普通高等教育“十一五”国家级规划教材

高等职业院校计算机教育规划教材

Gaodeng Zhiye Yuanxiao Jisuanji Jiaoyu Guihua Jiaocai

Java程序设计

（第二版）

Java CHENGXU SHEJI

朱喜福 编

- 零起点学习Java编程，通俗易懂
- 内容新颖、概念清晰、详略得当
- 实例丰富、联系实际、侧重应用

人民邮电出版社
POSTS & TELECOM PRESS

精品系列

普通高等教育“十一五”国家级规划教材

高等职业院校计算机教育规划教材

Gaodeng Zhiye Yuanxiao Jisuanji Jiaoyu Guihua Jiaocai

Java程序设计

（第二版）

Java CHENGXU SHEJI

朱喜福　编

人民邮电出版社

北京

精品系列

图书在版编目（CIP）数据

Java 程序设计/朱喜福编. —2 版. —北京：人民邮电出版社，2007.3（2008.6 重印）
ISBN 978-7-115-15764-5

Ⅰ. J… Ⅱ. 朱… Ⅲ. JAVA 语言—程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 005680 号

内 容 提 要

Java 语言是目前最流行的面向对象的网络编程语言之一。本书从最基本的入门概念开始，对 Java 面向对象程序设计的基本概念和技术等内容进行较为详细的讲解；并通过大量的编程实例讲述如何使用 Java 语言及其类库编写解决实际问题的 Java 应用程序和 Java 小应用程序；对 Java 中提供的常用数据结构类的使用、异常和多线程的概念和应用等进行了细致和深入的讲解；对 Java 的输入输出处理、图形用户界面的设计等也进行深入的介绍。本书每章都安排了大量有针对性的编程实例及练习题，并对全书的内容给出了一些综合应用的实例。

本书可作为高等院校应用型本科、高职高专的教材和教学参考书，也可作为对 Java 编程感兴趣的读者的入门参考书。

普通高等教育“十一五”国家级规划教材
高等职业院校计算机教育规划教材
Java 程序设计（第二版）

◆ 编　　朱喜福
责任编辑　潘春燕
执行编辑　李海涛

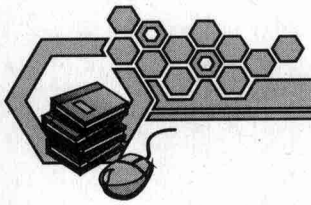
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：22
字数：530 千字　　2007 年 3 月第 2 版
印数：8 001 – 10 000 册　　2008 年 6 月北京第 4 次印刷

ISBN 978-7-115-15764-5/TP

定价：32.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154



丛书出版前言

目前，高职高专教育已经成为我国普通高等教育的重要组成部分。在高职高专教育如火如荼的发展形势下，高职高专教材也百花齐放。根据教育部发布的《关于全面提高高等职业教育教学质量的若干意见》（简称16号文）的文件精神，本着为进一步提高高等教育的教学质量和服务的根本目的，同时针对高职高专院校计算机教学思路和方法的不断改革和创新，人民邮电出版社精心策划了这套高质量、实用型的教材——“高等职业院校计算机教育规划教材”。

本套教材中的绝大多数品种是我社多年来高职计算机精品教材的积淀，都经过了广泛的市场检验，赢得了广大师生的认可。为了适应新的教学要求，紧跟新的技术发展，我社再一次组织了广泛深入的调研，组织了上百名教师、专家对原有教材做认真的分析和研讨，在此基础上重新修订出版。

本套教材中虽然还有一部分品种是首次出版，但其原稿也经过实际教学的检验并不断完善。因此，本套教材集中反映了高职院校近几年来的教学改革成果，是教师们多年来的教学经验的总结。本套教材中的每一部分作品都特色鲜明，集高质量与实用性为一体。

本套教材的作者都具有丰富的教学经验和写作经验，思路清晰，文笔流畅。教材编写充分体现高职高专教学的特点，深入浅出，言简意赅。理论知识以“够用”为度，突出工作过程导向，突出实际技能的培养。

