计算机实用技术与案例丛书

Java 编程实用技术与案例

杨绍方　编著

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书在介绍 Java 编程的相关知识的基础上，以一个个具体的实例，分别演示了 Java 编程技术的某几个方面，特别是网络、数据库以及服务器端的分布式对象编程，由浅入深，阐述详尽。

本书的特点是在注重系统性和科学性的同时，力求突出其实用性，在介绍相关的编程原理和知识的前提下，着重利用丰富、实用的案例来演示 Java 编程技术的魅力。

本书可以作为高等院校“Java 编程技术”课程的教材或教学参考书，也可供软件工作人员及需要开发 Java 应用软件的广大计算机用户参考阅读。

书　　名：Java 编程实用技术与案例
作　　者：杨绍方 编著
出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
责任编辑：陈红英
印 刷 者：北京密云胶印厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印 张**：22.5　**字 数**：529 千字
版　　次：2000 年 11 月第 1 版　2001 年 2 月第 2 次印刷
书　　号：ISBN 7-302-01096-X/TP・411
印　　数：5001～10000
定　　价：34.00 元

前　言

当今，Internet 正在以惊人的速度迅猛发展，它对全人类的科学研究、经济活动、工作和生活方式等各个方面产生着不可估量的影响，Internet 将成为 21 世纪社会发展最重要的基石之一。由美国 Sun Microsystems 公司开发的新一代面向对象的程序设计语言 Java，以其独有的、与网络紧密结合的特点，已经成为 Internet 领域功能最强大、最有前途的编程语言之一。

《Java 编程实用技术与案例》一书主要针对 Java 编程的初中级读者，从最基础的编程讲起，通过一些可模仿和可移植的小应用实例带领您迅速入门，并通过一些完整的实用案例的设计、分析，使读者轻松地进入“实际工作”中。本书力求“实用”，对 Java 编程基本知识的介绍主要是为了满足本书实例程序的需要，并追踪 Java 发展的最新动态，同时，尽量压缩篇幅，降低成本，减轻读者的负担。

本书的正文共 18 章。第 1 章对 Java 进行了简要的介绍；第 2 章力求使读者在最短的时间里建立起对 Java 程序的总体印象；第 3 章到第 5 章介绍了 Java 编程所必须的知识。第 6 章到第 12 章，每章以一个具体的实例，分别演示了 Java 编程技术的某几个方面，由浅入深，详尽阐述；第 13 章到第 15 章，分别介绍了 JDBC、Java 国际化和 JNI 编程技术；第 16 章到 18 章，介绍了 Java 服务器端的编程技术：RMI、CORBA 和 Servlet，在实际应用中他们通常与 JDBC 一起混合使用，共同实现复杂的、服务器端的、分布式对象的编程。

本书并不要求读者一定要逐章阅读，尽管前面的章节是后面章节的基础，但每一章都尽可能地保持独立，如果读者已经具有 Java 编程的基本知识，那么，可以根据自己的兴趣和需要从任何一章开始阅读。

书中的所有程序均在 PC 机上调试运行通过，它们展示了很难单独用语言描述的特殊性质，建议读者在阅读本书的同时录入并运行这些程序，以便加深理解。另外，许多程序仅仅是为了说明某项编程技术的使用方法，作者尽可能使它们简短，以展示相关的主题，在开发高质量的商业级软件时，读者应根据实际情况改进。

《Java 编程实用技术与案例》通俗易懂，实例丰富，既有对 Java 常见编程技术全面概括的介绍，又有对某几种典型技术深入详尽的介绍和演示，很适合于广大计算机用户参考和自学；对于高等院校计算机系高年级本科生或研究生来说，本书可以作为 Java 编程课程的教材。

