Java 经典教材译丛

Java 程序设计大全

著者：[美] Joyce Farrell
译者：武嘉澍

THOMSON

北京大学出版社
http://cbs.pku.edu.cn

Java 经典教材译丛

Java 程序设计大全

Java Programming: Comprehensive

[美] Joyce Farrell 著
武嘉澍 译

内 容 简 介

本书由教学经验丰富的教师编写，用来指导初级程序员使用 Java 程序设计语言开发应用程序和 applet。本书主要介绍面向对象的程序设计的概念，以及实现程序设计的 Java 语法规则，其中许多新的语言特性，如继承、异常处理、AWT 等使得新技术得以直接应用。通过对本书的学习，编写应用程序和 applet——按照从下到上的方式，而不是使用预先编写的对象编译应用程序和 applet，有助于读者更深刻地理解面向对象的程序设计中所用到的概念。

本书的目的是帮助读者深刻地理解 Java 如何支持编程技术，从而成为一名优秀的程序设计人员。本书适合作为高校计算机专业 Java 语言和面向对象编程等课程的教科书，也是 Java 程序员和爱好者必备的参考书。

Original English language title:Java Programming:Comprehensive,by Joyce Farrell.
EISBN: 0-7600-1070-6

981-254-144-6

北京市版权局著作权合同登记号　图字：01-2003-8587 号

图书在版编目(CIP)数据

Java 程序设计大全/(美)法雷尔(Farrell,J.)著；武嘉澍译. —北京：北京大学出版社，2003.12
(Java 经典教材译丛)
ISBN 7-301-06675-9
Ⅰ. J… Ⅱ. ①法… ②武… Ⅲ. JAVA 语言—程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2003)第 099258 号

书　　名：Java 程序设计大全
著作责任者：[美] Joyce Farrell 著
译　　者：武嘉澍
责任编辑：温丹丹
标准书号：ISBN 7-301-06675-9/TP・0742
出 版 者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
网　　址：http://cbs.pku.edu.cn　http://www.macrowin.net
电　　话：发行部 62750672　编辑部 62750581　邮购部 62752015
电子信箱：xxjs@pup.pku.edu.cn　macrowin@macrowin.net
排 版 者：北京东方人华北大彩印中心　电话：62754190
印 刷 者：河北涿县鑫华书刊印刷厂
发 行 者：北京大学出版社
经 销 者：新华书店
787 毫米×1092 毫米　16 开本　25.75 印张　799 千字
2003 年 12 月第 1 版　2003 年 12 月第 1 次印刷
定　　价：39.00 元

编 者 序

欢迎学习 Java 程序设计语言！

自计算机发明以来，无处不在显示着它对整个人类文明和社会进步产生着如此巨大、如此深刻的影响，推动着人类社会一日千里地向前发展

历经半个多世纪，计算机软件——计算机的重要组成部分，也发生了日新月异的变化。从机器语言到汇编语言再到高级语言，从结构化语言到面向对象语言，人们不断地探索和发明新的、功能强大又简单易学的计算机语言。

用计算机语言编程，有如我们用世界上的任何一种语言写作一样，两者都是创造性的工作，而且都要用到相应的技术。如果您掌握了一种语言，熟知它的词汇和语法规则，也许您能熟练地运用这种语言进行表达，但这并不能使您成为一位作家。写作是一种创造性的工作，它需要有创造性的人来做。用计算机语言编程也是同样的道理。也许您掌握了当今最流行的程序设计语言，而且也能迅速地写出程序语句，但是，程序的内容只能靠您的创造性来产生。

1. 为什么要学习 Java 程序设计语言？

工欲善其事，必先利其器。没有得心应手的工具又怎么能随心所欲地发挥创意？计算机高级语言众多，而 Java 语言的得天独厚的优点，使它从众多语言中脱颖而出，越来越受到人们的青睐。Java 是一种跨平台、适合于分布式计算环境的面向对象编程语言。具体来说，它具有如下特性：简单、面向对象、分布式、易解释、可靠、安全、与平台无关、体系结构中立、可移植、高性能、多线程、动态等。

