然后引入对类的讲解,接着分别介绍了常见输入输出流的功能及应用,最后介绍了对象克隆与序列化的相关概念及应用。

第 7 章:主要介绍了用 JDBC 访问数据库的相关技术及应用。首先对 JDBC 技术的相关概念及原理等进行了介绍,接着以使用 JDBC 访问数据库的关键步骤为主线,对访问数据库的相关操作进行了详细的讲解,最后通过一个实际的项目案例展现了 JDBC 访问数据库技术的完整代码示例。

第 8 章:主要对 Java 网络编程的相关类进行了介绍,从对网络通信相关协议的讲解开始,接着介绍了使用 URL 类读取网页信息,最后对 Socket 编程进行了详细的讲解。

第 9 章:介绍了 Java 的绘图功能,通过具体实例对 Java 中常见的绘制图形和文字的工具类进行了详细的介绍,最后对使用 Java 绘制二维图形进行了深入的讲解。

第 10 章:主要对 Java 的多媒体音频播放技术进行了介绍,对使用 Applet 播放音频和 Java 程序播放音频两种方式进行了详细的讲解。

第 11 章:从对进程和线程的概念入手,讲解了线程的创建方法,接着介绍了线程的同步和线程锁的基本概念和使用方法,最后对线程常用的方法进行了详细的讲解。

全书共 11 章,建议理论及实训时间安排如下所示,每个学校根据实际情况可以稍作调整。

章 节	理论课时	实训课时
第 1 章	8	4
第 2 章	6	4
第 3 章	4	4
第 4 章	4	4
第 5 章	6	4
第 6 章	4	6
第 7 章	4	6
第 8 章	3	3
第 9 章	2	2
第 10 章	1	1
第 11 章	2	4
合 计	44	42

本书是孔敏任总主编的 Java 系列丛书中的第二本,由朱寅非任主编。朱寅非编写第 1、2、3、4、5 章,田明君编写第 6、7、8、9、10 章,井辉编写第 11 章,谭凯参与了习题的编写及文稿的校验。全书由朱寅非统稿完成。

本书在编写过程中,参考了大量的相关技术资料,吸取了许多同仁的宝贵经验,在此深表谢意。鉴于时间仓促和作者的水平有限,书中难免有不足之处,恳请各位专家和读者批评指正。所有意见和建议请发往:yinfei.zhu@njtvu.edu.cn。

编 者
2013 年 11 月

目 录

第 1 章 AWT 编程	1
1.1 生活场景	1
1.2 AWT 概述	2
1.3 AWT 包结构介绍	2
1.4 容器组件	4
1.4.1 Frame 组件	4
1.4.2 Dialog 组件	5
1.4.3 Panel 组件	7
1.4.4 Applet 组件	8
1.4.5 ScrollPane 滚动面板	9
1.5 标准组件	10
1.5.1 标签和文本组件	11
1.5.2 按钮(Button)组件	15
1.5.3 选项框(Choice)组件	16
1.5.4 复选框和单选按钮(Checkbox)组件	18
1.5.5 列表框(List)组件	20
1.5.6 菜单(Menu)组件	21
1.5.7 画布(Canvas)组件	26
1.6 布局管理器	27
1.6.1 FlowLayout 布局管理器	28
1.6.2 BorderLayout 布局管理器	29
1.6.3 GridLayout 布局管理器	30
1.6.4 CardLayout 布局管理器	31
1.6.5 GridBagLayout 布局管理器	32
1.6.6 Null 布局管理器	34
1.7 本章小结	34
1.8 本章习题	35
1.9 本章实训	35
第 2 章 事件处理	36
2.1 生活场景	36