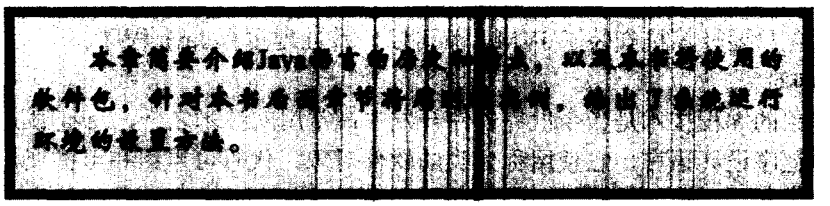
由于编著者时间和水平有限，对于书中的不足甚至错误，敬请广大读者不吝赐教。

编著者

2000 年 7 月

目　录

第 1 章　Java 语言简介



第 1 节　概　　述

1.1.1　Java 的起源

Java 语言诞生于 1991 年，它是由 Sun Microsystems 的一个开发小组在开发 Green 项目时完成的，该项目本意是开发一个用于消费类电子产品的与平台无关的软件技术，该小组最初将其称为 Oak。在 1991 到 1993 年间，这种新语言一直被认为是用来开发消费类电子产品和交互式电视控制器的利器，到 1994 年，Sun 的两个开发人员创建了 HotJava 的第一个版本，当时称为 WebRunner，它就是现在 Web 上使用的图形浏览器，之后，又被称为 Java。这时，Java 语言(包括其编译器在内)都完全是用 Oak 语言编写的(而不是 C)。1995 年 5 月，Sun 在 SunWorld 95 大会上正式推出 Java 语言，当时，由于 Internet 的飞速发展，Java 语言立即成为全球信息网舞台上一颗璀璨夺目的明星。

1.1.2　Java 语言的特点

Java 语言是一种网络编程语言，它最大限度地利用了网络资源，Applet(Java 小应用程序)可以跨平台、跨操作系统、跨网络运行。另外，由于 Applet 代码短小，易于在网络上快速地下载和发送，且具有不需要修改应用程序就可以增加 Web 页新功能的特性，因此，它在 Internet/Intranet 中得到广泛地应用。此外，Java 还配有丰富的类库，为用户编程提供了极大方便。

Java 语言的主要特点如下。

✦ 平台独立性

众所周知，Internet 由世界范围内数以万计的各种各样的计算机系统组成。这些系统中的计算机软、硬件千差万别，各不相同，要让应用软件在网络上任何一种计算机系统中都能正常地运行，就像是专门为该系统设计的一样，就必须使软件具有平台的独立性，也就是说，软件本身不受计算机硬件和操作系统的限制，软件代码可以在不同的计算机环境

中良好地运行。

长期以来，软件的平台独立性一直是软件发展的需求和编程人员追求的目标，而 Java 就是一种具有平台独立性的编程语言，它在源程序级保证了其基本数据类型与平台无关，Java 源程序编译后产生的二进制代码是一种与具体机器指令集无关的指令集合，通过 JVM(Java 虚拟机)，可以在不同平台上运行。也就是说，JVM 为我们隔离了纷繁复杂的外部网络世界，只要计算机安装了 JVM，就可以一致性地运行 Java 程序。

✦ 面向对象

面向对象的编程技术是当今世界软件开发中最常用的技术之一，Java 就是一种新型的面向对象的程序设计语言，它使用称为类的软件对象，代码可重用和可扩展，这样，可以将这些由变量和方法组成的类作为一个模板，增加其他功能来创建其他的类，无须重写父类或超类的代码，使应用程序的开发变得容易和简单，且代码较少。

✦ 安全

作为 Web 编程语言，Java 具有强大的安全结构和策略，代码在编译和实际运行过程中都会接收一层层的安全检查，可以防止恶意程序或病毒的入侵。为了实现其安全性，主要采取的措施有：取消指针操作，消除了复写内存单元和破坏有用数据的可能性；内存布局由 JVM 决定，并依赖于 Java 的运行时间(Runtime)系统和 JVM 所在的宿主机平台的特性，内存管理自动化；在字节码装载过程中使用了字节码检验器，确保了指令中参数类型的正确性、对象域访问的合理性以及操作数的边界检查(例如数组边界的自动检查)；使用类装入器，将本地类与从网络上下载的类分别置于不同空间，并对类之间的引用进行严格的审查和控制；利用原始沙箱模型，严格控制代码的访问权限；Java 软件包提供多种网络软件协议(例如 FTP，HTTP 和 Telnet)的用户接口，用户可以在网络传输中使用多种加密技术来保证网络传输的安全性和完整性。