鉴于此，我们从国外遴选了 8 本流行的 Java 书，组成了《Java 经典教材译丛》，以期读者能从中受益。

2. 您渴望从《Java 经典教材译丛》学到什么？

博大精深是我们选此套丛书的宗旨，从 Java 基本概念和技术到其高级主题，丛书的内容涵盖了 Java 的全部知识。原书多由国外教学经验丰富的教师编写，书中提供大量的实例和案例。为了给读者提供良好的服务，我们将大部分书中的实例中的源代码放在 http://cbs.pku.edu.cn 中的【下载专区】上，读者可从中下载。

3. 什么人使用《Java 经典教材译丛》？

无论您是一位计算机语言初学者，还是一位熟知计算机语言的编程人员，《Java 经典教材译丛》中必有一本适合您。当然，本套丛书主要针对在校大专院校的学生而编写，所以大多数读者都可以轻松上手，教师更可以从中找到教学用书和参考读物，部分书中带有星号(*)的章节则是为更高层次的读者而设计，为选学内容。

4. 对计算机的硬件和软件有什么需求？

- 软件　可以通过访问 Sun 的网站(http://java.sun.com)下载 JDK 编译程序，下载的软件大约为 20MB。
- 硬件　必须具备可以运行 Windows 操作系统的计算机或者 Windows NT 工作站，100MB 的空闲磁盘空间，最少 32MB 的内存(如果使用 Windows NT，建议准备 64MB 的内存)。

丛书编委会

2003 年 11 月

本书使用说明

本书用来指导初级程序员使用 Java 程序设计语言开发应用程序和 applet(小程序)。本书假定学生没有任何程序设计语言的经验。

学生文件

要完成对本书各章和各章习题的学习，需要从http://cbs.pku.edu.cn的【下载专区】中下载本书的学生文件，包括实例与练习文件。

使用自备的计算机

可以使用自己的计算机完成本书中各章内容和习题的学习。要使用自备的计算机，需要满足下列条件：

- **软件** 可以通过访问 Sun 的网站(http://java.sun.com)下载 JDK 编译程序，下载的软件大约为 20MB。
- **硬件** 必须具备可以运行 Windows 操作系统的计算机或者 Windows NT 工作站，100MB 的空闲磁盘空间，最少 32MB 的内存(如果使用 Windows NT，建议准备 64MB 的内存)。

启动 JDK 编译程序

执行本书中的各个步骤需要具备以下条件：您可以编译保存在各个路径上的*.java 文件，当前的路径是包括当前指南的文件夹的驱动器，例如，A:\Tutorial.06>。要用这样的方式配置 Java 运行，请执行下列步骤:

(1) 转到命令提示符。

(2) 键入 path = drive:\pathname\bin，其中 drive 是包括 JDK 安装程序的驱动器号，pathname 是 JDK 文件夹的完整路径名。例如，安装程序在驱动器 C 上的 JDK1.2 文件夹中，则命令为 path =C:\JDK1.2\bin。

(3) 按 Enter 键。现在可以编译任何文件夹中的 Java 文件。然而，每当启动 Java 时，必须键入路径命令。如果您愿意，则可以将路径命令行添加到计算机的 autoexec.bat 文件，以使 Java 能够自动启动。

目　录

第 1 章　使用 Java 编写第一个程序

案例

当你阅读电子邮件时，突然有一种沉重的感觉袭上心头。电子邮件是这样写的：如果有空，请来我的办公室——Lynn Greenbrier。Lynn Greenbrier 是事件处理程序公司负责程序设计的主管，而你作为实习生只为她工作了两个星期。事件处理程序公司管理私人和公司各方的详细信息；每个客户有不同的需求，因此工作很有趣并且令人兴奋。

“我做错什么事了吗?”进入她的办公室以后你问道，“你准备解雇我吗?”

