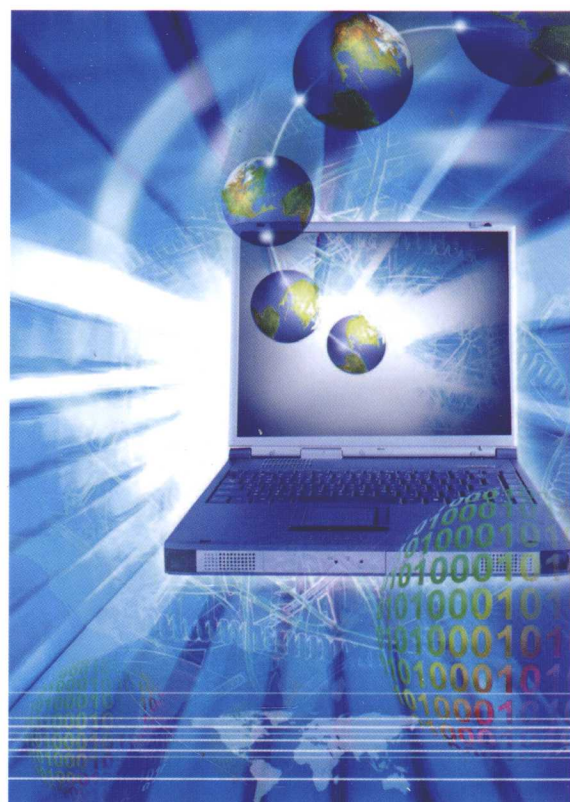


Java简明教程

- ◆ 程序基础
- ◆ 结构化程序设计→方法
- ◆ 面向对象程序设计→类
- ◆ 高级编程



林巧民 主编

马子超 何良 毛金锋 副主编



清华大学出版社

013032769

TP312JA
1478

高等学校计算机应用规划教材

Java 简明教程

林巧民 主 编

马子超 何 良 毛金锋 副主编



清华大学出版社

北 京



北航

C1640814

TP312JA
1478

内 容 简 介

本书以 Java 为描述语言, 详细介绍计算机语言的结构化编程和面向对象编程。全书共分 12 章, 主要内容包括: Java 入门、Java 编程基础、Java 程序基本结构、方法和数组、类和对象、继承、多态与接口、字符串、多线程与 Applet 技术、图形用户界面、Java I/O、Java 游戏开发基础以及游戏开发实例等。如果说结构化编程的特征是方法, 那么面向对象编程的体现就是类的设计和使用, 全书对这两种不同的设计思想都作了充分介绍。此外, 每章最后都配有思考练习, 习题有选择题、填空题、简答题、编程题等多种类型, 选择题、填空题和简答题有助于读者对所学知识的理解与掌握, 编程题则可以提高读者的动手实践能力。

本书结构清晰、内容翔实, 既可以作为高等院校相关专业的教材, 也可作为从事软件开发工作的专业技术人员参考书。

本书对应的电子教案、实例源代码和习题答案可以到 <http://www.tupwk.com.cn/downpage/index.asp> 网站下载。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Java 简明教程/林巧民 主编. —北京: 清华大学出版社, 2013.3

(高等学校计算机应用规划教材)

ISBN 978-7-302-31477-6

I. ①J… II. ①林… III. ①Java 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 023860 号

责任编辑: 胡辰浩 袁建华

装帧设计: 牛静敏

责任校对: 成凤进

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62796045

印 装 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 22.5

字 数: 519 千字

版 次: 2013 年 3 月第 1 版

印 次: 2013 年 3 月第 1 次印刷

印 数: 1~5000

定 价: 36.00 元

前 言

Java 语言自从面世至今一直受到大学生和广大软件研发人员的青睐。目前,许多高校已改变先讲授 Pascal 语言或 C 语言,再让学生选修 Java 语言的惯例,开始让学生在大学低年级就学习 Java 语言。还有不少高校甚至对非计算机专业的大一新生也开设了 Java 课程。但目前,市面上大多数 Java 教程在讲述面向对象技术时几乎都忽视了对 Java 语言基础的介绍,片面追求技术的新、奇、特,无法满足编程初学者的入门需要。

本书旨在突破市面上大多数 Java 教材的局限,尝试用一种语言来充分阐述两种编程思想,即结构化程序设计和面向对象程序设计,以满足普通初学者的需要。事实上,结构化程序设计是面向对象程序设计的基础,面向对象程序的基本组成还是结构化程序。面向对象程序设计引入了类的概念,使得编程人员可以站在设计类(而不是方法)的高度,对程序进行设计和实现,同时必须重视结构化程序设计基本功的锻炼,因为类的设计恰恰是建立在结构化设计的基础之上的。因此,本书以 Java 语言为工具,从结构化程序设计和面向对象程序设计两种不同编程思想的角度,分别对 Java 编程的相关基础知识予以介绍,希望能对广大编程爱好者尤其是初学者有所裨益。

