

Java语言 程序设计

本书编委会

浙江科学技术出版社

(计算机类)

世纪高等教育精品大系

Java 语言 程序设计

本书编委会

浙江科学技术出版社

(计算机类)

世纪高等教育精品大系

图书在版编目(CIP)数据

Java 语言程序设计/《Java 语言程序设计》编委会编. —
杭州: 浙江科学技术出版社, 2006
(世纪高等教育精品大系)
ISBN 7-5341-2824-2

I . J... II . J... III . JAVA 语言 - 程序设计
IV . TP312

中国版本图书馆 CIP 数据核字 (2006) 第 003345 号

丛 书 名	世纪高等教育精品大系
书 名	Java 语言程序设计
编 著	本书编委会
出版发行	浙江科学技术出版社
联系电话	(0571) 85152486
电子邮箱	javaprogramming@163.com
印 刷	宁波市大港印务有限公司
开 本	787×1092 1/16
印 张	17
字 数	387 000
版 次	2006 年 1 月第 1 版
印 次	2006 年 1 月第 1 次印刷
书 号	ISBN 7-5341-2824-2
定 价	23.00 元
责任编辑	张祝娟
封面设计	孙 菁

《Java 语言程序设计》

编 辑 委 员 会

主 任 方永平

副主任 胡维华

委 员 (以姓氏笔画为序)

陈庆章 何钦铭 项小仙 赵建民

俞瑞钊 蒋联海 雷 炜 楼程富

主 编 胡维华

副主编 韩建平

前 言

Java 语言具有面向对象、跨平台、安全性、多线程以及强大的网络支持等显著特点,这使得 Java 语言成为许多应用系统的理想开发工具。随着 Internet 的产生和发展,Java 语言的这些优点引起软件开发人员极大的关注,Java 语言已经成为最流行的网络编程语言之一。在欧美国家,许多大学将 Java 语言作为第一门程序设计课程。在国内,越来越多的高校选择 Java 语言作为程序设计课程的入门语言。

本书主要是为高等学校学生学习“Java 语言程序设计”课程而编写的教材,教学对象是 Java 语言的初学者。在学习本书之前,并不要求具备面向对象的基础知识,也不要求学习或接触过 C 或其他高级程序设计语言。

本书主要讲述 Java 程序设计的基础知识和基本方法。在编写过程中,结合作者多年 Java 程序设计教学中的经验与体会,针对程序设计语言初学者的特点,力求体现以下特色:

适当加大了流程控制语句等 Java 语言基础知识方面的讲述,在教学安排上注重编程能力的培养,着力训练好程序设计的基本功。

在全面讲述了 Java 语言基本语法和面向对象程序设计基本概念的基础上,更强调如何利用 Java 语言解决实际应用问题的能力。

本书的内容安排及阐述遵循由浅入深、循序渐进的原则,书中的实例程序力求简洁明了,能在最基本的 Java 运行环境下执行。每章后面精心设计了练习题,读者可以通过做这些习题,巩固并灵活运用所学知识。

本书还结合新技术的发展,通过一组实例介绍网络、多媒体、JDBC 数据库等编程技术,使读者在掌握 Java 基本概念和编程方法的同时,能全面了解 Java 的特点,掌握 Java 最新实用技术。

本书编写得到了浙江省高校计算机教学研究会和浙江科学技术出版社的大力支持。全书由胡维华教授担任主编并设计全书结构,韩建平副教授担任副主编并统稿。全书共分 12 章,其中瞿有甜老师编写第 1 章和第 4 章,田贤忠老师编写第 2 章和第 3 章,韩建平老师编写第 5 章和第 6 章,杨蓉老师编写第 7 章和第 8 章,杨柏林老师编写第 9 章和第 10 章,罗显贵老师编写第 11 章和第 12 章。全书由胡维华和韩建平统稿,徐小良担任主审。