为方便老师授课，本套教材将提供完善的教学服务体系。教师可通过访问人民邮电出版社网站 http://www.ptpress.com.cn/download 下载相关资料。

欢迎广大教师对本套教材的不足之处提出批评和建议！

编 者 的 话

近年来，随着我国经济和社会发展进一步与国际接轨，计算机网络正在以前所未有的速度迅猛发展，对计算机网络应用人才的需求也在不断增加。在网络程序设计应用领域，Java面向对象的编程技术正逐步成为计算机网络应用开发的主流，从电子商务、远程教学到网络游戏等都在纷纷使用Java技术，Java手机编程和基于Java技术的各种芯片的应用等在日常生活中也随处可见。

相比其他的编程语言，学习和掌握Java语言更为容易，因此，全国各地高校及高职高专都相继开设了Java程序设计或与Java技术相关的课程，并将重点放在培养掌握应用技术为主的中等层次人才上，以适应当前社会对信息技术和网络应用人才的需求。本书作者在应用型本科和高职高专计算机应用技术和网络专业教授“Java程序设计”课程的实践中，积累了一定的教学经验和体会，深感学生理解和掌握Java面向对象的编程技术、应用庞大的Java类库编写解决实际问题的程序等都有一定的难度。因此，在编写《Java程序设计》教材时，根据学生的特点，在内容取舍、编排顺序、概念描述和讲述方法等方面都做了较多的探讨和实践，力争能较好地适合学生使用，讲述内容紧跟Java语言及其相关技术的发展。本书内容新颖、结构合理、概念清晰、通俗易懂、实用性强。例题的选择既考虑加深对知识的理解和掌握，又考虑到学生的学习兴趣和编程应用，并对例题进行了详细的讲解和分析。每章实训习题的编写具有较强的针对性，以帮助学生巩固所学知识和提高实际编程能力。根据学生基础不同和讲述内容的取舍不同，本书的讲授可安排30～54课时（不含上机实训）。

全书的内容编排和有关教学的建议如下。

第1章对Java程序的开发和运行环境、Java语言的基础做了基本的、较为详细的说明，尽可能使学生通过本章的学习，对Java语言和使用Java语言编程有一个基本的概念。本章重点介绍Java程序的基本组成、基本数据类型、变量和方法的定义和使用、类和对象的基本概念、循环控制结构等。为了便于后面章节内容的学习，本章的内容还需要学生在学习过程中多次复习和查阅。本章的教学可安排2～6课时。

第2章给出了大量解决简单问题的程序，重点在巩固、强化和拓展第1章学习的有关Java编程的基础知识和概念等。通过本章的例题讲解和上机实训，可使学生对进一步学习Java编程打下扎实的基础。本章的教学可安排4～8课时。

第3章讲述Java面向对象程序设计的基本概念和基本技术，通过大量的实例使学生理解面向对象程序设计中的抽象、封装、继承和多态等重要概念，并说明其在程序设计中的具体应用。本章的教学可安排6～10课时。

第4章对Java语言的类库进行简单的介绍，并重点讲述随机数的产生、字符串的使用、常用数据结构在Java中的实现和使用，说明了Java帮助文档及其查阅方法。本章的教学可安排4～8课时。

第5章对Java中的异常处理、多线程的基本应用等通过实例进行了讲述。本章的教学可安排2～4课时。

第 6 章讲述 Java 的输入输出处理。本章的教学可安排 4～6 课时。

第 7 章讲述 Java 的图形用户界面的编程，重点讲述 java.swing 包中的相关类的使用和编程应用。包括字体和颜色的设置、图形绘制和图像显示、Graphics2D 画图，对 Swing 包中的常用组件及相应的事件处理列举了大量的程序实例，从应用角度对各种布局、对话框应用、菜单设计和定时控制等做了较为详细的讲解。本章的教学可安排 8～12 课时。