全书共分 12 章,各章主要内容如下。

第 1 章是 Java 入门,简要介绍 Java 的诞生、Java 语言的特点、Java 开发工具以及具体的开发步骤等。

第 2 章是 Java 编程基础,主要介绍 Java 的基本数据类型、赋值语句、条件表达式、运算以及复合语句等。

第 3 章是 Java 程序基本结构,详细介绍程序的 3 种基本流程结构:顺序结构、分支结构和循环结构。

第 4 章是方法和数组,主要介绍方法的概念和定义、方法的调用、变量的作用域、数组以及数组与方法的关系等。

第 5 章是类和对象,详细介绍类的概念和定义、对象的创建与使用、访问控制符和包等。

第 6 章是继承、多态与接口,详细介绍了继承与多态技术、抽象类和接口等知识。

第 7 章是字符串,主要介绍 Java 提供的 String 和 StringBuffer 类。

第 8 章是多线程与 Applet 技术,详细介绍线程的概念、创建、生命周期及状态、线程同步、优先级和调度等;还介绍了 Applet 的概念和原理、基本开发技术以及多媒体编程等。

第 9 章是图形用户界面,详细介绍 AWT 组件集中的常用组件,包括容器类组件、布局类组件、普通组件以及事件处理机制等。此外,本章还简要介绍了 Swing 组件集。

第 10 章是 Java 输入输出,详细介绍 Java 输入输出流的概念、字节流类、字符流类、File 类以及 RandomAccessFile 类等。

第 11 章是 Java 游戏开发基础,介绍游戏编程的相关基本知识,包括图形环境的坐标

体系、图形图像的绘制、各种坐标变换、动画的生成和动画闪烁的消除等。

第 12 章是游戏开发实例，以星球大战这款游戏为例介绍 Java 游戏的开发过程，同时还介绍独立应用程序和小应用程序两种不同形式的游戏开发。

本书在编写过程中力求做到概念清楚、由浅入深、通俗易懂、论述详尽、实例丰富以方便读者自学。全书内容具有较强的实用性。

本书由林巧民、马子超、何良和毛金锋共同编著，林巧民任主编。南京邮电大学秦军教授、吴伟敏副教授等老师在百忙之中，认真细致地审阅了全部书稿，提出了许多有益的修改意见。此外，研究生林萍、郑新资以及陈志军、王利贤、李廷欣、何怀怡、范瑜、金龙、应显奇、林文恭、朱丽珍、杨珏梅、田斌、陈映钊和赵臻等同学也为本书的输入、校对等付出了辛勤的劳动。在此一并表示由衷的感谢。在编写本书的过程中参考了相关文献，在此向这些文献的作者深表感谢。

由于作者水平所限，书中难免会有不足之处，敬请广大同行和读者给予批评和指正。我们的邮箱是 huchenhao@263.net，电话是 010-62796045。

作 者

2012 年 10 月

目

录

第 1 章 Java 入门	1
1.1 概述	1
1.1.1 Java 的诞生	2
1.1.2 Java 的特点	3
1.1.3 Java 与其他编程语言间的关系	5
1.2 第一个 Java 程序	6
1.3 Java 开发工具	7
1.4 Java 程序开发步骤	9
1.4.1 软件安装	10
1.4.2 环境变量配置	10
1.4.3 编译运行	12
1.5 小结	15
1.6 思考练习	15
第 2 章 Java 编程基础	16
2.1 引言	16
2.1.1 符号	16
2.1.2 分隔符	17
2.1.3 常量	18
2.1.4 变量	19
2.1.5 final 变量	20
2.1.6 变量类型转换	20
2.2 基本数据类型	21
2.2.1 布尔型	21
2.2.2 整型	22
2.2.3 浮点型	24
2.2.4 字符型	25
2.3 程序语句	25
2.3.1 赋值语句	25
2.3.2 条件表达式	27
2.3.3 运算	28

2.3.4 复合语句	29
2.4 小结	31
2.5 思考练习	31
第 3 章 Java 程序基本结构	32
3.1 复合语句	32
3.2 顺序结构	33
3.3 分支结构	37
3.3.1 单分支条件语句	37
3.3.2 双分支条件语句	42
3.3.3 分支结构嵌套	45
3.3.4 switch 语句	51
3.4 循环结构	54
3.4.1 while 语句	55
3.4.2 do-while 语句	59
3.4.3 for 语句	60
3.4.4 循环嵌套	62
3.4.5 跳转语句	63
3.5 小结	67
3.6 思考练习	67
第 4 章 方法与数组	71
4.1 方法的概念和定义	71
4.2 方法的调用	74
4.2.1 调用方式	74
4.2.2 参数传递	77
4.2.3 返回值	78
4.2.4 方法嵌套及递归	79
4.3 变量作用域	85
4.4 数组	86
4.4.1 数组的概念	86
4.4.2 数组的声明和创建	87
4.4.3 数组的应用举例	90