本书可以作为各类大专院校、各类培训与等级考试的教学用书,也可作为对 Java 语言程序设计感兴趣者的自学用书。

本书在编写过程中,参阅了大量国内外书籍和网站参考资料,在此表示衷心的感谢。

由于作者水平有限,书中难免还存在缺点和错误,期待广大读者批评和指正。

本书编委会

2006 年 1 月

目 录

第 1 章 Java 概述	1
1.1 Java 语言简介	1
1.1.1 Java 的来历	1
1.1.2 Java 语言的现状	2
1.1.3 Java 语言的特点	3
1.1.4 Java 程序的工作原理	5
1.2 Java 语言开发环境	7
1.2.1 Java 平台的三个版本	7
1.2.2 Java 程序执行开发环境的建立	8
1.2.3 Java 集成环境开发工具简介	11
1.2.4 Java 在线帮助	11
1.3 初识 Java 程序	13
1.3.1 Java Application	13
1.3.2 Java Applet	14
1.4 Java 程序的基本输入与输出	16
1.4.1 Java Application 字符界面的输入与输出	16
1.4.2 Java Applet 图形界面输入与输出	19
1.4.3 Java Application 图形界面输入与输出	21
习题	22
第 2 章 基本数据类型与表达式	23
2.1 基本数据类型	23
2.1.1 数据类型分类	23
2.1.2 基本数据类型	24
2.1.3 数据类型的转换	26
2.2 常量与变量	27
2.2.1 标识符与关键字	28
2.2.2 常量	28
2.2.3 变量	30
2.3 表达式	32
2.3.1 赋值表达式	32



2.3.2 算术表达式.....	33
2.3.3 关系表达式.....	36
2.3.4 逻辑表达式.....	37
2.3.5 条件表达式.....	38
2.3.6 位运算表达式.....	39
习题.....	41
第3章 流程控制语句.....	43
3.1 Java 语句概述.....	43
3.1.1 Java 语句.....	43
3.1.2 程序的三种基本结构.....	44
3.2 选择语句.....	45
3.2.1 if 语句.....	45
3.2.2 switch 语句.....	52
3.3 循环语句.....	53
3.3.1 while 语句.....	53
3.3.2 do-while 语句.....	55
3.3.3 for 语句.....	57
3.3.4 break 和 continue 语句.....	60
3.4 循环的嵌套.....	62
习题.....	64
第4章 数组和字符串操作.....	66
4.1 一维数组.....	66
4.1.1 一维数组的声明.....	67
4.1.2 一维数组的创建与初始化.....	67
4.1.3 一维数组的访问.....	68
4.2 二维数组.....	69
4.2.1 二维数组的声明.....	69
4.2.2 二维数组的创建与初始化.....	69
4.2.3 二维数组的访问.....	70
4.3 数组应用举例.....	71
4.4 字符串的操作.....	76
4.4.1 字符串概述.....	76
4.4.2 String 字符串说明及初始化.....	76
4.4.3 String 字符串的基本操作.....	77
4.4.4 字符串的查找.....	78



4.4.5 字符串比较.....	79
4.4.6 字符串缓冲区类 StringBuffer	80
习题	82
第 5 章 类与对象	86
5.1 类的定义	86
5.1.1 类与对象的基本概念.....	86
5.1.2 类定义的基本框架.....	87
5.1.3 成员变量定义.....	88
5.1.4 方法的定义.....	88
5.2 对象的创建和使用.....	89
5.2.1 声明与创建对象.....	89
5.2.2 构造方法与对象初始化.....	90
5.2.3 访问对象.....	91
5.3 方法的进一步讨论.....	92
5.3.1 方法的调用与返回值.....	92
5.3.2 方法的参数传递.....	94
5.3.3 递归方法.....	96
5.3.4 方法的重载.....	98
5.4 静态变量与静态方法.....	100
5.4.1 静态变量.....	100
5.4.2 静态方法.....	101
5.4.3 静态初始化器.....	103
5.5 this 关键字	103
5.5.1 使用 this 解决局部变量与成员变量同名问题.....	103
5.5.2 在构造方法中调用另一个重载的构造方法	103
5.6 包	104
5.6.1 包的含义.....	104
5.6.2 创建包.....	104
5.6.3 使用包中的类.....	105
习题	106
第 6 章 继承、封装与多态	110
6.1 类的继承	111
6.1.1 继承的基本概念.....	111
6.1.2 子类的定义.....	111
6.1.3 域的继承、隐藏与添加.....	114



6.1.4	方法的继承、重载与覆盖	114
6.1.5	super 关键字的使用	115
6.1.6	子类构造方法与父类构造方法	116
6.1.7	父类对象与子类对象	118
6.2	封装与访问控制	119
6.2.1	封装	119
6.2.2	类成员的访问控制符	120
6.2.3	类的访问控制符	123
6.3	抽象类与接口	123
6.3.1	抽象类和抽象方法	123
6.3.2	接口的声明与实现	126
6.4	final 关键字	128
6.4.1	final 变量	128
6.4.2	final 方法	129
6.4.3	final 类	129
6.5	多态性	129
6.5.1	覆盖实现多态性	130
6.5.2	重载实现多态性	131
6.6	内部类和匿名类	132
6.6.1	内部类	132
6.6.2	方法中的内部类	133
6.6.3	匿名类	134
6.7	Java 类库与常用类	135
6.7.1	Java 类库及结构	135
6.7.2	Object 类	136
6.7.3	Math 类	138
6.7.4	基本类型的包装类	140
6.7.5	System 类	141
习题		142
第 7 章	异常处理	146
7.1	异常与异常处理类	146
7.1.1	异常的基本概念	146
7.1.2	异常处理类	147
7.2	异常处理	148
7.2.1	捕获与处理异常	148
7.2.2	多异常的捕获与处理	150



7.2.3 finally 子句	151
7.3 异常抛出	152
7.3.1 异常抛出	152
7.3.2 用 throw 语句抛出异常	153
7.3.3 声明异常	153
7.4 自定义异常	154
习题	155
第 8 章 Java 输入输出及文件处理	157
8.1 流式输入输出	157
8.1.1 流式输入输出概述	157
8.1.2 输入输出流的类层次	158
8.1.3 基本输入输出流类	159
8.1.4 其他输入输出流类	161
8.1.5 标准输入输出	162
8.2 文件与目录	163
8.2.1 文件与目录管理	164
8.2.2 文件输入输出流	166
8.2.3 随机存取文件	169
习题	170
第 9 章 图形用户界面编程	172
9.1 图形用户界面编程概述	172
9.1.1 AWT 组件和 Swing 组件	172
9.1.2 容器和组件	173
9.1.3 如何创建一个图形程序	174
9.2 布局管理	175
9.2.1 Flow 布局管理器	175
9.2.2 Border 布局管理器	177
9.2.3 Grid 布局管理器	178
9.2.4 Card 布局管理器	180
9.2.5 GridBag 布局管理器	180
9.3 事件处理	180
9.3.1 事件处理模型	181
9.3.2 事件监听器接口	183
9.3.3 事件适配器 (Adapter)	187
9.3.4 用内部类实现事件处理	188



9.4 使用 Swing 组件创建 GUI	191
9.4.1 Swing 组件的分类	191
9.4.2 使用 Swing 容器建立窗口	192
9.4.3 Swing 图形界面组件	198
9.5 文本、字体、颜色和图形绘制	209
9.5.1 文本与字体	209
9.5.2 颜色	210
9.5.3 图形的绘制	210
习题	213
第 10 章 Applet 程序设计	218
10.1 Applet 概述	218
10.2 Applet 的生命周期和主要方法	219
10.2.1 init ()	219
10.2.2 start ()	219
10.2.3 stop ()	220
10.2.4 void paint (Graphics g)	220
10.2.5 destory ()	220
10.3 Applet 应用	222
10.3.1 在 Web 浏览器中运行 Applet	222
10.3.2 Applet 实例	223
习题	226
第 11 章 多线程	229
11.1 多线程概述	229
11.1.1 什么是多线程	229
11.1.2 Java 线程模型	229
11.2 创建线程	231
11.2.1 主线程	231
11.2.2 继承 Thread 类创建线程	232
11.2.3 实现 Runnable 接口创建线程	234
11.3 线程调度与优先级	236
11.3.1 线程调度	236
11.3.2 线程优先级	238
11.4 线程同步	239
习题	242



第 12 章 Java 网络、数据库和多媒体编程.....	244
12.1 网络通信.....	244
12.1.1 URL 通信.....	244
12.1.2 Java Socket 通信.....	246
12.2 JDBC 编程.....	249
12.2.1 JDBC 模型.....	249
12.2.2 Java 访问数据库步骤	249
12.3 Java 与多媒体	252
12.3.1 图像的使用.....	252
12.3.2 声音播放.....	255
习题.....	257
参考文献.....	258



Java 概述

 本章要点:

- ★ Java 语言有哪些特点?
- ★ 如何来建立 Java 语言的开发环境, 常见的开发工具有哪些?
- ★ 开发 Java 程序的基本步骤是怎样的?
- ★ Java 程序是如何来实现字符界面及图形界面的输入输出的?

自第一台电子计算机诞生以来, 计算机产业的发展以及计算机向其他各领域的渗透速度异常迅速, 曾经一度是计算机专业技术人员为之骄傲而又令非专业人员望洋兴叹的编程技术已经逐渐揭开了其神秘的面纱, 计算机已逐渐成为各行各业的基本工具, 编程技术也逐渐成为高校理工科各专业学生的基本技能之一。计算机程序设计语言的发展历程也正好顺应了各类计算机应用技术发展的这种需求, 从第一代(机器)语言、第二代(汇编)语言到第三代(高级)语言, 每一发展阶段都使得程序越来越容易编写、阅读、维护、复用和移植。Java 语言就是在这样的环境中发展起来的一种高级语言, 易学易用, 并越来越受推崇。

1.1 Java 语言简介

Java 是一种很新的计算机语言, 拥有“因特网上的世界语”的美称。本节主要介绍 Java 的来历、语言特点和工作机制。

1.1.1 Java 的来历

1991 年 Sun Microsystems 公司资助了一项绿色项目, 该项目由 James Gosling 负责, 致

力于开发一种可移植、跨平台的语言，该语言能够生成运行于不同环境、不同 CPU 芯片上的代码。开发组开始准备采用 C++ 后来发现 C++ 太复杂且安全性差，最后基于 C++ 开发了一种新的语言，取名为 Oak（橡树），这就是 Java 的雏形语言，整个研发过程历时一年半。

尽管 Oak 是一种精巧而安全的语言，但是 Sun 公司发现智能消费型电子产品发展没有预想的那么快，而且 Sun 公司在投标一个交互式电视大项目时败给了 SGI。可怜的 Oak 几乎无家可归，就在 Sun 公司差一点就要取消这个项目的时候，Mark Andreessen 开发了 Mosaic 和 Netscape，使 Internet 迅速兴起，这启发了 Oak 项目组成员，他们发现在编制嵌入式控制器代码时经常遇到的可移植性问题，在编制因特网代码的过程中也出现了。Java 语言同样可以被用在因特网上。这个认识使他们将 Java 的重心由电子消费品转移到 Internet 编程。于是他们用 Java 编制了 HotJava 浏览器，结果得到了 Sun 公司首席执行官 Scott Mc-Nealy 的充分肯定和支持，这触发了 Java 进军 Internet 的兴趣。1993 年，Sun 公司重新分析市场需求，认为网络具有很好的发展前景，而且 Java 语言似乎非常适合于网络编程。于是 Sun 公司将 Java 语言的应用背景转向网络市场，为网页增加了“动态的内容”。因此，跨平台的编程语言的需要是促使 Java 诞生的源动力，而 Internet 最终导致了 Java 的成功。

Java 语言为网络的发展开辟了一个新纪元。Java 语言被计算机杂志《PC Magazine》评为 1995 年十大优秀科技产品。微软公司总裁比尔·盖茨当时的一句话“Java 语言是有史以来最卓越的计算机程序设计语言”也反映了当时人们对 Java 语言的普遍评价。

Java 的取名还有一个趣闻，有一天，几位 Java 会员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java（爪哇——咖啡的原产地），有一个人灵机一动说就叫 Java 怎样，得到了其他人的赞赏，于是，Java 这个名字就这样传开了。

1.1.2 Java 语言的现状

Java 是 Sun 公司推出的新一代面向对象程序设计语言，特别适合于 Internet 应用程序开发，它的平台无关性直接威胁到 Windows 和 Intel 的垄断地位。一时间，“连 Internet，用 Java 编程”，成为技术人员的一种时尚。虽然新闻界的报导有些言过其实，但 Java 作为软件开发的一种革命性的技术，其地位已被确立，这表现在以下几个方面：

1) 计算机产业的许多大公司购买了 Java 的许可证，包括 IBM、Apple、DEC、Adobe、SiliconGraphics、HP、Oracel、Toshiba、Netscap、Novell 及 SGI 等，包括最不愿的 Microsoft。这一点说明，Java 语言已得到了工业界的认可。

2) 众多的软件开发商开始支持 Java 的软件产品。例如：Borland 公司开发了基于 Java 的快速应用程序开发环境 JBuilder。Sun 公司自己的 Java 开发环境 Java Workshop 等。数据库厂商如：Illustra、Sysbase、Versant、Oracle 公司等都在开发 CGI 接口，以支持 HTML 和 Java 语言。

3) Intranet 正在成为企业信息系统最佳的解决方案，而其中 Java 将发挥不可替代的作用。Intranet 的目的是把 Internet 用于企业内部的信息系统，它的优点表现在：便宜，易于使用和管理。用户不管使用何种类型的机器和操作系统，界面是统一的 Internet 浏览器，而数据库、Web 页面、应用程序（用 Java 编的 Applet）则存在 WWW 服务器上，无论是开发人员，还是管理人员，还是用户都可以受益于该解决方案。



1.1.3 Java 语言的特点

Java 之所以能够受到如此众多的好评以及拥有如此迅猛的发展速度，与其语言本身的特点是分不开的。其主要特点总结如下：

1. 简单性

Java 语言是在 C 和 C++ 计算机语言的基础上进行简化和改进的一种新型计算机语言，它去掉了 C 和 C++ 中最难准确应用的指针和最难理解的多重继承技术等内容，通过垃圾自动回收机制简化了程序内存管理，统一了各种数据类型在不同操作系统平台上所占用的内存大小。Java 程序的简单性是其得以迅速普及的最重要的原因之一。

2. 面向对象

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术，在面向对象特性上它比 C++ 更彻底。Java 支持静态和动态风格的代码继承及重用。

3. 支持网络编程

Java 是面向网络的语言。通过 Java 提供的类库可以处理 TCP/IP 协议。用户可以通过 URL 很方便地访问网络资源。Java Applet 是动态安全、跨平台的网络应用程序，可以嵌入 HTML 语言中，通过网页发布到 Internet。Java 语言是目前对网络支持最全面、与网络关系最密切的一种计算机语言。

4. 鲁棒性（健壮性）

鲁棒性指的是程序执行的稳定性，也常常称为健壮性。Java 致力于检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。Java 语言中的垃圾自动回收机制和严格的异常处理机制在很大程度上提高了程序的鲁棒性。这些功能特征大大缩短了开发 Java 应用程序的周期。

5. 平台无关性（结构中立）/可移植性

Java 语言的设计目标是让其程序不用修改就可以在任何一种计算机平台上运行，Java 将它的程序编译成一种结构中立的与机器无关的中间文件格式（byte-code 格式）。只要有 Java 运行系统（Java 虚拟机）的机器都能执行这种中间代码。现在，Java 运行系统可以安装在多种软、硬件系统平台上。例如，UNIX 系统和 Windows 系统等。

6. 安全性

Java 的安全性可从两个方面得到保证。一方面，在 Java 语言里，像指针和释放内存等 C++ 功能被删除，避免了非法内存操作；另一方面，当 Java 用来创建浏览器时，语言功能和浏览器本身提供的功能结合起来，使它更安全。

Java 语言在机器上执行前，要经过很多次的测试。它经过代码校验，检查代码段的格式，检测指针操作，对象操作是否过分以及是否试图改变一个对象的类型等。这些机制保证了 Java 程序运行的可靠性。



7. 解释性

Java 解释器（运行系统）能直接运行目标代码指令。链接程序通常比编译程序所需资源少，所以程序员可以在创建源程序上花上更多的时间。因为解释执行的语言一般会比编译执行的语言（如 C 和 C++ 语言）的执行效率低，所以，这其实也是 Java 的一个缺点。

8. 高性能

如果解释器速度不慢，Java 可以在运行时直接将目标代码翻译成机器指令。Sun 用直接解释器一秒钟内可调用 300 000 个过程。翻译目标代码后运行的速度与 C 和 C++ 的性能没什么区别。

9. 多线程

Java 提供的多线程功能使得在一个程序里可同时执行多个小任务。线程（有时也称小进程）——是一个大进程里分出来的小的独立的进程。因为 Java 的多线程技术实现得较好，所以比 C 和 C++ 更健壮。多线程带来的更大的好处是更好的交互性能和实时控制性能。当然实时控制性能还取决于系统本身（UNIX、Windows、Macintosh 等）。任何用过当前浏览器的人，都感觉为调一副图片而等待是一件很烦恼的事情。在 Java 里，你可用一个单独的线程来调一副图片，而你可以访问 HTML 里的其他信息而不必等它。

多线程主要是用来处理复杂事务或需要并行的事务。Java 虚拟机本身就是一个多线程的程序。采用多线程机制是提高程序运行效率的一种方法，当然也增加了程序设计的难度。

10. 动态

Java 的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行过程中所需要的类，这是 C++ 语言进行面向对象程序设计所无法实现的。Java 从如下几方面采取措施解决这个问题。Java 编译器不是将对实例变量和成员函数的引用编译为数值引用，而是将符号引用信息在字节码中保存下传递给解释器，再由解释器在完成动态连接类后，将符号引用信息转换为数值偏移量。这样，一个在存储器生成的对象不在编译过程中决定，而是延迟到运行时由解释器确定。这样，对类中的变量和方法进行更新时就不至于影响现存的代码。解释执行字节码时，这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一次，随后代码便可以全速执行。在运行时确定引用的好处是可以使用已被更新的类，而不必担心会影响原有的代码。如果程序连接了网络中另一系统中的某一类，该类的所有者也可以自由地对该类进行更新，而不会使任何引用该类的程序崩溃。Java 还简化了使用升级或者更新协议的方法。如果你的系统运行 Java 程序时遇到了不知怎样处理的程序，没关系，Java 能自动下载你所需要的功能程序。

总而言之，Java 语言是一种易学好用，健壮性高，但执行效率相对较低的计算机语言。它适合于各种对执行时间要求不是很苛刻的应用程序。用 Java 语言编写程序一般比其他计算机语言编写程序花费更少的时间，而且调试所需的时间也会较短。对于计算机初学者或正打算开始学习一门计算机语言的工程师或教学科研工作者来说，选择 Java 程序设计是一个很好的方案。



1.1.4 Java 程序的工作原理

用任何一种高级编程语言书写的源程序在计算机上运行前须翻译成机器可以理解和执行的指令代码之后才能运行，翻译的方式有编译和解释两种，完成相应的翻译工作的部件或程序分别称为编译器（或编译程序）和解释器（或解释程序）。

Java 是一种解释型的高级语言，但 Java 与传统的解释型高级语言还有所不同，源程序不是直接交给解释器解释运行，而是首先由编译器将 Java 源程序编译为字节码文件（.class 文件），然后再由 Java 解释器通过 Java 虚拟机实现解释执行。图 1-1 给出了其先编译然后被解释的基本过程。下面我们着重介绍一下 Java 程序的工作原理。

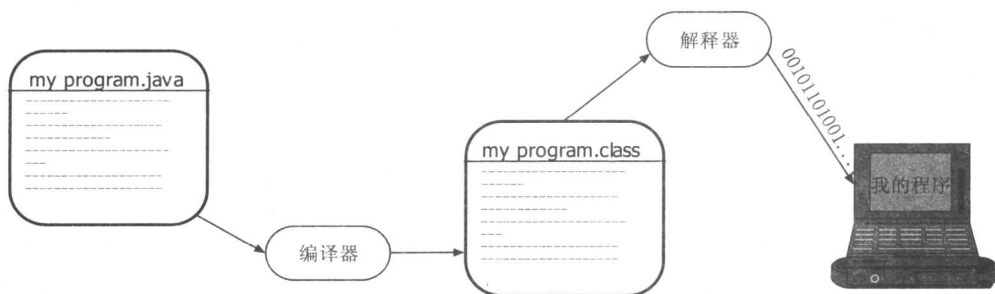


图 1-1 用 Java 编程语言写的程序先被编译，然后被解释

1. Java 虚拟机

为实现 Java 语言的平台无关性，Sun 公司提出了 Java 虚拟机（Java Virtual Machine，简称为 JVM）的机制，其工作原理如图 1-2 所示。

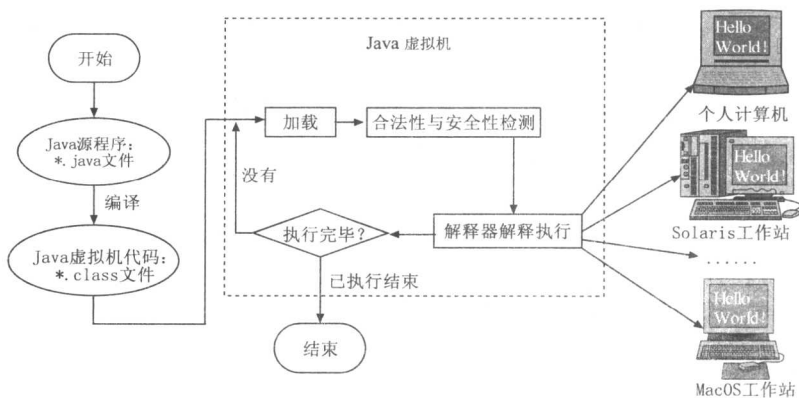


图 1-2 Java 虚拟机的工作原理及其工作流程

Java 虚拟机是 Sun 公司采用软件技术实现的一种虚拟计算机，它的目的是统一不同计算机系统的机器代码，统一后的代码为 Java 虚拟机代码，它以字节码（Bytecode）的方式进行编码。Java 源程序在编译后生成的后缀为“class”的文件就是这种以字节码进行编码