2.2	Java 事件处理模型概述	37
2.3	事件处理步骤	38
2.4	鼠标事件(MouseEvent)	42
2.5	键盘事件(KeyEvent)	43
2.6	焦点事件(FocusEvent)	45
2.7	事件适配器(Adapter)	46
2.8	剪贴板(Clipboard)	48
2.9	本章小结	53
2.10	本章习题	53
2.11	本章实训	53
第 3 章	Swing 编程	55
3.1	生活场景	55
3.2	Swing 概述	56
3.3	Swing 类层次结构	57
3.4	Swing 多样化组件	58
3.4.1	Swing 中的组件	58
3.4.2	Swing 中的布局管理器	65
3.5	本章小结	67
3.6	本章习题	67
3.7	本章实训	68
第 4 章	Applet 基础	69
4.1	生活场景	69
4.2	Applet 概述	70
4.2.1	Applet 工作原理	70
4.2.2	Applet 类的创建	71
4.2.3	Applet 的生命周期	71
4.2.4	Applet 案例	72
4.3	Applet 的图形绘制	73
4.4	HTML 与 Applet 的参数传递	74
4.5	本章小结	76
4.6	本章习题	76
4.7	本章实训	76
第 5 章	Java 数据结构对象	77
5.1	生活场景	77
5.2	Java 数据结构概述	78
5.3	Collection 接口	78

5.4	向量(Vector)	79
5.5	栈(Stack)	82
5.6	List 接口	84
5.7	链表(LinkedList)	85
5.8	ArrayList	88
5.9	本章小结	89
5.10	本章习题	89
5.11	本章实训	89
第 6 章	输入输出流	90
6.1	生活场景	90
6.2	输入/输出流基础	91
6.2.1	流的概述	91
6.2.2	流的分类	92
6.3	File 类	93
6.3.1	File 类概述	93
6.3.2	File 类构造方法	93
6.3.3	File 类常用方法介绍	93
6.4	文件读写	95
6.4.1	文件输入流(FileInputStream)	95
6.4.2	文件输出流(FileOutputStream)	97
6.5	字符文件读写	98
6.6	文件随机访问	99
6.7	对象克隆与序列化	101
6.7.1	对象克隆	101
6.7.2	序列化与反序列化	102
6.8	本章小结	103
6.9	本章习题	104
6.10	本章实训	104
第 7 章	Java 数据库连接(JDBC)	105
7.1	生活场景	105
7.2	关系数据库与 SQL 语言概述	106
7.3	JDBC 与数据库访问技术	107
7.3.1	JDBC 概述	107
7.3.2	JDBC 的结构	107
7.3.3	JDBC 的驱动程序	108
7.3.4	常用 JDBC API	110
7.4	使用 JDBC 访问数据库	111

7.4.1	准备工作	111
7.4.2	连接数据库	112
7.4.3	查询数据库	112
7.4.4	可滚动结果集	115
7.4.5	更新、添加、删除记录	115
7.4.6	预处理语句	118
7.5	JDBC 应用完整实例	119
7.6	本章小结	124
7.7	本章习题	124
7.8	本章实训	124
第 8 章	Java 网络编程基础	125
8.1	生活场景	125
8.2	网络通信及编程基础	126
8.2.1	TCP 协议	126
8.2.2	IP 地址	127
8.2.3	Java 网络编程概述	127
8.3	URL 类	128
8.3.1	URL 类简介	128
8.3.2	读取 URL 中的资源	129
8.4	套接字(Socket)编程	130
8.4.1	端口和套接字	130
8.4.2	InetAddress 类	132
8.4.3	Socket 类	133
8.4.4	ServerSocket 类	134
8.5	本章小结	136
8.6	本章习题	136
8.7	本章实训	137
第 9 章	图形与图像	138
9.1	生活场景	138
9.2	绘制基本图形	139
9.2.1	图形	139
9.2.2	Color 类	139
9.2.3	绘制基本图形	140
9.3	建立字体	143
9.4	复制与清除图形	145
9.5	Java 2D	146
9.6	本章小结	147

