# Java 2 程序设计

王克宏 主编

王少锋 姜河 钦明皖 马小军 杨文军 编著

清华大学出版社

http://www.tup.tsinghua.edu.cn

(京)新登字 158 号

## 内 容 简 介

本书介绍了 Sun 公司最新发布的 Java 2 平台中的许多高级特性，全书内容丰富，涉及到分布式应用系统、企业级计算、数据库系统、图形用户界面、Java 扩展框架、对象串行化、安全性模型等许多方面的内容。书中重点讲述了 Java 2 中许多新增加或增强的特性，并结合具体例子说明如何使用 Java 2 进行高级程序设计。

本书适合有一定 Java 语言基础的人员使用，也可作为 Java 语言的培训教材或大专院校的教材。



书　名：Java 2 程序设计
作　者：王克宏 主编　王少锋　姜河　钦明皖　马小军　杨文军 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
印刷者：北京市密云胶印厂
发行者：新华书店总店北京发行所
开　本：787×1092 1/16　印张：21.75　字数：500 千字
版　次：2000 年 9 月第 1 版　2001 年 3 月第 2 次印刷
书　号：ISBN 7-302-00719-5/TP・245
印　数：6001～11000
定　价：32.00 元

# 前　言

Internet 的发展异常迅猛，Java 是 Internet 上的编程语言，其平台无关性、面向对象、面向网络、简单等特性受到世人瞩目，正在引起一场革命。1998 年 12 月 8 日，Sun 公司最新一个关键版本的 Java Software Development Kit 正式发布，命名为 Java 2，而不是延续先前的 JDK 1.0，JDK 1.1 的惯例称为 JDK 1.2。Java 2 不仅仅是一个 JDK 的升级版本，而且还是完善的 Java 平台。

与以往的 JDK 比较，Java 2 平台提供了更多的好处，如更好的性能，更灵活的安全模式，与其它企业级应用的互用性，强健的跨平台性，应用开发更容易等。本书对 Java 2 平台作了全面的介绍，重点讲述 Java 2 中新增加的或增强的特性，主要内容包括：第 1 章是 Java 语言概述，帮助读者回顾一下 Java 语言的基础知识；第 2 章讲述 Sun 公司新发布的高性能 Java 编译器 HotSpot；第 3 章讲述 Java 2 中新的安全性模型；第 4 章讲述 Java 的扩展框架；第 5 章讲述引用对象；第 6 章讲述 Java 2 中的集合框架；第 7 章讲述 Java 的对象串行化机制；第 8 章讲述 JDBC 2.0，新的 JDBC 2.0 支持 SQL3，增强了性能和稳定性；第 9 章讲述 Java 2 处理声音的功能；第 10 章讲述新的图形用户界面 Swing；第 11 章讲述可存取性；第 12 章讲述如何在两个应用程序之间实现与平台无关的拖放功能；第 13 章讲述 Java 的应用服务；第 14 章讲述 Java 文档文件的特点和功能；第 15 章讲述 Java 的版本标识功能；第 16 章讲述 Java 的服务器端构件技术 EJB，EJB 技术是对企业计算的重要支持；第 17 章讲述 Java 本地接口；第 18 章讲述 Java 插入件；第 19 章讲述 Java 映象 API；第 20 章讲述 Java 远程方法调用 RMI，Java RMI 是开发 100%纯 Java 的分布式应用系统的重要技术；第 21 章讲述 Java 接口定义语言 IDL，Java IDL 是开发异构的分布式应用系统的重要技术。

本书主编为王克宏教授。其中第 1 章至第 8 章由姜河编写，第 9 章由杨文军编写，第 10 章和第 11 章由钦明皖编写，第 12 章和第 13 章由马小军编写，第 14 章至第 21 章由王少锋编写。王少锋还负责全书初稿的审校工作，王克宏教授对全书做了最后审校。

希望本书能够帮助读者更全面、更细致地了解 Java 2 开发平台，但由于时间紧促，加之 Java 2 平台涉及到的内容非常广泛，在编写过程中难免会有各种错误，敬请广大读者批评指正，批评意见请发往 wang_shaofeng@263.net。