考虑到本课程教学和学生的实际情况，本书没有对 Java 的套接字通信、数据报通信编程和 JDBC 数据库编程等内容进行讲述，这些内容以及其他有关 Java 编程的内容（如 JSP、J2EE 和 J2ME 等）将在本书的后续教材《Java 网络应用编程入门》（人民邮电出版社）中讲述，其内容可作为高职高专及应用型本科学生的综合实训周的实训内容，以对本书所学知识内容的较为全面、综合的拓展和补充。通过全书内容的学习和编程练习，力求使学生掌握深入学习 Java 的许多必备知识，比较深入全面地掌握面向对象编程技术，并打下比较扎实的利用 Java 类库编写和开发 Java 程序的基础，为进一步学习 Java 技术提供很好的入门。

编　者

2006 年 10 月

目　录

第 1 章　Java 语言基础

1.1　计算机编程语言

计算机系统由硬件和软件两大部分组成。硬件是构成计算机的各种物理设备和组件。软件是指为了运行和管理计算机，使计算机实现图形图像处理、网络通信、数据管理及多媒体应用等各种功能而编制的程序。这些程序是由各种计算机语言所编写的代码组成的。

计算机语言可以分为 3 大类：机器语言、汇编语言和高级语言。机器语言由二进制代码（0 或 1）串组成，并且是唯一能被 CUP 直接“理解”的语言。汇编语言是利用指令助记符来代表机器语言指令进行程序设计的语言，这些助记符因机器类型、型号而异。由于程序员编写的程序不能直接被机器“理解”，需要通过汇编语言软件翻译成机器语言表示的目标程序，再通过连接形成可执行程序，才能在计算机中执行。由于汇编语言是面向机器的，即为特定的处理器而设计的，因此对于不同的机器要重新编码，同时使用汇编语言编写的程序即使完成一个简单的任务也需要多条指令。高级语言采用类似自然英语及数学符号来书写语句，组成程序。高级语言是面向用户的语言，使用一条简单的高级语言语句就可以完成由许多条汇编语句才能完成的任务。高级语言编写的程序同样不能直接被机器“理解”，需要将其转化为机器语言，这是通过执行称为“编译器”的编译程序实现的。高级语言比机器语言和汇编语言都更易学、易用和易于理解，同时提供了大量的帮助文档。高级语言编写的程序更易于维护，不受某一机器类型的限制，几乎可以不加修改地用于不同计算机。

高级语言程序的翻译和执行方法可归类为两大基本技术：编译执行和解释执行。像 C 或者 C++这类编译语言写出来的程序，通过编译程序处理后，其目标语言是与之相对应的机器语言，它们能被独立地执行。像 BASIC 这种解释语言编写的程序，通过解释程序所处理的翻译和执行过程通常是交替执行的，源程序某一部分所生成的代码在其产生时就被执行。Java 语言是解释执行的高级编程语言。

1.2　Java 语言发展

Java 语言诞生于 20 世纪 90 年代初期，适用于在 Internet 环境下编写各种网络应用程序，并且具有平台无关性等特点，现已迅速发展成为最受欢迎的计算机网络编程语言之一。

Java 语言是面向对象的程序设计语言，它是解释执行的，能跨平台使用，具有较高的性能和高度的安全性，并且支持多线程，具有内存垃圾自动收集机制。

最早的 Java 版本 JDK1.0 于 1996 年正式推出，经不断改进和升级后，发布了 JDK1.1、JDK1.1.5 等版本。1999 年升级为 1.2 版，并改称为 Java 2。2000 年发布了 Java 2 的 1.3 版本。

2002 年发布了 Java 2 的 1.4 版本。这些升级的版本使 Java 程序运行更快、多媒体功能更强，并扩充了对网络的支持和对 XML 的处理。随着 Java 技术的不断发展，它根据市场进一步细分为以下 3 个版本：针对企业级 e-Business 架构和 Web 服务开发与应用的平台 J2EE（Java 2 Enterprise Edition）；针对普通 PC 应用的 Java 开发平台 J2SE（Java 2 Standard Edition）；针对嵌入式设备及消费类电器（如手机、智能卡等）的开发平台 J2ME（Java 2 Micro Edition）。