9.7 本章习题	148
9.8 本章实训	148
第 10 章 Java 多媒体	149
10.1 生活场景	149
10.2 在小程序中播放音频	149
10.3 在应用程序中播放音频	151
10.4 本章小结	152
10.5 本章习题	153
10.6 本章实训	153
第 11 章 Java 多线程	154
11.1 生活场景	154
11.2 Java 线程	155
11.2.1 进程与线程	155
11.2.2 线程的生命周期	155
11.3 创建 Java 线程	156
11.3.1 利用 Thread 类创建线程	156
11.3.2 使用 Runnable 接口创建线程	158
11.4 线程同步	159
11.5 线程死锁与解锁	164
11.6 线程优先级别	169
11.7 join 方法的使用	169
11.8 使用 interrupt 方法结束线程	171
11.9 本章小结	172
11.10 本章习题	172
11.11 本章实训	172
参考文献	173

第 1 章

AWT 编程

本章目标

- 了解 AWT 和 GUI 的概述
- 了解 AWT 的包结构
- 熟练掌握 AWT 常见容器组件的创建和应用
- 熟练掌握 AWT 常见标准组件的创建和应用
- 熟练掌握 AWT 常见布局管理器的创建和应用

建议课时

- 理论 8 课时
- 实训 4 课时

1.1 生活场景

小学一年级的数学老师要求学生熟练掌握 1 位数的加法运算。学校希望 Java 软件工程师给他们设计一个图形界面程序。该界面要求能实现如下功能：

- 自动随机产生两个一位数的加法算式；
- 提供一个文本框供学生将答案输入；
- 判断学生做的答案是否正确的按钮，并将结果在文本框里显示；
- 做完一题后，系统提供一个按钮供学生重新做题。

具体的加法练习的初始化主界面如图 1-1 所示。

经过学生输入正确答案后的判断界面如图 1-2 所示。

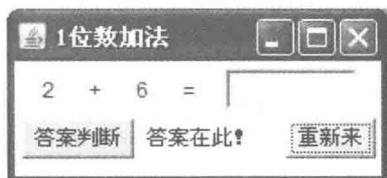


图 1-1 1 位数加法算式图

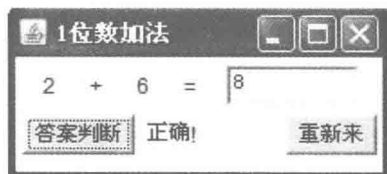


图 1-2 正确答案图

经过学生输入错误答案后的判断界面如图 1-3 所示。

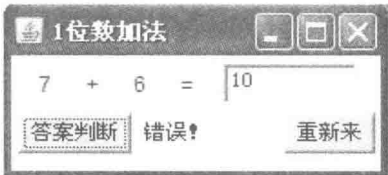


图 1-3 错误答案图

1.2 AWT 概述

在开发 Applet 和 Java 图形化用户界面(GUI)的应用程序时,一般需要用到 Abstract Window Toolkit(AWT)抽象窗口工具集所提供的基本元素。AWT 是 Java 开发工具包(JDK)的一部分,主要包括按钮、列表、菜单、文本域等基本的界面组件,这些组件主要用来建立图形用户界面的独立平台。除此之外,AWT 还提供了事件处理接口、图形图像工具以及布局管理等,同时还支持剪贴板、数据传输和图像操作。随着 2D API 的出现,AWT 还提供包括高级字体操作、打印、地理数据获取和输入方法等功能的软件包。AWT 的初始版本是基于在简单用户界面中开发 Applet 小程序而设计的,与之相比,当前的 AWT 做了很大的改进,它提供事件模型重新设计、剪贴板和数据传输支持以及打印和无鼠标操作等功能。

AWT 是 Java 基础类库(JFC,Java Foundation Class)的一部分,它为 GUI 的开发提供了丰富的基础结构。AWT 支持 JavaBeans 体系结构,它的每一个组件都是一个 JavaBean。