编　者

2000 年 1 月于清华大学

# 目　录

# 第 1 章　Java 语言基础知识

## 1.1　Java 语言概述

### 1.1.1　Java 语言的产生

1991 年，Sun 公司的 James Gosling 和 Bill Joe 等人，为在电视、烤面包箱等家用消费类电子产品上进行交互式操作，开发了一个名为 Oak（一种橡树的名字）的软件，但当时并没有引起人们的注意。直到 1994 年下半年，Internet 的迅猛发展，万维网（WWW）的快速增长，促进了 Java 语言研制的进展，使得它逐渐成为 Internet 上受欢迎的开发与编程语言。一些著名的计算机公司纷纷购买了 Java 语言的使用权，如 Microsoft，IBM，Netscape，Novell，Apple，DEC，SGI 等。因此，Java 语言被美国的著名杂志《PC Magazine》评为 1995 年十大优秀科技产品（计算机类就此一项入选），随之出现了大量用 Java 编写的软件产品，受到计算机产业界的重视与好评。

计算机产业界不少人预言："Java 语言的出现，将会引起一场软件革命"，这是因为传统的软件往往都是与具体的实现环境有关，换了一个环境就需要作一番改动，耗时费力，而 Java 语言能在执行码（二进制码）上兼容。这样，只要所用的机器能提供 Java 语言解释器，以前所开发的软件就能运行在不同的机器上。因此，Java 的诞生对整个计算机产业产生深远的影响，对传统的计算模型提出了新的挑战。

Java 语言跨越平台的特点使用户从网络上下载 Java 程序后可以直接运行。软件开发人员编写的 Java 程序可以不加修改就可在多种平台上运行。实际上，Java 编译器产生的是字节代码（bytecode），它是独立于平台的，用户下载的就是这种字节代码，然后由用户机器上的 Java 解释器解释执行。

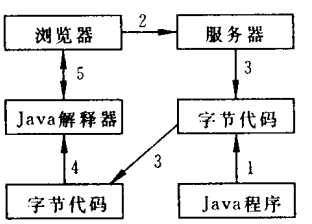

图 1-1　Java 程序的下载及执行过程

Java 程序的下载及执行过程如图 1-1 所示。

（1）Java 编译器将 Java 源程序编译成字节代码；

（2）客户机的浏览器与服务器连接，要求下载 Java 字节码文件；

（3）服务器将字节码文件传给客户机；

（4）客户机上的 Java 解释器解释执行字节代码；

（5）在浏览器上显示并与用户交互。

### 1.1.2　Java 语言的特点及其优势

Java 是一种广泛使用的网络编程语言，它是一种新的计算概念。

首先，作为一种程序设计语言，它具有简单、面向对象、不依赖于机器的结构，具有可

移植性、鲁棒性、安全性等特点，并且提供了并发的机制，具有很高的性能；其次，它最大限度地利用了网络，Java 的小应用程序(applet)可在网络上传输而不受 CPU 和环境的限制；另外，Java 还提供了丰富的类库，使程序设计者可以很方便地建立自己的系统。

下面分别从这三个方面来讨论 Java 语言的特点，然后将 Java 与 C，C++ 相比较，进一步指出它所具有的优点。

**1. Java 语言的特点**

Java 语言有下面一些特点：

(1) 使用简单

Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，开发人员只需理解一些基本概念，就可以用它编写出适合于各种情况的应用程序。Java 语言略去了运算符重载、多重继承等模糊的概念，并且通过内存垃圾自动收集机制大大简化了程序设计者在内存管理方面的工作。另外，Java 也适合于在小型机上运行，它的基本解释器及类的支持只有 40KB 左右，加上标准类库和线程的支持也只有 215KB 左右。

(2) 面向对象

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一组对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的重用。

(3) 分布性

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过通用资源定位指针 URL 地址在网络上很方便地访问其它对象。

(4) 鲁棒性

Java 在编译和运行程序时，都要对可能出现的问题进行检查，以避免程序中错误的发生。它提供内存垃圾自动收集机制来进行内存管理，防止程序员在管理内存时产生错误。通过集成面向对象的例外处理机制，在编译时 Java 提示出可能出现但未被处理的例外，帮助程序员正确地进行选择以防止系统的崩溃。另外，Java 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