4.5 数组与方法	95	6.3.1 final 关键字	143
4.6 小结	96	6.3.2 实例成员和类成员	144
4.7 思考练习	97	6.3.3 类 java.lang.Object	148
第 5 章 类和对象	100	6.4 小结	150
5.1 引言	100	6.5 思考练习	150
5.2 类	102	第 7 章 字符串	152
5.2.1 类声明	103	7.1 字符串的创建	152
5.2.2 类体	104	7.1.1 创建 String 类的字符串	152
5.2.3 成员变量	105	7.1.2 创建 StringBuffer 类的 字符串	153
5.2.4 成员方法	106	7.2 String 类型字符串的操作	154
5.2.5 方法重载	109	7.3 StringBuffer 类型字符串 的操作	163
5.2.6 构造方法	111	7.3.1 字符串操作	163
5.2.7 main()方法	111	7.3.2 字符分析器	168
5.2.8 finalize()方法	112	7.3.3 main()方法	169
5.3 对象	112	7.4 小结	170
5.3.1 对象的创建	113	7.5 思考练习	170
5.3.2 对象的使用	114	第 8 章 多线程与 Applet 技术	173
5.3.3 对象的清除	117	8.1 多线程	173
5.4 访问控制符	117	8.2 多线程的创建	174
5.4.1 类的访问控制符	118	8.2.1 Thread 子类创建线程	174
5.4.2 对类成员的访问控制	118	8.2.2 使用 Runnable 接口	176
5.5 包	122	8.3 线程的生命期及其状态	177
5.5.1 包的创建	123	8.3.1 线程的状态	177
5.5.2 import 语句	124	8.3.2 与线程状态有关的 Thread 类方法	178
5.5.3 编译和运行包	125	8.4 线程的同步	183
5.6 小结	126	8.5 线程的优先级和调度	188
5.7 思考练习	126	8.5.1 线程的优先级	188
第 6 章 继承、多态与接口	129	8.5.2 线程的调度	188
6.1 继承与多态	129	8.6 守护线程	189
6.1.1 子类、父类与继承机制	129	8.7 线程组	191
6.1.2 Java 的继承	130	8.8 Applet 概述	193
6.1.3 多态性	134	8.9 Applet 开发技术	194
6.2 抽象类和接口	138	8.9.1 Applet 开发步骤	194
6.2.1 抽象类	138		
6.2.2 接口	140		
6.3 其他	143		

8.9.2 Applet 技术解析.....	196	第 11 章 Java 游戏开发基础.....	293
8.10 Applet 多媒体编程.....	201	11.1 概述.....	293
8.10.1 文字.....	201	11.2 绘制 2D 图形图像.....	293
8.10.2 图形.....	202	11.2.1 坐标体系.....	293
8.10.3 图像.....	206	11.2.2 绘制图形.....	294
8.10.4 声音.....	207	11.2.3 绘制图像.....	296
8.10.5 动画.....	207	11.3 图形图像的坐标变换.....	299
8.11 小结.....	212	11.3.1 使用 Graphics2D 类进行 坐标变换.....	299
8.12 思考练习.....	213	11.3.2 使用 AffineTransform 类 进行坐标变换.....	302
第 9 章 图形用户界面.....	215	11.4 生成动画.....	306
9.1 概述.....	215	11.5 消除动画闪烁.....	309
9.2 AWT 组件集.....	216	11.6 小结.....	312
9.2.1 容器类组件.....	217	11.7 思考练习.....	312
9.2.2 布局组件类.....	217	第 12 章 游戏开发实例.....	313
9.2.3 普通组件.....	227	12.1 游戏总体介绍.....	313
9.2.4 事件处理.....	239	12.2 游戏辅助类.....	321
9.3 Swing 组件集简介.....	250	12.2.1 Point2D 类.....	321
9.4 小结.....	256	12.2.2 SpriteImage 类.....	322
9.5 思考练习.....	256	12.2.3 AnimatedSprite 类.....	324
第 10 章 Java I/O.....	258	12.3 完善 StarWars.java.....	332
10.1 引言.....	258	12.3.1 Sprite 的初始化.....	332
10.2 流的概念.....	258	12.3.2 键盘事件处理.....	334
10.2.1 标准输入.....	259	12.3.3 更新 Sprites.....	337
10.2.2 标准输出.....	261	12.3.4 碰撞检测.....	338
10.3 字节流.....	266	12.3.5 删除与绘制 Sprite.....	340
10.3.1 InputStream.....	266	12.3.6 完整的 StarWars 类.....	340
10.3.2 OutputStream.....	272	12.4 Applet 游戏开发与部署.....	342
10.4 字符流.....	276	12.5 小结.....	345
10.4.1 Reader.....	276	12.6 思考练习.....	346
10.4.2 Writer.....	280	附录 ASCII 码表.....	347
10.5 文件.....	286	参考文献.....	350
10.5.1 File 类.....	286		
10.5.2 RandomAccessFile 类.....	288		
10.6 小结.....	291		
10.7 思考练习.....	291		

第1章 Java入门

本章学习目标:

- 了解 Java 语言的历史和特点
- 理解 Java 与其他编程语言的关系
- 掌握 Java 程序的基本构成
- 了解流行的 Java 程序集成开发环境
- 掌握 Java Application 的一般开发步骤