像所有能猜出别人心思的人一样，Lynn 站起来迎接你并说道：“别那么愁眉苦脸的！我想知道，你是否有兴趣迎接一项新的挑战。我们的程序设计部门准备在接下来的几个月里开发几个新的程序。我们决定采用 Java 程序设计语言。Java 语言是一种面向对象的并且和平台无关的语言，很适合在万维网上应用，而万维网正是我们开拓市场的方向。”

“我不太明白‘面向对象的’和‘与平台无关的’的确切含义，”你说道，“但我一直对计算机技术很感兴趣，我喜欢学习更多的程序设计技能。”

“根据智力测验结果，我认为你很适合程序设计工作，”Lynn 说道，“让我们现在就开始吧。我会向你介绍有关的基础知识。”

1.1　创建一个程序

1.1.1　程序设计

计算机程序只不过是一组由人编写的命令计算机执行操作的指令。计算机是由许多小型的通/断开关线路组成的，因此可以沿着下列线路编写指令进行计算机程序设计：

- 第 1 个开关——通
- 第 2 个开关——断
- 第 3 个开关——断
- 第 4 个开关——通

程序可以对应数千个开关不断执行下去。用这种方式编写的程序称为用机器语言编写的程序，机器语言是最基本的电路层的语言。这种方法的问题在于：在编写有价值的任务时，需要跟踪许多涉及程序设计的开关；如果程序不像期望的那样执行，需要查找错误的开关。另外，开关的数量和位置会因计算机的机型不同而存在差异，这意味着，用户需要根据每种运行程序的机器机型定制机器语言程序。

幸运的是，由于高级程序设计语言的发展，程序设计工作逐步发展成为一种更容易完成的任务。高级程序设计语言允许用户使用诸如“read”、“write”或者“add”等适用的指令代码代替执行这些任务的通断开关序列。高级语言还允许用户把直观的名称(例如，“hoursWorked”或者“rateOfPay”)分配到计算机的存储区域，省得用户非得记住这些值的存储单元(开关编号)不可。

每种高级语言都具有自己特有的语法或者语言规则。例如，根据特定的高级语言，用户可以使用动词“print”或者“write”产生输出。所有的语言均具有特定的并且有限的词汇或一组特殊的有关使用这些词汇的规则。程序员可以使用一种称为编译程序(或者解释程序或者汇编程序)的计算机程序把高级语言语句翻译成为机器代码。每当程序员错误地使用程序设计语言时，编译程序生成错误信息；随后，程序员可以改正代码中的错误并通过编译程序再次进行代码转换。如果你正在学习计算机程序设计语言(例如，Java 程序设计语言、C++或者通用商业语言 COBOL)，实际上学习的内容就是程序设计语言的词汇

和语法规则。

除了学习特殊语言的正确语法以外，程序员还必须懂得计算机程序设计的逻辑。程序中蕴涵的逻辑指的是，按照正确的序列执行各种语句和过程以产生期望的结果。例如，你也许能在乐器上准确地弹出单个音符，但是，如果不能按照正确的次序弹奏连续的音符(或者原来准备弹出 F 半升音调却弹成了 B 降半音)，就不能演奏出令人愉快的曲调。与演奏乐器的情况相似，或许你可以正确地运用计算机语言的语法，却不能执行按照逻辑构造的可工作程序。逻辑错误的实例有：用户原准备把两个数值相除，却执行了乘法运算；或者在取得合适输入值以前产生输出值。

1.1.2 面向对象的程序设计

目前，有两种通用的编写计算机程序的方法，即过程程序设计和面向对象的程序设计。

过程程序设计包括：使用程序设计语言的知识创建存储数值的计算机存储单元，然后编写一系列的处理数值的步骤或操作。由于计算机存储单元中存储的数值会发生变化，因此计算机存储单元又称为变量。例如，为公司编写的工资计算程序可能包括一个名为 rateOfPay 的变量。名称 rateOfPay 引用的存储单元可能包括不同时期的不同值(公司的每名员工所对应的不同数值)。执行工资计算程序的过程中，在名称 rateOfPay 项下保存的每个值可以对 rateOfPay 执行许多操作——例如，从输入设备读取数值、把数值和表示工时的另一个变量相乘、在纸上打印数值。为了方便起见，用于计算机程序的单独操作往往组合到称为过程的逻辑单元中。例如，同时用 4 个或 5 个等系列决定个人的、联邦政府预扣所得税的比较和计算可以被组合成一个名为 calculateFederalWithholding 的过程。过程程序定义可变的存储单元，然后调用一系列过程用来输入、操作和输出保存在存储单元的数值。单个过程程序往往包括上百个变量和上千个过程调用。