✦ 多线程

Java 通过多线程运行机制来支持多任务和并行处理。

毋庸置疑，Java 将成为 21 世纪最重要和最有前途的 Web 编程语言之一，学习并掌握该编程语言无疑是明智的选择。

第 2 节　软件包及环境设置

除第 18 章还需要使用 jsdk2.1 以外，本书的所有实例都是基于 JDK 1.2，因此，在阅读后面的章节之前，建议读者安装并设置好自己的 Java 开发环境。

1.2.1　JDK 软件包

Java 开发工具包(Java Development Kit，JDK)是一种用于构建在 Java 平台上发布的应用程序、Applet 和组件的开发环境。

如果您还没有 JDK 1.2 和 jsdk2.1 软件包，可以从http://www.sjug.org免费下载，下载

得到的文件为 Jdk12-win32.exe，安装时直接运行该文件即可。Win32 版的 JDK 1.2 用于 Intel 硬件的 Windows 95、Windows 98 和 Windows NT(仅支持 Windows NT 4.0 版)，推荐使用 486/DX 或更快处理器，至少 48 MB RAM。在安装 JDK 软件前必须有 65 MB 可用磁盘空间，如果还要单独安装文档(docs 目录)，则还需要 85 MB 可用磁盘空间。

在 Windows 中，JDK 内还安装了独立的 Java 运行时环境(包括 Java Plug-in Control Panel)。

JDK 软件中包含一些工具，可用于开发和测试 Java 编程语言编写的程序，并可在 Java 1.2 平台上运行，这些工具以命令行的方式使用。除 appletviewer 和 policytool 外，这些工具都不提供图形用户界面，后面的章节将逐渐使用到这些工具，那时会详细地予以说明。

1.2.2　bin 目录的路径设置

运行 sysedit.exe，编辑 AUTOEXEC.BAT 文件，在 PATH 环境变量中增加 bin 目录的路径，例如：

PATH=C:\WIN98;C:\WIN98\COMMAND;c:\jdk1.2\bin

这样做的目的主要是在使用 javac 编译 Java 源代码以及使用 java 或 appletviewer 运行 Java 程序时无须指定其路径。

1.2.3　.java.policy 文件

由于 Java 安全方面的种种限制，为了使后面章节中的某些实例能够正确地运行，应该将本书第 9 章中讨论的.java.policy 文件拷贝到 Windows 的 HOME 目录下，例如 C:\WIN98 目录下。

1.2.4　设置与清除 CLASSPATH

在 DOS 提示符下，可用 set 命令修改 CLASSPATH 环境变量。其格式为：

set CLASSPATH=path1;path2 ...

路径应该以指定驱动器的字母开头，例如“C:\...”。这样，在偶然切换到不同驱动器时仍可找到类(例如，如果路径项以\...开头，并且当前位于驱动器“D:”上，则所需的类将在“D:”而不是“C:”驱动器上找)。

如果 CLASSPATH 环境变量被设置成不正确的值，或启动文件或脚本程序设置了不正确路径，则可通过使用下列命令清除 CLASSPATH：

set CLASSPATH=

该命令仅清除当前会话的 CLASSPATH，要确保在以后的会话中具有正确的 CLASSPATH 设置，则应该删除或修改启动设置。

当然，还可以按照设置 bin 目录的方式来设置 CLASSPATH。

第 2 章　初识 Java：HelloWorld

在开始学习 Java 语言编程之前，先了解一下 Java 究竟是什么样子，如何编写、编译和运行，建立一个感性的印象。即使您完全不熟悉下面的代码也没关系，尽管认用它们，在学习了后面的章节之后，很快就会觉得它们非常简单。

第 1 节　HelloWorld 应用程序

可以使用自己熟悉的任何文本编辑器(例如：Word、写字板)来编写 Java 程序，但必须保证 Java 程序是无格式的纯文本文件。

例 2.1 的代码就是 HelloWorld.java 文件的全部内容：

【例 2.1】 HelloWorld.java 程序

```
/*
 *@(#)HelloWorld.java      2000/ 05/ 15
 */

public class HelloWorld
{
  public static void main(String args[])
  {
  // 向屏幕写字符串"Hello World!"
  System.out.println("Hello World!");
  }
}
```