(5) 安全性

在网络和分布环境下，防止病毒的入侵是必须重视的重大问题。Java 语言不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止了程序员使用“特洛伊”木马等欺骗手段访问对象的私有变量，同时也避免了指针操作中容易产生的错误。

(6) 体系结构中立

Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行系统，Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台上运行。

(7) 可移植性

与平台无关性使得 Java 程序可以方便地被移植到网络上的不同机器。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语

言实现的，Java 运行系统由标准 C 语言实现，这使得 Java 系统本身也具有可移植性。

(8) 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时信息，使得连接过程更加简单。

(9) 高性能

和其它解释执行的语言如 BASIC，TCL 不同，Java 字节代码的设计使之能够很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

(10) 多线程

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易地实现网络上的实时交互行为。

(11) 可扩展性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。除此之外，Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

**2. Java 丰富的类库**

Java 提供了大量的类以满足网络化、多线程、面向对象系统的需要。

(1) 语言包　它提供的支持包括字符串处理、多线程处理、例外处理和数学函数处理等，可以用它简单地实现 Java 程序的运行平台。

(2) 实用程序包　它提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

(3) 输入输出包　它用统一的“流”模型来实现所有格式的 I/O，包括文件系统、网络、输入/输出设备等。

(4) 低级网络包　它用于实现 Socket 编程。

(5) 抽象图形用户接口包　它实现了不同平台计算机的图形用户接口部件，包括窗口、菜单、滚动条、对话框等，使得 Java 可以移植到不同平台的机器。

(6) 网络包　它支持 Internet 的 TCP/IP 协议，提供了与 Internet 网的接口。它支持通用资源定位指针 URL 连接、对万维网(WWW)的即时访问，并且简化了客户/服务器模型的程序设计。

(7) java.applet 包　它提供了用于编写 Java Applet 程序的类库。

Java 2 平台提供了许多新的包，在本书的后面章节中将作详细介绍。

**3. Java 与 C，C++语言的区别**

对于变量声明、参数传递、操作符和流控制等，Java 使用了和 C，C++相同的内容，使得熟悉 C，C++的程序员能很方便地进行编程。同时，Java 为了实现其简单、鲁棒、安全等特性，也摒弃了 C 和 C++中许多不合理的内容。

(1) 全局变量

Java 程序中，不能在所有类之外定义全局变量，只能通过在一个类中定义公用、静态的变量来实现一个全局变量。例如：

```
Class GlobalVar{
```

```
    public static global_var;
}
```

在类 GlobalVar 中定义变量 global_var 为 public static，使得其它类可以访问和修改该变量。

Java 对全局变量进行了更好的封装。而在 C 和 C++中，依赖于不加封装的全局变量常常造成系统的崩溃。

(2) Goto

Java 不支持 C，C++中的 goto 语句，而是通过例外处理语句 try，Catch，final 等来代替 C，C++中用 goto 来处理遇到错误时跳转的情况，使程序更可读且更结构化。

(3) 指针

指针是 C，C++中最灵活，也是最容易产生错误的数据类型。由指针所进行的内存地址操作常会造成不可预知的错误，同时通过指针对某个内存地址进行显式类型转换后，可以访问一个 C++中的私有成员，从而破坏安全性，造成系统的崩溃。而 Java 对指针进行完全的控制，程序员不能直接进行任何指针操作，例如不能够把整数转化为指针，或者通过指针释放某一内存地址等。同时，数组作为类在 Java 中实现，很好地解决了数组访问越界这一 C，C++中不作检查的错误。

(4) 内存管理

在 C 语言中，程序员通过库函数 malloc( )和 free( )来分配和释放内存，C++中则通过运算符 new 和 delete 来分配和释放内存。再次释放已释放的内存块或未被分配的内存块，会造成系统的崩溃；同样，忘记释放不再使用的内存块也会逐渐耗尽系统资源。而在 Java 中，所有的数据结构都是对象，通过运算符 new 为它们分配内存块。通过 new 得到对象的处理权，而实际分配给对象的内存可能随程序运行而改变，Java 对此自动地进行管理并且进行内存垃圾收集，有效地防止了由于程序员的误操作而导致的错误，并且更好地利用了系统资源。