1.1 概 述

Java 是由美国 Sun 公司(现已被 Oracle 公司收购)开发的支持面向对象程序设计的计算机语言。它最大的优势就是借助于虚拟机机制实现的跨平台特性,实现所谓的“一次编译,随处运行”,使移植工作变得不再复杂。也正因为此,Java 迅速流行起来,成为一种深受广大开发者喜欢的编程语言。目前,随着 Java ME、Java SE 和 Java EE 的发展,Java 已经不仅仅是一门简单的计算机开发语言了,它已经拓展出了一系列的业界先进技术。

Microsoft、IBM、DEC、Adobe、SiliconGraphics、HP、Toshiba、Netscape 和 Apple 等大公司均已购买 Java 的许可证,Microsoft 还在其 IE 浏览器中增加了对 Java 的支持。另外,众多的软件开发商也开发了许多支持 Java 的软件产品,如美国 Borland 公司的 JBuilder,蓝色巨人 IBM 的 Eclipse 和 Visual Age for Java, Sun 公司的 NetBeans 与 Sun Java Studio 5 以及 BEA 公司的 WebLogic Workshop 等。数据库厂商 Sybase 也在开发支持 HTML 和 Java 的 CGI(Common Gateway Interface),Oracle 公司甚至将自己的数据库产品用 Java 来进行开发。Intranet 正在成为企业信息系统的最佳解决方案,它具有便宜、易于使用和管理等优点,用户不管使用何种类型的机器和操作系统,界面都是统一的 Web 浏览器,而数据库、Web 页面(HTML 和用 Java 编的 JSP、Servlet 等)、中间件(Java Bean 或 Enterprise Java Bean 等)则存在 WWW 和应用服务器上。开发人员只需维护一个软件版本,管理人员也省去了为用户安装、升级客户端以及培训人员之繁琐,用户则只需一个操作系统和一个 Internet 浏览器,即采用 B/S(浏览器/服务器)模式,B/S 与 C/S(客户/服务器)模式的显著不同之处在于其是“瘦客户端”的,即程序运行对客户端的要求降至很低的水平,一般将 C/S 模式开发的软件称为两层架构的,而 B/S 模式的软件为三层(或多层)架构的,Java EE 系列技术就是致力于帮助客户构建多层架构的应用。Java ME、Java SE 和 Java EE 的侧重点各有不同,现将其列举如下。

- Java ME(Java Micro Edition)是 Java 的微型版, 常用于嵌入式设备及消费类电器(如手机等)上的开发。
- Java SE(Java Standard Edition)是 Java 的标准版, 用于针对普通 PC 的标准应用程序开发, 现已改名为 Java SE。
- Java EE(Java Enterprise Edition)是 Java 的企业版, 用来针对企业级应用服务的开发。

Java ME、Java SE、Java EE 是 Java 针对不同的应用而提供的不同服务, 即提供不同类型的类库。初学者一般可从 Java SE 入手学习 Java 语言。Java SE 是一个优秀的开发环境, 开发者可以基于这一环境创建功能丰富的交互式应用, 并且可以把这些应用配置到其他平台上。Java SE 1.4 版本具有 GUI 控制功能、快速的 Java 图形功能、支持国际化与本地化扩展以及新的配置选项, 并对 Windows XP 提供扩展支持。本书后面的内容以 1.4 版本为例(虽然该版本有点老, 但用于初学者的学习已经足够)。此外, Java SE 是多种不同风格软件的开发基础, 包括客户端 Java 小程序和应用程序, 以及独立的服务器应用程序等, 同时 Java SE 也是 Java ME 和 Java EE 的基础。事实上, 大部分非企业级软件还是在 Java SE 上开发部署的比较多。首先, 这是因为很多的应用软件都是在 Java SE 上开发的; 其次, Java SE 和 Java EE 是兼容的, 企业版是在标准版上的扩充, 在 Java SE 的版本上开发的软件, 拿到企业版的平台上是一样可以运行的; 再次, 通常的手机及嵌入式设备的应用开发还是在 Java SE 的环境中完成的, 因为毕竟 Java ME 提供的只是微型版的一个环境, 而人们完全可以在 Java SE 上将这个环境虚拟出来, 然后再将开发出来的应用软件拿到微型版的实际环境中去运行。

1.1.1 Java 的诞生

早在 1990 年 12 月, SUN 公司就由 Patrick Naughton、Mike Sheridan 和 James Gosling 成立了一个叫做 Green Team 的小组。该小组的主要目标是要发展一种分散式系统架构, 使其能在消费性电子产品作业平台上执行, 如 PDA、手机、资讯家电(IA, 全称 Internet/Information Appliance)等。1992 年 9 月 3 日, Green Team 发表了一款名为 Star Seven(*7)的机器, 它有点像现在人们熟悉的 PDA(个人数字助理), 不过它有着比 PDA 更强大的功能, 如无线通信(Wireless Network)、5 寸彩色的 LCD、PCMCIA 界面等。