2.1.1　认识 HelloWorld 应用程序

☞ **提示：**

了解 C++语言的读者对程序中的注释可能相当熟悉，它们以“/*...*/”或“//”来标记。在 Java 语言中，这些注释如果遵循一些规则，就可以使用 javadoc 工具来进行软件的文档管理。这在大型软件开发中非常有用，这些规则将在第 3 章中介绍，在程序中始终使用并遵循这些规则是一种值得赞赏的良好习惯。

注意:

编写 Java 程序的首要工作就是创建一个类(class)。这里，类名与所使用的文件名必须完全一样，包括字母的大小写，否则，在编译时会提示出错。

在本例中，必须使用 HelloWorld 类名，即：

public class HelloWorld

紧接着的代码为：

public static void main(String args[])

之所以称其为应用程序，其标志就是含有 main 方法，也只有含有 main 方法的 Java 程序才能使用 java 命令来运行。

该 Java 应用程序的目的就是向屏幕输出一个字符串，"Hello World! "：

System.out.println("Hello World!");

2.1.2　编译并运行 HelloWorld 应用程序

单击“开始”，选择“程序”→“MS-DOS 方式”，打开命令行窗口，进入到 HelloWorld.java 文件所在的目录，运行下面的命令编译 Java 程序：

javac　HelloWorld.java

Java 程序编译后得到的字节码(bytecode)文件为.class 文件，也就是说，编译 HelloWorld.java 后，将得到 HelloWorld.class 文件。

运行 HelloWorld 应用程序，可以运行程序：

java　HelloWorld

如果一切正常，屏幕将显示字符串：“Hello World!”。如图 2.1 所示。

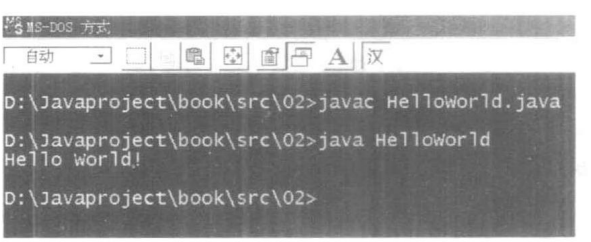


图 2.1　编译并运行 HelloWorld 应用程序

第 2 节　将 Hello World 编写为 Applet

2.2.1　Applet 的定义

从第 1 章已经了解到，Java 是一种以 Internet/Intranet 为主要对象的编程语言，Applet 则是一种可以嵌入 HTML 文件的 Java 程序。它可以通过 Internet/Intranet 下载来运行，在

许多书籍和杂志中将 Applet 称为 Java 小应用程序的原因就是因为其代码较小(但功能强大)，易于通过 Internet/Intranet 下载。Applet 中含有多个方法(method)，用于控制 Java 程序的运行和终止，例如：init()，start()，stop()等。

Applet 按其下载方式不同，可以分为本地(Local)和远程(Remote)两种。本地 Applet 是指 Applet 就存储在本地计算机上，并在本地计算机上运行，而远程 Applet 则是从网络上的其他计算机(可能在附近，或者是互联网上的任何一台计算机)下载得到的。

2.2.2 一个最简单的 Applet：HelloApplet

现在，将 2.1.1 节的 HelloWorld 应用程序编写为一个同样功能的 Applet，文件名为 HelloApplet.java，其完整的代码如例 2.2。

【例 2.2】 HelloApplet.java 程序

```
/*
*@(#)HelloApplet.java   2000/ 05/ 16
 */

import java.applet.Applet;
import java.awt.Graphics;

public class HelloApplet extends Applet
{
  public void paint (Graphics g)
  {
     g.drawString ("Hello World!",10,50);
  }
}
```

编译的方法和上一节的方法一样，运行命令：

```
javac  HelloApplet.java
```

2.2.3 认识 HelloApplet

2.2.3.1 import

在注释之后，有两行代码：

```
import java.applet.Applet;
import java.awt.Graphics;
```

在任何Applet中，都会含有类似的代码，它和C/ C++中的#include语句功能一样，它的目的是使用JDK提供的(或原先已经创建的)功能，这些功能代码位于指定的.class文件中，import后面的java.applet.Applet就是指定Applet.class文件的位置，对于熟悉C/ C++语言