1.3 AWT 包结构介绍

与 AWT 相关的软件包有很多,表 1-1 对主要的软件包进行了详细的介绍。

表 1-1 AWT 主要软件包

包名	描述
java. awt	AWT 核心包,提供组件类、事件类等
java. awt. event	提供事件类和监听器
java. awt. color	定义颜色及其空间
java. awt. font	提供各种字体
java. awt. datatransfer	提供数据传输和剪贴板功能
java. awt. dnd	图形化用户之间的拖拽功能
java. awt. geom	几何软件包
java. awt. im	引入方法
java. awt. image	图像处理
java. awt. peer	同位体组件界面
java. awt. print	提供打印功能
java. awt. accessibility	提供辅助功能

java.awt 中的各个类如图 1-4 所示。

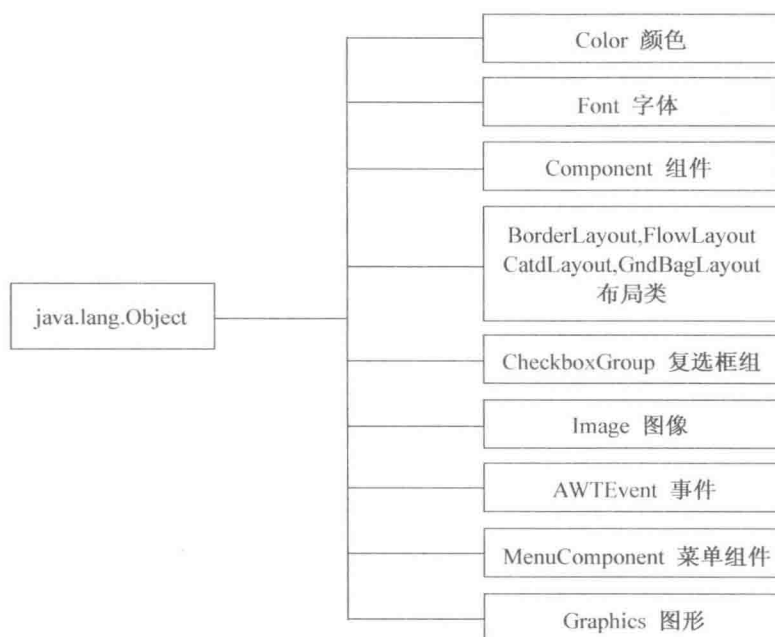


图 1-4 java.awt 中的各个类

java.awt 包中的常用图形组件类的继承关系如图 1-5 所示。

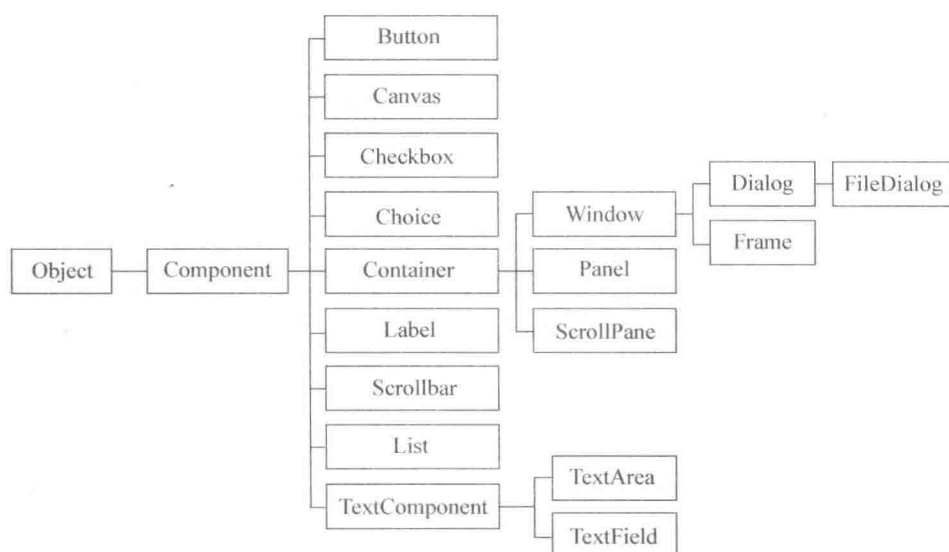


图 1-5 java.awt 包中的常用图形组件类的继承关系

要设计一个图形化用户界面,主要包含容器(Container)的设计、布局管理器的设计(LayoutManager)以及标准组件的设计(Component)。这三者的关系就如将一套家具放到一套房子中去,然后按照你喜欢的样式进行摆放家具。容器就是例子中的房子,标准组件就是例子中的家具,布局管理器就是家具摆放的样式。要设计出一个看起来舒适美观的图形