Java 语言的前身 Oak 就是在那个时候诞生的, 其主要的目的当然是用来撰写在 Star 7 上的应用程序。为什么叫 Oak 呢? 原因是 James Gosling 办公室的窗外, 正好有一棵橡树(Oak), 顺手就取了这么个名字。Java 所提供的一些特性, 在 Oak 中就已经具备了, 像安全性、网络通信、面向对象、垃圾收集(Garbage Collected)、多线程等。Oak 是一个相当优秀的程序语言。至于为什么 Oak 会改名为 Java 呢? 这是因为当时 Oak 要去注册商标时, 发现已经有另外一家公司先用了 Oak 这个名字。既然 Oak 这个名字不能用, 那要取啥新名字呢? 工程师们边喝着咖啡讨论着, 看看手上的咖啡, 突然灵机一动, 就叫 Java 好了。就这样, 它就变成了业界所熟知的 Java 了。

在 1995 年 5 月 23 日, JDK(Java Development Kits) 1.0 版本正式对外发表, 它标志着 Java 的正式诞生。2011 年 7 月 28 日, Oracle 正式发布 Java 7(JDK 1.7)。它是 Java 的最新

发行版，其中包含许多新功能、增强功能和 bug 修复，从而提高了开发和运行 Java 程序的效率。

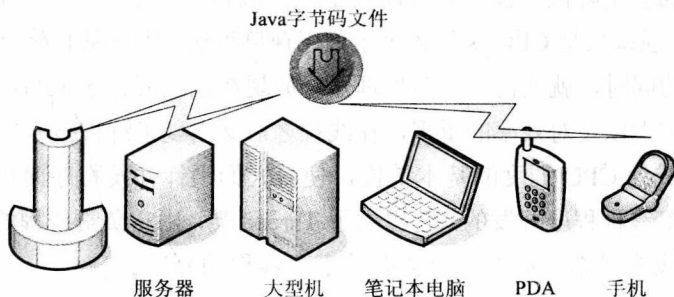
1.1.2 Java 的特点

Java 之所以流行，和它的优秀特性是分不开的。

1. 平台独立性

平台独立性意味着 Java 可以在支持 Java 的任何平台上“独立于所有其他软硬件”而运行。例如，不管操作系统是 Windows、Linux、Unix 还是 Macintosh，也不管机器是大型机，小型机还是微机，甚至是 PDA 或者手机、智能家电，Java 程序都能运行，当然在这些平台上都应装有相应版本的 JVM(Java 虚拟机)，即平台必须支持 Java。

现在很多的手机都是支持 Java 的，大多数手机游戏也都是用 Java 开发的，这样任何支持 Java 的手机都能玩这些游戏，这是平台独立所带来的好处，如图 1-1 所示。



平台独立保证了软件的可移植性，而软件的可移植性是软件投资在未来的保证。用 Java 开发的软件保证了程序在将来无须再移植。可移植性一直是业界所宣扬的最大卖点和亮点，但以前从未实现过，是 Java 使软件行业真正实现了软件的可移植性。

2. 安全性

现今的 Java 语言主要用于网络应用程序的开发，因此对安全性有很高的要求。如果没有安全保证，那么用户运行从网络上下载的 Java 语言应用程序是十分危险的。Java 语言通过一系列的安全措施，在很大程度上避免了病毒程序的产生和网络程序对本地系统的破坏，具体体现如下。

- (1) 去除指针这种数据类型，简化了编程，更是避免了对内存的非法访问。
- (2) Java 是一种强类型的程序设计语言，要求显式的声明，保证编译器可以提前发现程序错误，提高程序的可靠性。
- (3) 垃圾自动回收机制，让程序员从繁琐的内存管理工作中解脱出来，专注于程序开发。更重要的是，通过这种内存自动回收机制，可以很好地确保内存管理的正确性，避免出现“内存泄露”现象。
- (4) Java 语言提供了异常处理机制。

(5) Java 程序在运行时,解释器会对其进行数组和字符串等的越界检查,确保程序的安全。

3. 多线程

在 DOS 时代,人们一次只能运行一个程序,执行完才能运行另一个。后来出现了视窗 Windows 之后,人们可以同时运行几个程序,并可以在各个运行程序之间切换,如一边听音乐一边编辑 Word 文档。这时的操作系统出现了进程的概念,每个运行中的程序都是一个进程。再后来,为了提高程序的并发性,又引入了线程的概念,线程也称为轻量级进程。进程是系统分配资源的基本单位,而线程则是系统 CPU 调度执行的基本单位,一个进程可以只有一个线程,也可以有多个线程。在很多情况下,开发多线程的程序还是很有必要的。例如,在早期单线程进程时代,安装软件开始安装后,就只能一路安装下去了,而现在的软件安装程序一般都提供了“取消”操作,允许安装者在安装过程中的任意时刻取消安装,这也是软件安装程序“多线程”的一个表现。

多线程的目的就是降低总程序的执行粒度,让子程序们“同时”并发执行,这里的“同时”加引号是为了强调只是 CPU 执行各个子程序速度很快,从宏观上看,像是同时在执行。如果要实现真正的同时,就要借助于多处理器,如现在已经流行起来的双核 CPU。另外,随着程序规模的扩大以及对效率的重视,在线程之后又出现了纤程技术。纤程对线程又做了进一步细分,成为 CPU 调度的基本单位,使得人们设计并发程序时更加灵活。