1.3 Java 程序的开发和运行环境

本书使用的是针对普通 PC 应用的 Java 开发平台 J2SE，其全称是“Java 2 Software Development Kit, Standard Edition”，它是 Java 2 开发的标准版，简称 j2sdk。随着 Java 技术的发展，它不断升级为新的版本，如 j2sdk1.4、j2sdk1.5 等。本书所讲程序都是在 Windows 操作系统下实现的，使用的是基于 Windows 下的 j2sdk，具体版本名称是 jdk-1_5_0-windows-i586.exe。j2sdk 是基于命令行（DOS 界面）的开发环境，在该环境下需要打开 DOS 界面编译和运行程序。许多公司提供了付费的、更方便的图形界面下的 Java 集成开发环境，有兴趣的读者可参考相关书籍来了解和熟悉它们的使用。

j2sdk 可以在 Sun 公司的网站（http://java.sun.com）免费下载，也可以在其他提供计算机编程软件下载的网站下载。为了方便学习和使用 Java，Sun 公司提供了中文版 Java 帮助文档。

例如下载的是 jdk-1_5_0-windows-i586.exe，若没有改变默认的安装路径，则被安装在 c:\j2sdk1.4.2 目录下，该目录下有一个子目录 bin，编译和运行 Java 程序的命令都在该目录下。

为了能在任何目录下都能编译和运行 Java 程序，可在 c 盘的根目录下编写批处理文件 autoexec.bat，在该文件中加入：

```
set path=%path%; c:\Program Files\Java\jdk1.5.0\bin
set classpath=%classpath%;e:\gzk;.
```

上面第 1 行设置编译和运行程序所需要的 Java 命令的安装路径 c:\Program Files\Java\jdk1.5.0\bin；第 2 行设置编译和运行的字节码文件（.class）的路径 e:\gzk，这是自定义的文件夹，里面存放 Java 源文件（.java）和编译后的字节码文件（.class），分号“;”是 windows 路径分隔符，“.”表示当前目录。保存文件 autoexec.bat，并重启计算机，这样 DOS 界面下的任何目录都能识别 c:\Program Files\Java\jdk1.5.0\bin 目录下的 Java 命令，并能运行 e:\gzk 目录下的字节码文件和当前目录下的字节码文件。

尽管 Java 是解释执行的编程语言，也需要对源程序进行编译（使用 bin 目录下的 javac 编译源程序），但与 C 语言这种全编译（编译后生成可执行文件）的语言不同，Java 具有半编译、半解释的特性。Java 程序编译后不生成可执行文件，而是生成一种称为字节码（Bytex Code）的中间格式文件，这种字节码文件的后缀名是.class，字节码文件的格式是与平台无关的。不管 Java 应用程序在何种平台下编译，只要能够运行 Java 编译器，最终转换出来的字节码都是相同的，因此，也能够在任何支持 Java 虚拟机（Java Virtual Machine, JVM）的计算机上运行。JVM 是一个特殊的程序，它知道如何解释和执行 Java 字节码。Java 程序并不直接在操作系统的控制之下运行，而是在 JVM 的控制之下运行，但 JVM 本身直接运行在操作系统的控制之下。只要有给定平台上的适当版本的 JVM（Sun 公司提供了许多不同平台的

JVM），字节码文件便能够运行。

1.4　能够运行的两类 Java 程序

本节介绍两类能够运行的 Java 程序：Java Application，也称为 Java 应用程序；Java Applet，也称为 Java 小应用程序或 Java 小程序。

下面通过例子对这两类 Java 程序的构成进行简单的说明，然后学习如何编译和运行 Java 程序。

1.4.1　Java Application 的编译和运行

Java Application 通常翻译为 Java 应用程序，例 1.1 就是一个最简单的 Java 应用程序。

例 1.1　一个简单的 Java 应用程序：App1.java。

/*这是一个最简单的 Java 应用程序，其功能是在 DOS 界面输出字符串：Hello,World!。通过该程序来演示 Java 应用程序的编译和运行。*/

```
1:  public class App1 {
2:      public static void main(String[] args) {
3:          System.out.println("Hello,World!");      //输出字符串：Hello,World!
4:      }
5:  }
```