化用户界面的关键,就是要处理好容器、布局管理器以及标准组件的构建方式。下面我们从容器、布局管理器和标准组件三个方面分别阐述。

1.4 容器组件

要设计一个图形化用户界面的应用系统,必须需要创建容器组件(Container)。容器是一种特殊组件,它既能容纳其他的组件,同时也是用户操作的界面。AWT 中常用的主要容器有以下 4 类:

- Frame
- Panel
- Dialog
- Applet

Java 中主要的容器有两类:窗口(Window)和面板(Panel)。Window 容器主要包括框架(Frame)和对话框(Dialog)两种容器类组件;面板(Panel)容器的子类 Applet 主要应用在浏览器中,它是一个能够嵌入浏览器运行的特殊容器。这 4 种容器的关系如图 1-6 所示。

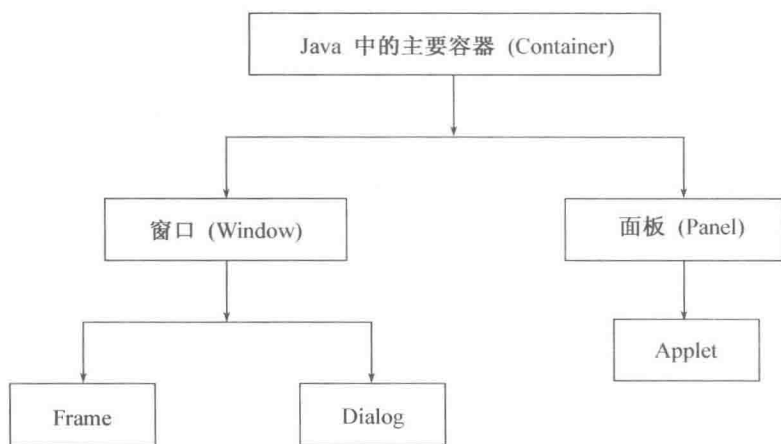


图 1-6 AWT 中的主要容器关系图

1.4.1 Frame 组件

框架类 Frame 是 Window 类的子类。它描述的是一种带标题框、并可以改变大小的顶层应用程序窗口。Frame 组件的对象是一个 Frame 窗口。默认状态下不可视,且呈最小化状态,一般需要使用该类中的 setSize()或 setBounds()方法进行重新设定窗口的大小,并调用 setVisible(true)来使窗口可见。下面我们来介绍一下 Frame 容器组件的构造方法和其他常用的相关方法,其中部分方法直接来自 Frame 类,部分方法继承于它的直接父类或间接父类。

- 常用构造方法

```
public Frame()
```

```
public Frame(String title)
```

其中参数 title 指定框架的标题;没有参数时,则框架无标题。

■ 其他常用方法

```
public void setSize(int width,int height) //设置组件的宽度和高度
```

```
public void setVisible(boolean b) //设置组件是否可见
```

```
public void setTitle(String title) //设置框架标题
```

```
public void getTitle() //获取框架标题
```

```
public void setBounds(int x, int y, int width, int height) //重新设置组件的大小并
//移动组件到指定的 x、y
//坐标的位置上。
```

```
public void setLayout(LayoutManager mgr) //设定布局管理器
```

```
public Component add(Component comp) //在容器中添加一个组件 comp
```

```
public void setBackground(Color c) //设置框架的背景色
```

```
public void setLocation(int x, int y) //设置组件位置
```

■ Frame 框架举例

本例创建一个 Frame 对象作为窗口,并设置了该框架的标题、大小及背景色。其中 Color. red 是 java. awt. Color 中的一个常量,表示红色。程序如下:

例 1-1:

```
001:import java. awt. * ;
002:public class FrameDemo{
003:    public static void main(String args[]) {
004:        Frame f=new Frame("创建 frame 窗口");
005:        f. setSize(300,100);
006:        f. setBackground(Color. red);
007:        f. setVisible(true);
008:    }
009:}
```