(5) 数据类型的支持

在 C，C++中，对于不同的平台，编译器对于简单数据类型如 int，float 等分别分配不同长度的字节数，例如，int 在 IBM PC 中为 16 位，在 VAX-11 中为 32 位，这导致了代码的不可移植性，但在 Java 中，对于这些数据类型总是分配固定长度的位数，例如，对 int 型，它总是占 32 位，这就保证了 Java 的平台无关性。

(6) 类型转换

在 C，C++中，可以通过指针进行任意的类型转换，因而常常带来不安全性，而在 Java 中，运行系统对对象的处理要进行类型相容性检查，以防止不安全的转换。

(7) 头文件

C、C++中用头文件来声明类的原型以及全局变量、库函数等，在大的系统中，维护这些头文件是很困难的。而 Java 不支持头文件，类成员的类型和访问权限都封装在一个类中，运行系统对访问进行控制，防止对私有成员的操作。同时，Java 中用 import 语句来与其它类进行通信，以使用它们的方法。

(8) 结构和联合

C,C++中的结构和联合中所有成员均为公有,这就带来了安全性问题。Java 中不包含结构和联合,所有的内容都封装在类中。

(9) 预处理

C,C++中用宏定义来实现的代码给程序的可读性带来了困难。在 Java 中,不支持宏,它通过关键字 final 来声明一个常量,以实现宏定义中广泛使用的常量定义。

### 1.1.3 Java 应用程序结构和执行机制

**1. Java 应用程序结构**

使用 Java 语言编写的程序可以分为两类:Application 和 Applet。

Application 称为独立的应用程序,它的执行不依赖于其它的程序,只要有 Java 解释器就可以运行 Application。Java Application 可以同使用其它语言编写的应用程序一样控制和使用计算机的资源。

Applet 被称作 Java 的小应用程序,它是动态的、安全的、跨平台的网络应用程序。Java Applet 不能独立运行,只能由支持 Java 的浏览器来运行它。Java Applet 嵌入 HTML 语言,通过主页发布到 Internet。网络用户访问服务器的 Applet 时,这些 Applet 从网络上进行传输,然后在支持 Java 的浏览器中运行。由于有了 Applet,HTML 页面从静态的文本、图像变成了动态的可执行程序。利用 Applet 可以完成图像、声音、动画等内容的下载和播放,使 HTML 页面变得更加生动。Applet 还可以完成与用户的交互操作以及和服务器的网络通信,因此极大地丰富了 WWW 的功能。由于 Java 语言的安全机制,用户一旦载入 Applet,就可以放心地来生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。

下面是将 Java Applet 嵌入到 HTML 文件中的一段代码:

```
<applet code=HelloWorld.class  width=100  height=100>
</applet>
```

图 1-2 是一个运行在浏览器中的 Java Applet。

由于 Java Applet 是从网络上下载的执行程序,属于不可信任的代码。因此除了在加载时进行安全性检查外,在 Applet 的运行过程中,也要受到一些限制,主要有以下几点:

(1) 不允许 Applet 执行客户端的其它程序;

(2) 不允许 Applet 读写客户端的文件;

(3) 不允许 Applet 调用客户机的本机方法;

(4) 不允许 Applet 与除了它所下载的服务器以外的计算机建立网络连接。

**2. Java 应用程序执行机制**

Java 程序是由在特定平台下运行的 Java 解释器来解释执行的,解释器对 Java 程序屏蔽了底层的操作系统和硬件平台的不同,因此同一个 Java 程序代码可以不加修改地运行在不同的硬件平台和操作系统上。可以说 Java 程序代码是在一个 Java 虚拟机(JVM, Java Virtual Machine)上运行。Java 虚拟机有自己的规范、指令集、寄存器等。编译后的 Java 程序代码(即字节码)就是 Java 虚拟机的指令。虚拟机将这些指令转换为特定硬件平