图 1-1 所示为 App1.java 的编译和运行结果。

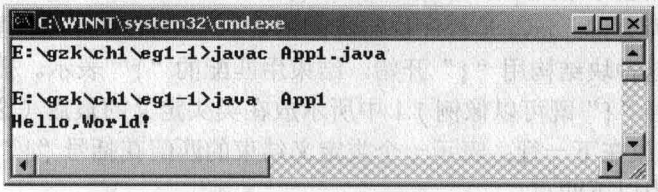


图 1-1　例 1.1 的编译和运行结果

1. 源程序的编写与保存

最简单的编写 Java 源程序的方法如下：在 Windows 的 DOS 界面下，进入将要保存文件的目录（例 1.1 保存文件的目录是：E:\gzk\ch1\eg1-1），键入 edit 命令并回车，然后键入例 1.1 中第 1 行～第 5 行的代码（注意：程序中的数字 1～5 和后面的冒号“:”是为了解释程序方便加上的行号，不是程序本身的内容，因此不要键入），然后将文件保存为 App1.java，至此源程序的编写工作就完成了。也可通过 Windows 下的记事本完成源程序的编写过程。

Java 程序是由类（class）构成的，其源文件是以.java 为后缀名的文本文件，文件名称必须与程序中类的名称 App1 完全一致。若将文件保存为 app1.java，程序编译时将出错，因为 Java 语言是区分大小写的。

2. 如何编译源程序

在命令行编译 Java 程序的命令是 javac，空格后写上需要编译的源文件 App1.java，回车

后则对源文件 App1.java 进行编译，如图 1-1 所示。如果源程序没有错误，则屏幕没有任何输出，只在当前目录下生成 App1.class 的字节码文件。如果源程序有错误，则屏幕输出错误提示，改正程序中的错误后再编译，直到程序编译通过并生成 App1.class 的字节码文件。

3. 如何运行程序

有了应用程序的字节码后，则可以运行该字节码文件。由于 Java 是解释执行的，因此需要在命令行先键入解释执行字节码文件的命令 java，空格后写上需要解释执行的字节码文件名 App1，回车后则运行程序 App1，如图 1-1 所示。图 1-1 中键入的 Java 命令是 Java 的解释器，它运行了 Java 虚拟机，通过虚拟机运行字节码文件 App1，程序运行后在命令行输出字符串“Hello,World!”。注意运行字节码文件时，只需写文件名，不能跟后缀名.class，否则将出错。

4. 源程序的解释说明

对例 1.1 的源程序解释如下。

（1）定义类

程序的第 1 行～第 5 行定义了名字为 App1 的类（class）。

程序第 1 行中的 public class App1 称为类头。类头中小写的 public（公共的）和 class（类）是 Java 语言中有特殊含义的英文单词，通常称为关键字，关键字对于编译器来说有特殊意义，用户只能根据 Java 中的特定规则来正确使用（第 2、3 章将会学习 Java 中许多关键字的含义和使用）。关键字 class 用来定义一个新的类，App1 是该类的名字，称为类名，它由用户自己给出，不能使用关键字命名一个类。如果一个类用 public 修饰，表明该类是公共的，能被任何其他类使用。

可见，类头由类的修饰符（如 public）、关键字 class 和自定义的类名组成。

例 1.1 中，类头定义的后面有一对花括号“{}”，构成类的实体的内容都放在类定义的花括号里面，通常称为类体。Java 是程序块结构的语言，一个类就是一个程序块，是构成 Java 程序的基本单位。这种块结构用“{”开始，结束用匹配的“}”表示。需要注意的是，定义一个类的开始花括号“{”既可以像例 1.1 中所示放在类头定义的最后（这是大多数程序员的习惯），也可以单独放在下一行；表示一个类定义结束的匹配花括号“}”单独放置在一行。