面向对象的程序设计是过程程序设计的延伸。在面向对象的程序设计中，用户编写计算机程序的方法与在过程程序设计中的方法略有不同。按照面向对象的方法进行思考时，需要把程序组件想像成和现实世界中具体的对象相似的对象。然后，用户操作对象以实现期望的结果。编写面向对象的程序包括创建对象和创建使用对象的应用程序两个部分。

如果你曾经使用过通过命令行操作系统(如 DOS)操作的计算机，并且也使用过 GUI (Graphical User Interface，图形用户界面；如 Windows)操作的计算机，就会了解过程程序和面向对象程序之间的差异。如果想要把多个文件从软盘移动到硬盘，可以通过在提示符下或命令行中输入命令，或者在图形环境下使用鼠标完成任务。两种方式的差异在于：在命令行方式下，需要按照次序输入一系列命令移动 3 个文件；而在图形环境下，只需要把表示文件的图标从一个屏幕位置拖动到另一个屏幕位置，就像实际工作中把文件从办公室的某个文件柜移动到另一个文件柜中一样。可以使用操作系统移动 3 个文件，但是 GUI 系统可以像操作实际的文件副本一样操作图形环境中的文件。换句话说，GUI 系统可以把文件作为对象进行处理。

现实世界和面向对象的程序设计中的对象都是由状态和方法两部分组成。对象的状态也称为对象的属性。例如，汽车的一部分属性包括制造商、型号、生产日期和购买价格。汽车的其他属性包括，汽车是否在行驶、汽车的传动装置、车速，以及汽车是否弄脏。所有的汽车具有相同的属性，但是毫无疑问的是，这些属性的数值不一定相同。同样，豢养的狗也具有很多属性：品种、名称、年龄和照片是否是当前的。带有凹痕的红色雪佛莱汽车是由所有汽车组成的类的一个实例，名为 Goldie 的金毛猎犬是由所有狗组成的类的一个实例。当把对象作为某一类的实例考虑时，可以把对于某类的一般知识应用到该类的单个成员中。这些对象的特殊实例从一般的分类中继承了属性。如果你的朋友购买了一辆汽车，你知道这辆汽车所具有的型号和名称等属性；如果你的朋友得到一条狗，你知道这条狗所具有的品种等属性。你或许不了解你朋友的汽车的速度或者小狗的照片的确切内容或者当前状态，但是你知道该 Automobile 类和 Dog 类所有的属性。与现实世界类似，在 GUI 操作环境中，由于每个组件作为 GUI 组件的通用类的成员都继承了类的属性，因此用户可以预计每个组件都具有特定的一致的属性，例如，菜单栏和标数

派题栏。

提示　按照惯例，使用 Java 程序语言的程序员都用大写字母作为类名称的开头。因此，定义汽车属性和方法的类可以命名为 Automobile，定义狗的属性和方法的类可以命名为 Dog。然而，生成可工作的程序并不需要遵循该惯例。

除属性以外，对象可以使用方法完成任务。例如，汽车可以向前行驶；也可以倒退行驶。汽车可以加满汽油，也可以用水冲洗，这两种都是改变一些属性的方法。方法可以确定对象的某些属性，例如，汽车的当前速度和汽车油箱的当前状态。类似的，狗可以行走、奔跑、吃食物或者洗澡，有的方法可以确定狗的饥饿程度。GUI 操作系统组件可以最大化、最小化，并且可以拖动。和过程程序一样，面向对象程序也具有变量(属性)和过程(方法)，但是面向对象程序的属性和方法被封装在可以像现实世界对象一样使用的对象中。封装是一项把对象的属性包装在一组紧密结合的单元中的技术，用户把这种结合在一起的单元作为不可分割的实体使用。有时候，程序员把封装技术描述为使用一个“黑匣子”或者使用一种不必考虑其内部机制的装置。

如果对象的方法编写完成并且用户不知道如何执行方法的低层的详细信息，在这种情况下，用户只需懂得方法和对象之间的接口或者交互作用就可以了。例如，如果你能够给汽车加油，那是因为你知道怎么打开加油管口和汽车油箱之间的接口。你既不需要懂得管口的机械工作原理，也不需要了解油箱在车辆内部的实际位置，就能够完成给汽车加油的任务。如果你可以读取里程表，则不必关心显示数据是如何计算出来的。事实上，如果有人生产出更高级的、更精确的测定速度的装置，并把它安装在汽车中，则不必知道也不必关心它的操作方式，只需要知道接口就可以了。相同的原理可以应用于面向对象程序中所使用的构造良好的对象中。