Java 是支持多线程程序开发的,它提供了 Thread 类,由它负责线程的启动运行、终止线程,并可测试线程状态。后面章节会有关于多线程的介绍。

4. 网络化

在网络环境中,对象可以在本地或远程机器上执行。Java 程序可以通过网络打开和访问对象,就像访问本地系统一样简单。Java 语言提供的丰富类库保证了其可以在 HTTP、FTP 和 TCP/IP 协议中良好运行。Java Applet 程序需要客户端浏览器的支持,并且通过标签对 `<applet></applet>` 将程序嵌入 HTML 文件中。当用户浏览该 Web 页时,Java Applet 程序才被从服务器端下载到客户端解释执行。因此也称 Java Applet 是可移动代码,这种移动性为分布式程序开发提供了一种新的技术途径。关于 Java Applet,后面会有详细介绍。

5. 面向对象

随着软件业的发展,面向对象的程序设计方法已经流行起来,出现了很多面向对象的程序设计语言,如 Java、C++、SmallTalk 等。现在用面向对象的编程语言进行软件开发已很普遍。简单说,面向对象主要是通过引入类,使得原本的面向过程程序设计有了质的飞跃。类中不仅包含数据部分,而且还包含操作方法。这个囊括了数据和算法的类成为面向对象程序设计中最关键的要素。可以说,所有功能的实现都是围绕类而展开的。同样,面向对象技术的特征也是由类体现出来的。面向对象最主要的 3 大特征如下。

(1) 封闭性

类定义的一般形式如下:

```
class Name
```

```
{
```

```
    细节
```

```
};
```

其中的细节被以类的形式封装起来了。该细节就是类的成员方法。它可以是数据，也可以是操作这些数据的方法(在面向过程设计中称为函数)。当这些数据和方法的访问权限被设置为私有后，它们就不能被对象从外部进行访问，就像是被隐藏起来了，而对外部只暴露那些访问权限被设置为公有的成员。

(2) 继承性

类是可继承的，就像遗产一样，这可以大大提高程序的复用性，提高程序的开发效率，同时也能降低系统复杂性，提升代码的可读性。

(3) 多态性

多态性也是面向对象技术的三大特征之一。

1.1.3 Java 与其他编程语言间的关系

程序开发语言可分为 4 代：机器语言、汇编语言、高级语言和面向对象程序设计语言。机器语言是机器最终执行时所能识别的二进制序列，任何其他语言编写的程序最后都要转换为相应的机器语言才能运行。在电子计算机刚刚诞生的一小段时间内，人们只能用 0、1 进行编程，后来为了提高编程效率，引入了英文助记符，才出现了汇编语言。汇编语言的出现，大大提升了代码的编写速度，同时也使代码可读性和可维护性大大提高。直到今天，仍然有人在用汇编语言进行编程，当然这主要是为底层使用(如一些硬件驱动之类)，毕竟汇编的执行效率高。但是，汇编对于程序员的自身要求还是很高的，一般需要程序员是专业出身的。这就限制了其他领域的科技工作者们利用计算机进行辅助工作。因此，为了普及计算机使之作为社会各行各业的一种工具，需要开发语法简单，编写容易的编程语言。Bill Gates 的第一桶金据说就是从这个需求中赚来的，他在大学时代设计开发了 Basic 语言，并将其出售给 IBM 公司。除了 Basic，还有很多其他的高级语言，如 Pascal、Fortran、C 等。随着软件业的不断发展，软件规模变得越来越大，迫切需要更高效的编程语言。应此需求，Java、C++、Visual Basic 和 Delphi 等应运而生。除此之外，世界上还有很多其他编程语言，只不过它们不是很流行，并不被人们熟知。每一种流行的开发语言都有其优势：C 语言适合用来开发系统程序，很多的操作系统及驱动程序都是用 C 语言来编写的；Fortran 适合于用来进行数值计算；Pascal 语言结构严谨，适合作为教学语言；Visual Basic 和 Delphi 适合用来开发中小型的应用程序；C++ 适合开发大型的应用程序；Java 适于开发跨平台的应用程序。

总之，每种语言都有其特色，至于选用什么语言作为开发工具，关键要看具体的开发任务。没有最好的，只有合适的。很多开发任务可能需要同时使用几种开发语言一起来完成。本教程主要面向没有任何编程基础的初学者而编写。下面就开始简单的 Java 之旅吧。

1.2 第一个 Java 程序

用 Java 编写的程序有两种类型:Java 应用程序(Java Application)和 Java 小应用程序(Java Applet)。虽然二者的编程语法是完全一样的,但后者需要客户端浏览器的支持才能运行,并且在运行前必须先将其嵌入 HTML 文件的<applet>和</applet>标签对中。当用户浏览该 HTML 页面时,首先从服务器端下载 Java Applet 程序,进而被客户端已装的 Java 虚拟机解释和运行。由于 Applet 与 HTML 联系紧密,且编程相对复杂些,因而将其放至后面章节中讲述,这里只对 Java Application 程序进行介绍。下面看一个最简单的 Java 程序。

```
public class Hello {  
    public static void main(String args[])  
    {  
        System.out.println("Hello,welcome to Java programming.");  
    }  
}
```