综上所述归纳为以下两点。

① Java 程序是由类（class）构成的，用关键字 class 来定义一个新的类，关键字都是小写的，如果用 Class 定义一个新的类，程序编译时将出错。

② Java 中类的最简单的定义格式如下：

```
[类的修饰符] class 类名 {
        构成类的实体的各种代码
}
```

上面类定义格式中，方括号“[]”中类的修饰符是可选的，在类定义中增加修饰符表明了该类具有特殊的含义。

（2）定义类中的方法和 main 方法的说明

构成一个类的实体的内容有变量和方法，类中定义的变量描述类的属性；类中定义的方法，描述该类具有的一些行为，实现一定的功能。变量和方法将在后面的章节中详细讲述，这里只对它们简单说明。

Java 中的方法是指 Java 语言中按一定格式书写的、具有一定功能的程序代码块。

Java 中的方法也是一种块结构，与类的定义相似，方法的这种块结构定义也包含方法头和方法体。其中，方法头的定义中必须有返回值类型、方法名，方法的名字后必须跟一对圆括号，圆括号里指明方法接收的参数类型（可能为空，表明该方法不接收参数）；方法头后面也有一对花括号“{}”，构成方法的实体的内容（完成一定功能的程序代码段）放到这一对花括号“{}”里面，称为方法体。定义一个方法的基本格式如下：

```
[方法的修饰符] 返回值类型 方法名字（[方法所接收的参数类型　参数名称]）{
        实现方法功能的代码;
}
```

上面方法定义格式中，方括号中的内容是可选的，即方法定义中可以没有修饰符和参数，修饰符的作用是使方法具有一些特殊的含义。

定义好一个方法后（即按上面的格式写出方法头和方法体中的代码），如果要具体实现方法中代码所描述的功能时，需要通过方法名调用该方法就可以完成方法的功能，如何调用方法将在下面和本章 1.4.1 节进行必要的说明。

例 1.1 中程序的第 2 行～第 4 行定义了名字为 main 的方法，这里的 main 方法是 Java 应用程序必须具有的一个特殊方法，其特殊性表现在它由 Java 虚拟机自动调用，是应用程序执行的入口，即当运行一个 Java 应用程序时，Java 虚拟机将从 main 方法中的第 1 行代码开始执行，直到执行完 main 方法中的所有代码。每一个 Java 应用程序都要有一个如下所示的 main 方法：

```
public static void main (String[ ] args){
}
```

例 1.1 程序中的第 2 行的是 main 方法的方法头，public、static 和 void 等都是 Java 中有特殊含义的关键字。public 表明所有的类都可以使用这一方法；static（静态的）指明该方法是一个静态方法（静态方法的含义在第 3 章讲述）；void 则指明 main( )方法不返回任何值；方法的名字是 main，定义方法时，方法的名字后必须跟一对圆括号( )，圆括号中的 String[ ] args 说明了传递给 main 方法的参数是 String（字符串）类型的数组，参数的名字 args，实际上参数的名字可以随意命名，例如圆括号中写为 String[ ] x 也是可以的。String（注意第一个字母 S 大写）是 Java 本身已定义好的字符串类。字符串是放在双引号""中的一串字符，例如"中国"、"12345"、"abcxyz"等都是字符串。

（3）方法调用——System.out.println 的说明

程序中的第 3 行是 main 方法的全部代码，这一行代码以分号结尾，称为一条语句。Java 中每条语句都以分号结尾。

System.out.println("Hello,World!")；是一个方法调用语句，它把圆括号中的字符串“Hello,World!”输出到屏幕上，输出完结果后光标移到下一行的开始位置。

请注意定义方法和调用方法的区别。方法定义是按照符合 Java 语言规范的格式把方法需要实现的功能用语句序列表示出来，如果需要向方法传递参数，需要在方法名字后面的圆括号中指明参数类型，通常称为形式参数。调用方法是具体实现方法的功能，调用方法时先通过方法名指明调用哪个方法，并把方法的形式参数替换为同类型的具体参数（通常称为实际参数），具体实现方法的功能。例如，例 1.1 程序中并没有 println 方法的定义，实际上 println