1.1.3　Java 程序设计语言

Java 程序语言是由 Sun Microsystems 公司开发的面向对象的语言，既可以用来开发多用途的商业程序，也可以用来开发基于万维网的交互式的因特网程序。由于具有以下优势，使得 Java 程序设计语言在近年来变得非常流行，一是它具有安全特性，二是它的体系结构是中立性的，即可以使用 Java 程序设计语言编写在任何平台上运行的程序。为了在一些机器上运行 Java 程序设计语言编写的程序，只须用称为解释程序的特殊程序把 Java 程序转换为主机可识别的代码即可。相比之下，当使用其他程序设计语言时，为了使所有用户都可以使用程序，软件供应商通常不得不制做出相同产品的多个版本(DOS 版本、Windows 3.1 版本、Windows 95 版本、Windows 98 版本、Macintosh 版本等)。在使用 Java 程序设计语言的情况下，一个程序版本可以在所有的这些平台上运行。用 Java 程序设计语言编写的程序编译为 Java 虚拟机代码，也称字节码。然后，经过编译的字节码在运行程序的机器上被转换成机器识别的代码。任何已编译的程序均可以在具有 Java 程序语言设计解释程序的机器上运行。

提示　为简单起见，“Java 程序”和“Java 程序设计语言的程序”这两个术语在整本教程中可以交替使用。

Java 程序设计语言的另一个优点是，比起其他许多面向对象的语言来说，Java 程序设计语言更易于使用。Java 程序设计语言是仿照 C++ 程序设计语言开发的。虽然这两种语言在第一次面世时都称不上“易读”或者“易懂”，但 Java 程序语言删除了 C++ 中一些最难懂的特性。

提示　对于 C++ 程序员来说，指针和多继承是两个最难掌握的语言特征。它们也是从 Java 程序语言中排除的难以处理的两项 C++ 特征。

1.1.4　启动一个程序

初看起来，即使最简单的 Java 程序也包含大量的令人困惑的语法。考察以下简单程序。该程序用 7 行语句编写而成，它的任务很简单，就是要在屏幕上打印出“First Java program”。

```
public class First
{
    public static void main(String[] args)
    {
        System.out.println("First Java program");
    }
}
```

在该程序中执行实际工作的语句是 System.out.println("First Javaprogram");。所有的 Java 程序设计语言语句都用分号结束。

文本“First Java program”是由字符组成的文字串；也就是说，它是在输入时准确显示的一连串的字符。Java 程序中的文字串均用双引号括起来。

由于字符串是为方法定义的参数，因此字符串“First Java program”出现在圆括号内。参数包括方法执行任务所需的信息。例如，你和一家销售体育用品的公司签订了目录订单。处理目录订单是包括一组标准过程的方法。然而，每个目录订单均需要信息——例如，订购的产品号、预定的产品的数量——这些信息可以认为是订单的参数。如果从目录中订购了两个产品号是 5432 的产品，那么得到的将不同于订购 1000 个产品号是 9008 的结果。同样，把参数“Happy Holidays”传递给方法得到的结果和把参数“First Java program”传递给方法得到的结果是不相同的。

在语句 System.out.println("First Java program");中，对其传递“First Java program”的方法称为 println( )方法。println( )方法在屏幕上打印输出行，把光标的位置确定在下一行，然后准备其他的输出。

提示 *引用的方法名称后面通常跟有圆括号，如 println( )。因此，可以把方法和变量名称区分开。*

在语句 System.out.println("First Java program");中，out 是一个对象，out 对象表示屏幕。一些方法，包括 println( )在内，在使用时可以带有 out 对象。当然，不是所有的对象都具有 println( )方法(例如，你不能打印到键盘、Automobile 或者 Dog 对象上)，但是 Java 平台的创建者假定你经常需要在屏幕上显示输出。因此，out 对象在创建时赋有称为 println( )的方法。在本小节，将创建自己的对象并给对象赋予自己的方法。

提示 *print( )方法和 println( )方法非常相似。使用 println( )时，在打印完信息后，光标出现在输出信息以后的行上。使用 print( )时，光标不移动到新的行上；光标仍然停留在与输出信息相同的行上。*