运行结果如图 1-7 所示。



图 1-7 Frame 窗口

1.4.2 Dialog 组件

Dialog 描述的是对话窗口,主要用来接收用户输入的信息。当对话窗口很小时,被称为

对话框。对话窗口一般对应于一个顶层窗口,如 Frame 窗口。下面我们来介绍一下 Dialog 容器组件的构造方法和其他常用的相关方法。

■ 常用构造方法

```
public Dialog(Dialog owner)
public Dialog(Dialog owner, String title)
public Dialog(Dialog owner, String title, boolean modal)
public Dialog(Frame owner)
public Dialog(Frame owner, boolean modal)
public Dialog(Frame owner, String title)
public Dialog(Frame owner, String title, boolean modal)
```

其中:owner 表示自定义对话框的所有者,它可以是一个 Dialog 或一个 Frame 对象;title 是对话框的标题;布尔类型的变量 modal 取值为 true 时,则为模式对话框,否则为无模式对话框。

所谓模式对话框,是指对话框打开后必须对该对话框作出响应,程序才能继续进行。

■ 其他常用方法

```
public void setTitle(String title) //设置对话框标题
public void getTitle() //获取对话框标题
public void show() //设置对话框可见
public void hide() //隐藏对话框
public boolean isModal() //判断该对话框是否是模式对话框
public void setModal(boolean modal) //设置该对话框为模式对话框或无模式对话框
```

■ Dialog 对话框举例

本例主要完成以下任务:

(1) 创建一个 Frame 对象作为窗口。

(2) 在该窗口中添加一个“打开对话框”按钮,当点击该按钮,就跳出一个 Dialog 对话框。在该对话框中添加了一个“你好”标签和一个“确定”按钮。程序如下:

例 1-2:

```
001:import java.awt.*;
002:import java.awt.event.*;
003:public class DialogDemo implements ActionListener{
004:    Frame f=new Frame("对话框示例");
005:    Button b=new Button("打开对话框");
006:    DialogDemo(){
007:        b.addActionListener(this);
008:        f.add(b);
009:        f.setSize(300,100);
010:        f.setVisible(true);
011:    }
012:    public void actionPerformed(ActionEvent e){
013:        Dialog dlg=new Dialog(f,"你好",true);
```

```

014:        dlg.setLayout(new FlowLayout());
015:        dlg.add(new Label("你好"));
016:        dlg.add(new Button("确定"));
017:        dlg.setSize(100,60);
018:        dlg.show();
019:    }
020:    public static void main(String[] args){
021:        new DialogDemo();
022:    }
023:}

```

运行结果如图 1-8 所示。

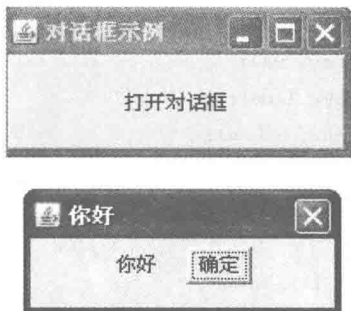


图 1-8 Dialog 窗口

1.4.3 Panel 组件

Panel 面板类是 Container 类的子类。它是一种没有标题的中间容器,实例化后必须使用 Container 类中的 add 方法装入到 Window 对象中,比如一个顶层容器或者存在于顶层容器中的一个中间容器。Panel 类常被用于图形界面的修饰。下面我们来介绍一下 Panel 容器组件的构造方法和其他常用的相关方法。

■ 常用构造方法

```
public Panel()
```

```
public Panel(LayoutManager layout)
```

其中,参数 layout 为指定的布局管理器;没有参数时,表示使用默认的布局管理器。

■ 其他常用方法

Panel 面板类的主要方法都是从 Container 和 Component 类继承过来的,具体请参见 JDK API 帮助文档,在此不再赘述。

■ Panel 面板举例

本例主要完成以下任务:

(1) 创建一个 Frame 对象作为窗口;

(2) 创建 3 个 Panel 面板并加入到 Frame 框架中。程序如下: