

普通高等院校计算机专业(本科)实用教程系列

Java 语言最新实用案例教程

杨树林 胡洁萍 编著

清华大学出版社
北 京

普通高等院校计算机专业(本科)实用教程系列
编 委 会

主 任 孙家广(清华大学教授,中国工程院院士)

成 员 (按姓氏笔画为序)

王玉龙(北方工业大学教授)

艾德才(天津大学教授)

刘 云(北方交通大学教授)

任爱华(北京航空航天大学教授)

幸云辉(北京邮电大学教授)

张海藩(北京信息工程学院教授)

徐孝凯(中央广播电视大学教授)

徐培忠(清华大学出版社编审)

樊孝忠(北京理工大学教授)

丛书策划 徐培忠 徐孝凯

序 言

时光更迭、历史嬗递。中国经济以她足以令世人惊叹的持续高速发展驶入了一个新的世纪,一个新的千年。世纪之初,以微电子、计算机、软件和通信技术为主导的信息技术革命给我们生存的社会所带来的变化令人目不暇接。软件是优化我国产业结构、加速传统产业改造和用信息化带动工业化的基础产业,是体现国家竞争力的战略性产业,是从事知识的提炼、总结、深化和应用的高智型产业;软件关系到国家的安全,是保证我国政治独立、文化不受侵蚀的重要因素;软件也是促进其他学科发展和提升的基础学科;软件作为 20 世纪人类文明进步的最伟大成果之一,代表了先进文化的前进方向。美国政府早在 1992 年“国家关键技术”一文中提出“美国在软件开发和应用上所处的传统领先地位是信息技术及其他重要领域竞争能力的一个关键因素”,“一个成熟的软件制造业的发展是满足商业与国防对复杂程序日益增长的要求所必需的”,“在很多国家关键技术中,软件是关键的、起推动作用(或阻碍作用)的因素”。在 1999 年 1 月美国总统信息技术顾问委员会的报告“21 世纪的信息技术”中指出“从台式计算机、电话系统到股市,我们的经济与社会越来越依赖于软件”,“软件研究为基础研究方面最优先发展的领域。”而软件人才的缺乏和激烈竞争是当前国际的共性问题。各国、各企业都对培养、引进软件人才采取了特殊政策与措施。

为了满足社会对软件人才的需要,为了让更多的人可以更快地学到实用的软件理论、技术与方法,我们编著了《普通高等院校计算机专业(本科)实用教程系列》。本套丛书面向普通高等院校学生,以培养面向 21 世纪计算机专业应用人才(以软件工程师为主)为目标,以简明实用、便于自学、反映计算机技术最新发展和应用为特色,具体归纳为以 点:

1. 进透基本理论、基本原理、方法和技术,在写法上力求叙述详细,算法具体,通俗易懂,便于自学。

2. 理论结合实际。计算机是一门实践性很强的科学,丛书贯彻从实践中来到实践中去的原则,许多技术理论结合实例讲解,以便于学习理解。

3. 本丛书形成完整的体系,每本教材既有相对独立性,又有相互衔接和呼应,为总的培养目标服务。

4. 每本教材都配以习题和实验,在各教学阶段安排课程设计或大作业,培养学生的实战能力与创新精神。习题和实验可以制作成光盘。

为了适应计算机科学技术的发展,本系列教材将本着与时俱进的精神不断修订更新,及时推出第二版、第三版……完整PDF请访问: www.ertongbook.com

新世纪曙光激人向上,催人奋进。江泽民总书记在十五届五中全会上的讲话:“大力推进国民经济和社会信息化,是覆盖现代化建设全局的战略举措。以信息化带动工业化,发挥后发优势,实现社会生产力的跨越式发展”,指明了我国信息界前进的方向。21 世纪日趋开放的国策与更加迅速发展的科技会托起祖国更加辉煌灿烂的明天。

孙家广

2004 年 1 月

前 言

Java 语言具有面向对象、与平台无关、安全、稳定和多线程等特点。不仅可以用来开发大型的应用程序,而且特别适合于开发网络应用程序。目前无论是高校的计算机专业还是 IT 培训学校都将 Java 作为主要的教学内容之一,这对于培养学生的计算机应用能力具有重要的意义。实践表明,这门课的教学存在一定的问题,主要表现在:学生理解抽象的程序设计语言较困难;学生的实践不充分,缺乏有效的指导,知识学习与应用能力培养相脱节;教学内容缺乏实用性和新颖性。

案例教学是计算机语言教学的最有效的方法之一。好的案例对学生理解知识,掌握如何应用知识十分重要。目前一些教材类书籍,例子缺乏实用性和新颖性,对学生的技术指导不够,而一些技术性较强的参考书,又过分强调技术,知识讲解不够系统,且缺乏有效联系,不适合于教学。为此,本书精心设计了与教学目的结合紧密,适合于学生学习和教师教学的案例,将知识讲解融入到案例之中,并能很好地指导学生实践,这对于学生从实践中理解和巩固知识,在实践中培养应用能力,具有重要的实际意义。

本书不同于普通技术参考书,它以指导案例教学为目的,围绕教学内容组织案例,对学生的知识和能力训练具有很强的针对性,主要特点是:

1. 以知识线索设计案例,分解知识点,有明确的目的和要求,针对性强。
2. 选择有代表性的实例,突出重点知识的掌握和应用。
3. 将技术指导、代码与注释、应用提高、相关知识有机结合起来。
4. 注意新方法、新技术的引用。
5. 处理好具体实例与思想方法的关系,局部知识应用与综合应用的关系。
6. 强调实用性,培养应用能力。

本书共分 8 章,具体包括如下内容:

第 1 章通过 14 个案例讲述 Java 语言编程的基础知识。

第 2 章通过 13 个案例讲述 Java 图形用户界面程序设计的有关知识,重点是如何使用 Swing 技术进行图形界面的开发。

第 3 章通过 9 个案例讲述 Java 图形、图像与多媒体程序设计的技术和方法。

第 4 章通过 11 个案例讲述 Java 数据库编程的有关技术和知识。

第 5 章通过 7 个案例讲述 Java 流和文件编程的基本技术和知识。

第 6 章通过 13 个案例讲述 Java 多线程及 Applet 程序设计的技术和知识。

第 7 章通过 11 个案例讲述 Java 网络编程的基本技术和知识。

第 8 章通过 10 个案例讲述 Java Servlet 及 JSP 的有关技术和知识。

本书由杨树林、胡洁萍编写,在编写过程中得到了北京印刷学院和北京市委组织部的大力支持,在此表示感谢。

由于时间仓促,作者水平有限,书中难免存在疏漏和不足,恳请读者批评指正,使本书得以改进和完善。

作 者

2005 年 8 月于北京

目 录

第 1 章 Java 语言基础	1
1.1 初识 Java 程序	1
案例 1-1 一个简单的 Java 应用程序	1
案例 1-2 一个简单的 GUI 程序	3
1.2 Java 语言基础	5
案例 1-3 两个数的计算程序	6
案例 1-4 求素数	10
1.3 数组与字符串	12
案例 1-5 数组排序与求和	12
案例 1-6 字符串类使用演示	16
1.4 类和对象	19
案例 1-7 学生类 Student	19
案例 1-8 用内部类实现两个监听者	23
1.5 继承和接口	26
案例 1-9 图形类	26
案例 1-10 栈	30
1.6 向量与哈希表	33
案例 1-11 向量的使用	33
案例 1-12 用户权限哈希表	35
1.7 异常处理	38
案例 1-13 温度转换	38
案例 1-14 求三角形面积	41
第 2 章 GUI 程序设计	44
2.1 窗口与 Swing 初步	44
案例 2-1 启动界面	44
案例 2-2 设计个性化的窗体	48
2.2 对话框	51
案例 2-3 登录窗口	51
案例 2-4 JOptionPane 对话框	55
2.3 菜单与工具栏的使用	60
案例 2-5 为窗体添加菜单	60
案例 2-6 工具栏	64
2.4 布局的方法	67

案例 2-7 基本布局演示	67
案例 2-8 使用网格袋布局	71
2.5 组件的使用	73
案例 2-9 学生档案输入界面	73
案例 2-10 使用列表	79
2.6 树和表格组件	83
案例 2-11 树	83
案例 2-12 表格	89
2.7 父子窗体的实现	92
案例 2-13 父子窗体	92
第 3 章 图形、图像与多媒体	96
3.1 绘图方法	96
案例 3-1 基本的绘图方法演示	96
案例 3-2 2D 绘图演示	100
3.2 绘图应用程序	105
案例 3-3 时钟	105
案例 3-4 绘图程序	109
3.3 图像显示与跟踪	116
案例 3-5 在 Applet 中显示图像	116
案例 3-6 图像浏览程序	118
3.4 图像缓冲技术	121
案例 3-7 用鼠标拖动图像	121
3.5 声音与视频的播放	124
案例 3-8 播放声音	124
案例 3-9 播放视频	126
第 4 章 Java 数据库编程	130
4.1 数据库的连接	131
案例 4-1 使用 JDBC-ODBC 桥实现数据库的连接	131
案例 4-2 使用纯 Java JDBC 驱动程序实现数据库的连接	133
4.2 数据的基本操作	135
案例 4-3 读取数据	135
案例 4-4 数据更新	139
4.3 数据库的元数据查询	146
案例 4-5 获取数据库基本信息	146
4.4 使用表格和树组件浏览数据	149
案例 4-6 使用表格浏览数据	149
案例 4-7 用树浏览数据	153

4.5	预处理与存储过程	158
	案例 4-8 使用语句预处理	159
	案例 4-9 调用存储过程	162
4.6	图形数据的读写	166
	案例 4-10 读写照片数据	166
4.7	事务处理	171
	案例 4-11 一个简单的事务处理程序	172
第 5 章	流与文件	176
5.1	流与文件的基本操作	176
	案例 5-1 获取文件属性	176
	案例 5-2 使用字符流复制文件	180
	案例 5-3 用字节流复制文件	184
5.2	随机读写文件	188
	案例 5-4 一个简单的随机文件读写程序	188
5.3	以对象为单位读写数据	190
	案例 5-5 读写学生对象数据	191
5.4	文件的压缩和解压缩	193
	案例 5-6 一个简单的压缩和解压程序	193
5.5	文件选择对话框	198
	案例 5-7 使用文件对话框打开和保存文件	198
第 6 章	多线程与 Applet	204
6.1	线程及其创建方法	204
	案例 6-1 文字时钟	204
	案例 6-2 随机画图	208
6.2	线程的基本控制	212
	案例 6-3 图片切换	212
	案例 6-4 数数程序	217
6.3	线程同步	219
	案例 6-5 打电话	219
	案例 6-6 一个死锁程序	221
6.4	线程间通信	223
	案例 6-7 哲学家用餐问题	223
	案例 6-8 吃苹果	225
6.5	使用管道进行线程通信	228
	案例 6-9 一个简单的线程间通信程序	229
6.6	Applet 程序设计	231
	案例 6-10 在 Applet 中绘制统计图	231

案例 6-11 图像动画	236
6.7 同页 Applet 间的通信	239
案例 6-12 一个简单的 Applet 通信程序	239
6.8 在 Applet 中应用 JAR 包	241
案例 6-13 在 Applet 中使用打包的图形按钮	241
第 7 章 网络程序设计	245
7.1 获取网站信息	245
案例 7-1 通过 InetAddress 获取网络域名或 IP	245
案例 7-2 获取网络属性信息	247
7.2 获取网络资源	249
案例 7-3 读取网络文本文件	249
案例 7-4 简单的浏览器	252
7.3 基于 TCP/IP 的服务器与客户机通信	254
案例 7-5 TCP 的客户端程序	254
案例 7-6 TCP 服务器端程序	259
7.4 基于数据报的主机通信	262
案例 7-7 一个基于 UDP 的聊天程序	262
7.5 收发电子邮件	265
案例 7-8 发送电子邮件	265
案例 7-9 接收电子邮件	269
7.6 远程方法调用	274
案例 7-10 一个简单的远程方法调用程序	274
7.7 FTP 网络文件传输	277
案例 7-11 FTP 客户端的实现	277
第 8 章 Servlet 与 JSP 技术	283
8.1 Servlet 技术	283
案例 8-1 一个简单的 Servlet	283
案例 8-2 一个小测试程序	287
8.2 JSP 程序设计基础	291
案例 8-3 一个简单的 JSP	291
案例 8-4 用 JSP 处理表单	296
8.3 在 JSP 中使用 JavaBean	301
案例 8-5 猜字游戏	301
8.4 通过 JSP 访问数据库	304
案例 8-6 一个注册和登录程序	304
案例 8-7 使用 JavaBean 查询数据	308
8.5 在 JSP 中使用标签	313

案例 8-8 简单标签	313
案例 8-9 带有主体和属性的标签	318
8.6 Struts 应用框架	323
案例 8-10 用 Struts 模式设计 Web 程序	323
参考文献	331

第 1 章 Java 语言基础

Java 语言是 Sun 公司推出的新一代面向对象程序设计语言,具有简单、稳定、与平台无关、安全、解释执行、多线程等特点,是目前使用最为广泛的网络编程语言。

本章主要介绍如下内容:

- 初识 Java 程序
- Java 语言基础
- 数组与字符串
- 类和对象
- 继承和接口
- 向量和哈希表
- 异常处理

1.1 初识 Java 程序

Java 程序分为 Java 应用程序(Java Application)和 Applet 小程序(Java Applet)两类。Java 应用程序是一种能在支持 Java 的平台上,通过解释器(Java.exe)独立运行的程序;Applet 小程序则是嵌入在 HTML 编写的 Web 页面中,由 Web 浏览器内含的 Java 解释器解释运行的非独立程序。

案例 1-1 一个简单的 Java 应用程序

【案例说明】

程序运行后显示“欢迎进入 Java 世界!”,运行结果如图 1-1 所示。

【技术要点】

Java 应用程序包含 main 方法,它是 Java 应用程序执行的起点。在该函数中通过系统对象 System.out 的方法 println(),可以在屏幕上输出字符串。

【代码与注释】

```
public class WelcomeJava{
    public static void main(String args[]) {
        System.out.println("欢迎进入 Java 世界!");
    }
}
```

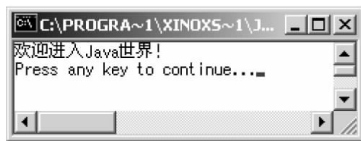


图 1-1 欢迎进入 Java 世界

Java 程序的基本单位是类。class 为定义类的关键字,类的名称为 WelcomeJava。关键字 public 表示这个类的访问属性是公共的。

一个源文件中可以声明多个类,但仅允许有一个公共的类。对于包含多个类的应用程序,应把包含 main()方法的类声明为 public 类,该类为主类,其他类不能用 public 修饰。而且,文件名应与主类的名称完全一致。一个类可以包含多个成员方法,其中 main()方法是一个特殊的方法,它是程序执行的入口。

【应用扩展】

用 Applet 小程序显示“欢迎进入 Java 世界!”,运行界面如图 1-2 所示。



图 1-2 Applet 小程序

```
import java.awt. * ;           //导入 java. awt 包中的类
import java. applet. * ;       //导入 java. applet 包中的类
public class WelcomeJavaApplet extends Applet {   //继承 Applet
    public void paint(Graphics g) {                //重写 paint 方法
        g. drawString("欢迎进入 Java 世界 !",50,60 ); //在(50,60)位置输出字符串
    }
}
```

嵌入字节码文件的 HTML 文件:

```
<HTML>
<HEAD></HEAD>
<BODY BGCOLOR="000000">
<APPLET code = "WelcomeJavaApplet. class" width="500" height="300" >
</APPLET>
</BODY>
</HTML>
```

Applet 程序必须有且仅有一个类是 Applet 的子类。extends 关键字说明 WelcomeJavaApplet 是 Applet 的子类。该类用 public 修饰,是主类。

Applet 程序中不需要有 main()方法。paint()方法是系统类 Applet 中已经定义好的成员方法。它在 WWW 所显示的 Web 页面需要重画时(例如浏览器窗口在屏幕上移动或放大、缩小等)被浏览器自动调用并执行。使用 Graphics 对象 g 的一个成员方法 drawString(),可在指定的位置显示字符串。

【相关知识】

1. 关于 Java

Java 是 1995 年由美国 Sun Microsystems 公司正式推出的完全面向对象的编程语言。它充分吸取了 C++ 语言的优点,采用了程序员所熟悉的 C 和 C++ 语言的许多语法,同时又去掉了 C 语言中指针、内存申请和释放等影响程序健壮性的部分。Java 语言的一个目标是

跨平台,因此采用了解释执行而不是编译执行的运行环境,在执行过程中根据所在的不同的硬件平台把程序解释为当前的机器码,实现跨平台运行。与 Java 相关的还有以下几点:

- Java 语言是当今最流行的一种面向对象的网络编程语言。
- Java 是 SUN 公司由 OAK 项目发展而来的,是消费电器设备方面的一个研究成果。
- Java 是在 C++ 的基础上改进而来的,语法与 C++ 比较类似,但远比 C++ 易学易用。
- Java 与平台无关,非常适合设计与互联网有关的软件。

2. Java 编程环境

JDK 是整个 Java 的核心,包括了 Java 运行环境(Java Runtime Environment),许多 Java 工具和 Java 基础的类库。

通过 Sun 公司的官方网站可以下载 JDK。如: <http://java.sun.com/j2se/1.4.2/download.html>。下载后,运行 `j2sdk-1_4_2_06-windows-i586-p.exe` 进行安装。

JDK 主要包括以下工具。

- `javac`: Java 语言编译器,输出结果为 Java 字节码。
- `java`: Java 字节码解释器。
- `javadoc`: 类帮助文档生成器。
- `jar`: 打包工具。
- `javap`: Java 字节码分解程序,本程序返回 Java 程序的成员变量及方法等信息。
- `javaprof`: 资源分析工具。
- `java Applet Viewer`: 小应用程序浏览工具,用于测试并运行 Applet 小程序。
- `java Debugger API`: Java 调试工具。

Java 程序的编辑可以使用任何一种文本编辑器,例如, UltraEdit、Notepad、Wordpad 甚至 Word,然后只要把编辑好的文件存成 .java 文件。当然也可以用一些集成开发环境。目前比较流行的集成开发环境主要有 JBuilder 和 JCreator。本书的案例都是利用 JCreator 2.5 编写和调试的,JDK 用的是 JDK 1.4 版。

当引用到某些自己定义的类时,必须指明它们的存放目录,这就需要利用环境变量参数 CLASSPATH。在 Windows 2000 中,右键单击桌面上“我的电脑”,选择“属性”,在弹出的“系统特性”窗口中,选择“高级”选项卡,然后单击“环境变量”按钮,在“环境变量”窗口中编辑 CLASSPATH 和 PATH。

案例 1-2 一个简单的 GUI 程序

【案例说明】

本案例在界面中加入了一个按钮和一个文本框,单击按钮可在文本框中显示“你好! ”。运行结果如图 1-3 所示。

【技术要点】

图形用户界面程序(Graphics User Interface)简称 GUI。浏览器本身是图形界面环境,所以 Applet 程序只能在图形界面下运行。一个 GUI 要导入图形类包 `java.awt` 中的类,若使用事件,还需要导入事件类包 `java.awt.event` 中的类。本程序用 `TextField` 定义文本框,用 `Button` 定义按钮,用



图 1-3 一个简单的 GUI 程序

动作事件处理单击按钮事件。

【代码与注释】

```
import java.awt.*;           //导入图形类
import java.applet.*;       //导入 Applet 类
import java.awt.event.*;    //导入事件类
public class Hello extends Applet implements ActionListener{ //动作事件接口
    TextField output=new TextField(10);           //定义并建立单行文本框,10 为可容纳的字符数
    Button ok=new Button("确定");                //定义并建立按钮
    public void init() {                          //初始化方法
        add(ok);                                  //添加文本框
        add(output);                              //添加按钮
        ok.addActionListener(this);              //注册事件监听者为当前对象
    }
    public void actionPerformed(ActionEvent e){   //实现行为事件方法
        output.setText("你好!");                 //设置文本框的字符串
    }
}
```

在类的说明部分,implements ActionListener 表明这个类实现了动作事件监听者接口,该类可作为动作事件的监听者。实现动作事件接口的类,必须实现接口所规定的方法 actionPerformed()。

【应用扩展】

可以建立一个具有同样功能的窗口程序。窗口程序需要继承 Frame 类,同时要包含 main() 方法。

【相关知识】

1. 运行 Java 程序

Java 应用程序执行的过程是,先利用 Java 编译器(Javac.exe)将 Java 源文件(.java)编译成字节码文件(.class),再使用 Java 解释器(java.exe)解释并执行字节码文件。

与应用程序不同,Applet 程序是由 Web 浏览器内嵌的 Java 解释器解释执行。

2. Java 程序的特点

(1) 简单

Java 语言的简单性主要体现在以下 3 个方面:

- Java 的风格类似于 C++,因而 C++ 程序员是非常熟悉的。从某种意义上讲,Java 语言是 C 语言及 C++ 语言的一个变种,因此,C++ 程序员可以很快就掌握 Java 编程技术。
- Java 摒弃了 C++ 中容易引发程序错误的地方,如指针和内存管理。
- Java 提供了丰富的类库。

(2) 面向对象

面向对象可以说是 Java 最重要的特性。Java 语言是完全面向对象的,它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码继承及重用。

(3) 分布式

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此,Java 应用程序

可凭借 URL 打开并访问网络上的对象,其访问方式与访问本地文件系统几乎完全相同。

(4) 健壮

Java 致力于检查程序在编译和运行时的错误。类型检查可帮助检查出许多开发早期出现的错误。Java 自己操纵内存,减少了内存出错的可能性。Java 还实现了真数组,避免了覆盖数据的可能。这个功能特征大大缩短了开发 Java 应用程序的周期。Java 提供了 null 指针检测、数组边界检测和异常出口字节代码校验。

(5) 结构中立

Java 将它的程序编译成一种结构中立的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码。Java 源程序被编译成一种高层次的与机器无关的字节码格式语言,这种语言被设计在虚拟机上运行,由与机器相关的运行调试器实现执行。

(6) 安全

Java 的安全性可从两个方面得到保证。一方面,在 Java 语言里,像指针和释放内存等 C++ 功能被删除,避免了非法内存操作。另一方面,当 Java 用来创建浏览器时,语言功能和一些浏览器本身提供的功能结合起来,使它更安全。Java 语言在机器上执行前,要经过很多次的测试。它经过代码校验,检查代码段的格式,检测指针操作、对象操作是否过分以及试图改变一个对象的类型。

(7) 可移植性

与体系结构无关的特性使得 Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行,这成为 Java 应用软件便于移植的良好基础。通过定义独立于平台的基本数据类型及其运算,Java 数据得以在任何硬件平台上保持一致。

(8) 解释性

Java 解释器(运行系统)能直接运行目标代码指令。链接程序通常比编译程序所需资源少,所以程序员可以在创建源程序上有更多的时间。

(9) 高性能

如果解释器速度不慢,Java 可以在运行时直接将目标代码翻译成机器指令。如 Sun 用直接解释器一秒钟内可调用 300 000 个过程。翻译目标代码的速度与 C/C++ 没什么区别。

(10) 多线程

Java 提供的多线程功能使得在一个程序里可同时执行多个小任务。Java 比 C 和 C++ 更健壮。多线程带来的更大的好处是更好的交互性能和实时控制性能。

(11) 动态

Java 的动态特性是其面向对象设计方法的扩展。它允许程序动态地装入运行过程中所需要的类,这是 C++ 语言进行面向对象程序设计所无法实现的。

1.2 Java 语言基础

Java 语言包括的基本数据类型、变量、常量、表达式和控制流语句是掌握 Java 编程的必要条件。

案例 1-3 两个数的计算程序

【案例说明】

实现一个窗口程序,用文本框输入两个操作数和一个算术运算符,单击按钮后将计算结果显示在第四个文本框中。运行界面如图 1-4 所示。

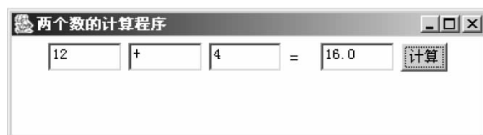


图 1-4 计算程序的界面

【技术要点】

通过继承 Frame(框架)类建立窗口,用 TextField(文本框)输入和显示数据,在动作事件中进行计算。

【代码与注释】

```
import java. awt. * ;
import java. awt. event. * ;
class Calculate extends Frame implements ActionListener{
    TextField t1=new TextField(5);           //第一个操作数文本框
    TextField t2=new TextField(5);           //运算符文本框
    TextField t3=new TextField(5);           //第二个操作数文本框
    TextField t4=new TextField(5);           //结果文本框
    Label L1=new Label("=");
    Button btn=new Button("计算");
    public Calculate() {
        setLayout(new FlowLayout());
        add(t1); add(t2); add(t3); add(L1); add(t4); add(btn);
        btn. addActionListener(this);        //注册动作事件监听者为当前对象
        addWindowListener(new WindowAdapter() { //关闭窗口事件
            public void windowClosing(WindowEvent e) {
                dispose();                    //释放窗口
                System. exit(0);              //退出程序
            }
        });
    }
    public void actionPerformed(ActionEvent e){
        float x,y;                            //操作数变量
        double result=0;                       //结果变量
        String op;
        try{                                    //异常捕获机制
            x=Float. parseFloat(t1. getText()); //将字符串数据转换成浮点型数据
            y=Float. parseFloat(t3. getText());
```

```
op=t2.getText();
if(op.equals("+"))result=x+y;    //运算符为"+"
else if(op.equals("-"))result=x-y; //运算符为"-
else if(op.equals("*"))result=x*y; //运算符为"*
else if(op.equals("/"))result=x/y; //运算符为"/
t4.setText(Double.toString(result));
} catch(Exception ee){t4.setText("数据错误");} //捕获异常,数据错误时,显示信息
}
public static void main(String args[]) {
    Calculate mainFrame = new Calculate();
    mainFrame.setSize(400,400);
    mainFrame.setTitle("两个数的计算程序");
    mainFrame.setVisible(true);
}
}
```

【应用扩展】

当第二个数据输入完后直接按回车键,就进行计算。可以捕获文本框 t3 的动作事件,只需在构造方法中增加如下语句:

```
t3.addActionListener(this);
```

【相关知识】

1. Java 的注释

Java 有 3 种类型的注释语句:

```
//注释一行
/* 一行或多行注释 */
/** * 文档注释 * */
```

文档注释一般放在一个类或方法定义前,在自动生成文档时,可以被提出。Java 自动生成文档的工具是 javadoc.exe。

2. 标识符和保留字

程序员对程序中的各个元素加以命名时使用的命名记号称为标识符。Java 语言中,标识符是以字母、下划线()、美元符(\$)开始的一个字符序列,后面可以跟字母、下划线、美元符、数字。例如,identifier,userName,User_Name,__sys_val,\$change 为合法的标识符,而 2mail,room#,class 为非法的标识符。

具有专门的意义和用途,不能当作一般的标识符使用的标识符称为保留字,也称为关键字,如 abstract,continue,for,new,switch ,boolean 和 default 等。

3. 数据类型

Java 语言的数据类型有简单类型和复合类型。

简单数据类型包括下面几种。

- 整数类型: byte,short,int,long
- 实数类型: float,double
- 字符类型: char