在语句 System.out.println("First Java program");中，System 是一个类。 因此，System 定义许多相似的“System”对象的属性，正像 Dog 类定义许多类似的 dog 对象的属性一样。System 的一个对象是 out。(大概可以猜出；System 的另一个对象是 in，它代表的是输入设备。)

提示 *Java 程序语言区分大小写——例如，称为 System 的类和称为 system、SYSTEM、sYsTeM 的类是完全不同的。*

语句 System.out.println("First Java program"); 中的圆点“.”是用来分隔类、对象和方法的名称的。可以在 Java 程序中反复地使用这种格式：类.对象.方法。

打印字符串“First Java program”的语句嵌入在图 1.1 中所示的程序中。

```
public class First
{
    public static void main(String[] args)
    {
        System.out.println("First Java program");
    }
}
```

图 1.1 打印字符串

在 Java 程序中，使用的一切对象必须属于某个类。在编写程序 public class First 时，其实是在定义名称为 First 的类。可以使用你需要的任何名称或者标识符定义 Java 程序设计语言的类，名称或者标识符必须符合以下要求：

- 类的名称必须以字母表中的字符(包括非英语的字母，例如，α 或者 π)、下划线(_)或者美元符号($)开头。
- 类的名称可以只包括字母、数字、下划线或者美元符号。
- 类的名称不能是 Java 程序设计语言保留关键字，例如，public 或者 class(有关保留关键字的列表，请参见图 1.2)。
- 类的名称不能为下列数值的其中一种：true，false 或者 null。

提示　Java 程序设计语言是一种基于 Unicode(统一的字符编码标准，一种字符表示的国际分类方法)的语言。术语“字母”指英语字母，以及阿拉伯语、希腊语及其他语言的字母。有关 Unicode 的更详细的信息，请参见本章的 1.2 节。

abstract	float	private
boolean	for	protected
break	future	public
byte	generic	rest
byvalue	goto	return
case	if	short
cast	implements	static
catch	import	super
char	inner	switch
class	instanceof	synchronized
const	int	this
continue	interface	throw
default	long	throws
do	native	transient
double	new	try
else	null	var
extends	operator	void
final	outer	volatile
finally	package	while

图 1.2　Java 程序语言保留关键字

用大写字母作为类的名称的开头，根据需要使用其他大写字母提高可读性，这是 Java 程序设计语言的标准。

图 1.3 列出了 Java 程序设计语言的一些有效的和惯用的类名称。

类名称	说明
Employee	以大写字母开头
UnderGradStudent	以大写字母开头，不包含空格，用词首的大写字母强调每个单词
InventoryItem	以大写字母开头，不包含空格，第二个单词的词首大写，以示强调
Budget2001	用大写字母开头，不包含空格

图 1.3　Java 程序设计语言中的一些有效的类名称

图 1.4 列举一些有效的但是非惯用的类名称。

提示　应该遵循确定的 Java 程序设计语言的约定编写程序，使编写的程序便于其他程序员理解并能继续编写。本书使用已确定的 Java 程序设计约定。

类名称	说明
employee	以小写字母开头
Undergradstudent	新单词的起始字母不用大写字母表示；难以阅读
Inventory_Item	下划线通常不用于表示新单词
BUDGET2001	全部用大写字母表示

图 1.4　Java 程序设计语言中一些非惯用的类名称

图 1.5 列举了一些非法的类名称。

类名称	说明
an employee	空格字符是非法的
Inventory Item	空格字符是非法的
class	class 是保留字
2001Budget	类名称不能以数字开头
phone#	#符号是不容许使用的

图 1.5 Java 程序设计语言中一些非法的类名称

在图 1.1 中，代码行 public class First 包括关键字 class，该关键字把 First 标识为一个类。关键字 public 是访问修饰符。访问修饰符用来定义可以访问类的环境。公有的访问是最自由的访问方式；在第 2 章中将学习公有访问及其他类型的访问方式。

所有类的内容密封在花括号内({和})。一个类可以包含任意数量的数据项和方法。在图 1.1 中，类 First 只在花括号内包括一种方法。方法的名称是 main( )，并且 main( )方法包含自己的一组花括号且只包括一个语句——println( )语句。

提示 一般说来，在 Java 程序设计语言中，空白是可选的。空白是空格、跳格设定和回车符(空白行)的任意组合。然而，不能在标识符或者关键字中使用空白。由于编译程序可以忽略附加的空格，可以通过输入空格、跳格设定或者空白行在代码或者行之间插入空白。你可以使用空白编排程序代码，使得程序代码更易阅读。

在 Java 程序中，每个开花括号({)一定对应于一个闭花括号(})。开花括号和闭花括号的位置对于编译程序来说并不重要。例如，以下方法的执行方式和图 1.1 中所示方法的执行方式几乎完全相同。唯一的区别在于，方法的组织方式不同。通常情况下，在每对开花括号和闭花括号中竖向对准的代码更易阅读。应该尽量使输入的代码容易阅读。

```
public static void main(String[] args){
System.out.println("First Java program");}
```

main( )方法的方法头部相当复杂。在学习完整本书以后，将会对方法头部中所使用的每个术语的含义和用途有更清楚的认识；现在看一下有关的简要介绍就足够了。

在方法头部 public static void main(String[ ] args)中，和定义 First 类时一样，字 public 是访问修饰符。在英文中，单词 static 的含义是出现很小的变化或者静止不变。在 Java 程序设计语言中，保留关键字 static 的含义也是“不变的”，并且表示为 First 类创建的每个成员具有相同的、不变的 main( )方法。Java 程序设计语言中，static 还意味着唯一性。只有一个 First 类的 main( )方法将保存在计算机的内存中。当然，其他类在最终也将有自己的不同的 main( )方法。

在英文中，单词 void 的含义是“空的”。当关键字 void 用于 main( )方法头部中时，并不表示 main( )方法是空的，而是表示 main( )方法在被调用时不返回任何数值。这并不表示 main( )不产生输出——实际上，main( )方法的确产生输出。main( )方法并不把任何数值返回给可能使用它的任何其他方法。第 2 章将会学习更多有关返回值的知识。

所有的 Java 应用程序都必须包括命名为 main( )的方法，大多数的 Java 应用程序还都具有附加的方法。当执行 Java 应用程序时，编译程序总是首先执行 main( )方法。

在方法头部 public static void main(String[ ] args)中，已经认识到，正像字符串“First Java program”传递给 println( )方法的参数一样，括号之间的内容(String[ ] args)一定表示传递给 main( )方法的参数，String 表示可用于表示字符串的 Java 类。标识符 args 用来存储可能发送给 main( )方法的字符串。main( )方法可以通过这些参数执行某项任务(如打印参数)，但是在图 1.1 中，main( )方法实际上不使用 args 标识符。尽管如此，必须把标识符放置在 main( )方法的括号内。标识符不一定命名为 args——它可以是任何有效的 Java 标识符——但是名称 args 是传统上使用的名称。

提示　当引用 main( )方法头部中的 String 类时，方括号[ ]表示一组 String[ ]对象。在第 5 章将讲到更多有关数组和 String 类的知识。

图 1.1 中的简单程序具有许多需要记忆的部分。然而，可以暂时把图 1.6 中所示的程序作为 shell 程序使用，在这个 shell 程序中可以用想要执行的语句替换代码行/ * * * * * * /。

```
public class First
{
   public static void main(String[] args)
   {
      /* * * * * */
   }
}
```

图 1.6　shell 输出程序

在了解了用 Java 程序设计语言编写程序的基本框架后，可以准备把你的第一个 Java 程序输入到文本编辑器中。程序员之间保留了这么一项传统，就是用任何语言编写的第一个程序输出为“Hello, world!”。现在，你就可以编写这样一个程序了。可以使用文本编辑器(例如，WordPad 或者 Notepad)或者其他字处理器进行编写。

编写第一个 Java 程序的步骤：

(1) 启动文本编辑器(例如，Notepad、WordPad 或者其他字处理器)，如有必要，再新建一个文档。(Notepad 是最容易使用的编写程序的文本编辑器。)

(2) 输入类头部“public class Hello”。在本例中，类的名称是 Hello。可以为类定义任何需要的有效名称。如果选择 Hello 作为类的名称，由于 Java 程序设计语言是区分大小写的，则必须把该类称为 Hello，而不能称为 hello。

(3) 按 Enter 键后，输入“{”，然后再次按 Enter 键，然后输入“}”。这样，就会把 main( )方法添加到花括号中。虽然这种操作不是必需的，但是让每个花括号自己占一行，却是一种很好的编程习惯。这种编程习惯可以使编写出的代码更易阅读。

(4) 如图 1.7 所示，在花括号之间添加 main( )方法头部，然后为 main( )输入一组花括号。

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

图 1.7　Hello 类的 main( )方法 shell 程序

接下来，在 main( )方法的括弧内添加产生输出“Hello，world!”的语句。

(5) 使用图 1.8 把 println( )语句添加到 main( )方法中。

```
public class Hello
{
   public static void main(String[] args)
   {
      System.out.println("Hello, world! ");
   }
}
```

图 1.8　Hello 类的完整的 main( )方法

(6) 把程序保存到你想要放进去的文件夹中，程序命名为 Hello.java。请注意，文件扩展名必须是 .java。如果不是这样，Java 程序设计语言的编译程序将不能识别该程序。

帮助 许多文本编辑器把它们自己的文件扩展名(例如.txt 或者.doc)附加到保存文件上。请仔细检查保存文件，确保保存文件不具有双重扩展名(如 Hello.java.txt)。如果文件具有双重的扩展名，请对文件重新命名。如果你明确地输入引号把文件名括起来(如“Hello.java”)，大多数的编辑器将把文件保存为你所指定的文件名，而不会添加编辑器自己的扩展名。确认你把 .java 文件作为文本文档保存。默认情况下， Notepad 把所有的文件作为文本保存。

1.1.5 给程序添加注释

正如你所看到的，即使最简单的 Java 程序也要用数行代码编写，并且多少包含一些复杂的语法。执行多项任务的大型程序包括的代码数就更多了。编写的程序越长，要记住采取一些步骤的原因或者使用特殊变量的方式，就变得越来越困难。程序注释是为了文档编制而添加到程序中的非执行语句。程序员使用程序注释为自己和有可能在将来阅读这个程序的其他人作出提示。至少，你编写的程序应该包括标明程序作者、编写日期和程序名称或功能的注解。

提示 建议在通读本书时，在每个程序的前 3 行添加程序注释。程序注释应该包括程序名称、你的姓名和编写程序的日期。教师可能要求你添加补充的注释。

在开发程序时，程序注释也能发挥有效的作用。如果程序不像期望的那样执行，可以将名称语句分别设为注释，然后运行程序并观察效果。在将某条语句设为注释后，编译程序将不能执行语句中的命令。这样有助于精确地确定错误语句在故障程序中的位置。

Java 程序语言中包括 3 种类型的注释：

- 行注释以两条斜杠(//)开头，直到当前行的末尾。行注释可以单独地出现在一行上或者出现在可执行的程序代码所在的行的结尾。
- 块注释以一个斜杠和一个星号(/*)开始，以一个星号和一个斜杠(*/)结束。块注释可以单独地显示在行上、一行中可执行的程序代码的前面、一行中可执行的程序代码的后面。根据需要，块注释还可以延伸跨过许多行。
- 块注释的特例是 javadoc 注释。这种注释以一个斜杠和两个星号(/**)开始，以一个星号和一个斜杠(*/)结束。可以使用 javadoc 注释生成带有名称为 javadoc 的程序文件。

提示 斜杠(/)和反斜杠(\)字符往往容易搞混，但是它们是两个完全不同的字符。它们不能彼此替代。

提示 Java 开发工具包(JDK)包括 javadoc 工具，javadoc 工具包括用 Java 程序设计语言编写程序时可以使用的类。

图 1.9 说明如何在代码中使用注释。

```
// Demonstrating comments
/* This shows
   that these comments
   don't matter */
{
   System.out.println("Hello"); // This line executes
         // up to where the comment started
   /** Everything but the println( )line
   is a comment. */
```

图 1.9 在程序中使用注释

接下来，把注释添加到 Hello.java 程序中。

把注释添加到程序中的步骤：

(1) 把光标移到文件的顶部，按 Enter 键插入新行，按下 Up 键转到该行，然后在文件的顶部输入下