Java 源程序是以文本格式存放的,文件扩展名必须为.java。如上面这个程序,本书将其保存为 Hello.java 文件。这里有个非常细小但千万要注意的问题:文件名务必与(主)类名一致,包括字母大小写也要一致;通常定义类时,类名的第一个字母都大写。所以,在正确编辑以上代码后,保存时应确保文件名是正确,否则后面将不能通过编译,更运行不了。所有的 Java 语句都必须以英文的“;”结束,编辑程序时千万注意别误输入中文的“;”,因为中文“;”不能被编译器识别。此外,Java 是大小写敏感的,编辑程序时应注意区分关键字及标识符中的大小写字母。

下面通过图 1-2 对该程序的构成做一简要介绍。

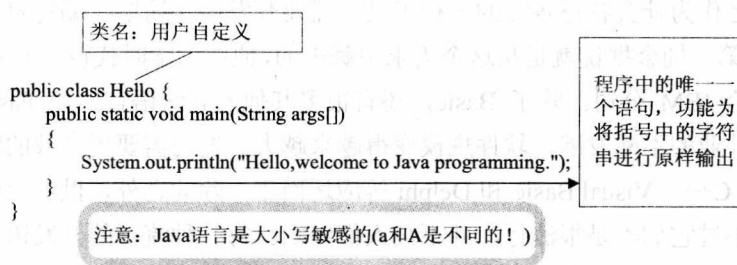


图 1-2 第一个 Java 程序

上述程序中,首先用关键字 `class` 来声明一个新类,类名为 `Hello`,它是一个公共类(`public`),整个类定义由大括号`{}`括起来。在该类中定义了一个 `main()`方法。其中,`public`表示访问权限,指明所有的类都可以调用(使用)这一方法;`static`指明该方法是一个静态类方法,它可以通过类名被直接调用;`void`则指明 `main()`方法不返回任何值。对于一个应用程序来说,`main()`方法是必需的,而且必须按照如上的格式来定义。Java 解释器在没有生成任何实例的情况下,以 `main()`方法作为入口来执行程序。Java 程序中可以定义多个类,

每个类中也可以定义多个方法，但是最多只能有一个公共类，main()方法也只能有一个。在main()方法定义中，括号()中的String args[]是传递给main()方法的参数，参数名为args，它是类String的一个实例，参数可以为0个或多个，每个参数用“类名参数名”来指定，多个参数之间用逗号分隔。在main()方法的实现(大括号中)时，只有一条语句：System.out.println("Hello,welcome to Java programming.");，它用来实现字符串的输出，这条语句实现与C语言中的printf语句和C++中cout<<语句具有相同的功能。

图1-2中，除了类名的定义和唯一的一个程序语句外，其他部分可以将其看作模板，照抄即可，但要注意大小写和大括号的配对。

简单Java程序的模板如下：

```
public class 类名 {  
    public static void main(String args[])  
    {  
        //程序代码  
    }  
}
```

提示：

- 类名称后面的大括号标识着类定义的开始和结束，而main方法后面的大括号则标识着方法体的开始和结束。Java程序中的大括号都是成对出现的，因而在写左大括号后，最好也把右大括号写上，这样可以避免漏掉，否则可能会给程序编译调试带来不便。初学者常在这里犯错，花了很多时间查错，最后发现原来是大括号不配对。
- 通常，习惯将类名的首字母用大写，而变量则以小写字母打头，变量名由多个单词组成时，第一个单词后边的每个单词首字母大写。
- 程序中应适当使用空格符和空白行来对程序语句元素进行间隔，增加程序的可读性。一般在定义方法内容的大括号中，将整个方法体的内容部分缩进，使程序结构清晰，一目了然。编译器会忽略这些间隔用的空格符及空白行，也就是说，它们仅仅起到提高程序可读性的目的，而不对程序产生任何影响。
- 在编辑程序时，最好一条语句占一行。另外，虽然Java允许一条长的语句分开写在几行中，但前提是不能从标识符或字符串的中间进行分割。另外，文件名与public类名在拼写和大小写上必须保持一致。
- 一个Java Application程序必须定义有且仅有一个main方法，它是程序的执行入口。除了main方法外，程序还可以有其他方法，后面章节会有介绍。

1.3 Java 开发工具

编写Java源程序的工具软件有很多，只要是能编辑纯文本的都可以，如notepad(记事本)、wordpad(写字板)、UltraEidt、EditPlus等。Java软件开发人员一般用一些IDE(Integreted

Development Environment, 集成开发环境)来编写程序,以提高效率和缩短开发周期。下面介绍一些比较流行的 IDE 及其特点。

1. Borland 的 JBuilder

有人说 Borland 的开发工具都是里程碑式的产品,从 Turbo C、Turbo Pascal 到 Delphi、C++ Builder 等都是经典。JBuilder 是第一个可开发企业级应用的跨平台开发环境,支持最新的 Java 标准,它的可视化工具和向导使得应用程序的快速开发变得可以轻松实现。

2. IBM 的 Eclipse

Eclipse 是一种可扩展、开放源代码的 IDE,由 IBM 出资组建。Eclipse 框架灵活、易扩展,因此深受开发人员的喜爱。目前,它的支持者越来越多,大有成为 Java 第一开发工具之势。

3. Oracle 的 JDeveloper

JDeveloper 的第一个版本采用的是购买的 JBuilder 的代码设计的,不过已经完全没有 JBuilder 的影子。现在 JDeveloper 不仅是很好的 Java 编程工具,而且还是 Oracle Web 服务的延伸。

4. Symantec 公司的 Visual Cafe for Java

很多人都知道 Symantec 公司的安全产品,但很少有人知道 Symantec 的另一项堪称伟大的产品: Visual Cafe。有人认为 Visual Cafe 如同当年 Delphi 超越 Visual Basic 一样,今天,它也超越了 Borland 的 Delphi。

5. IBM 的 Visual Age for Java

Visual Age 是一款非常优秀的集成开发工具,但用惯了微软开发工具的朋友在开始时可能会感到非常不舒服。这是因为 Visual Age for Java 采取了与微软截然不同的设计方式。为什么会这样呢?那是因为蓝色巨人怎么能跟着微软的指挥棒转呢!

6. Sun 公司的 NetBeans 与 Sun Java Studio 5

以前叫 Forte for Java,现在 Sun 将其统一称为 Sun Java Studio 5。出于商业考虑, Sun 将这两个工具合在一起推出,不过它们的侧重点是不同的。

7. Sun 公司的 Java WorkShop

Java WorkShop 是完全用 Java 语言编写的,是当今市场上销售的第一个完整的 Java 开发环境。目前,Java WorkShop 支持 Solaris 操作环境 SPARC 和 Intel 版、Windows 95、Windows NT 以及 HP/UX。

8. BEA 公司的 WebLogic Workshop

BEA WebLogic Workshop 8.1 是一个统一、简化、可扩展的开发环境。除了提供便捷

的 Web 服务外，它还能用于创建更多种类的应用。作为整个 BEA WebLogic Platform 的开发环境。不管是创建门户应用、编写工作流，还是创建 Web 应用，Workshop 8.1 都可以帮助开发人员更快更好地完成。

9. Macromedia 公司的 JRUN

提起 Macromedia 公司，人们可能会马上想到 Flash、Dreamweaver，但很少有人知道它还有一款出色的 Java 开发工具 JRUN。JRun 是第一个完全支持 JSP 1.0 规格书的商业化产品。

10. Sun 公司的 JCreator

JCreator 的设计接近 Windows 界面风格，用户对它的界面比较熟悉，但其最大特点却是与 JDK 的完美结合，这是其他任何一款 IDE 所不能比的。

11. Microsoft Visual J++

严格地说，Visual J++ 已经不是真正的 Java 了，而是微软版的 Java。作为开发工具，它保留了微软开发工具一贯具有的亲和性。

12. Apache 开放源码组织的 Ant

国内程序员中 Ant 的使用者很少，但它却很受硅谷程序员的欢迎。Ant 在理论上有些类似于 C 中的 make，但没有 make 的缺陷。

13. IntelliJ IDEA

IntelliJ IDEA 的界面非常漂亮，堪称 Java 开发工具中的第一“美女”，但用户在一开始很难将它的功能配置达到“完美”境界。不过正是由于可自由配置功能这一特点，让不少程序员眷恋难舍。

综上所述，可以用来开发 Java 的利器很多。在计算机开发语言的历史中，从来没有哪种语言像 Java 这样受到如此众多厂商的支持，有如此多的开发工具。Java 菜鸟们如初入大观园的刘姥姥，看花了眼，不知该何种选择。的确，这些工具各有所长，没有绝对完美的，就算是老鸟也很难做出选择。但要记住的是，它们仅仅是集成的开发环境，而在这些环境中，有一样东西是共同的，也是最核心和关键的，那就是 JDK(Java Development Kits)，中文意思是 Java 开发工具集，JDK 是整个 Java 的核心，包括了 Java 运行环境(Java Runtime Environment)、一堆 Java 工具和 Java 的基础类库(rt.jar)等，所有的开发环境都需要围绕它来进行，缺了它就什么都做不了。对于初学者的建议是：JDK+记事本就足够了，因为掌握 JDK 是学好 Java 的第一步也是最重要的一步。首先用记事本来编辑源程序，然后再利用 JDK 来编译、运行 Java 程序。这种开发方式虽然简陋，但不失为学 Java 语言的好途径。第一个 Java 程序已经编辑好了，并保存为 Hello.java，接下来就开始编译和运行它。

1.4 Java 程序开发步骤

要编译和运行 Java 程序，首先要下载并安装 JDK。