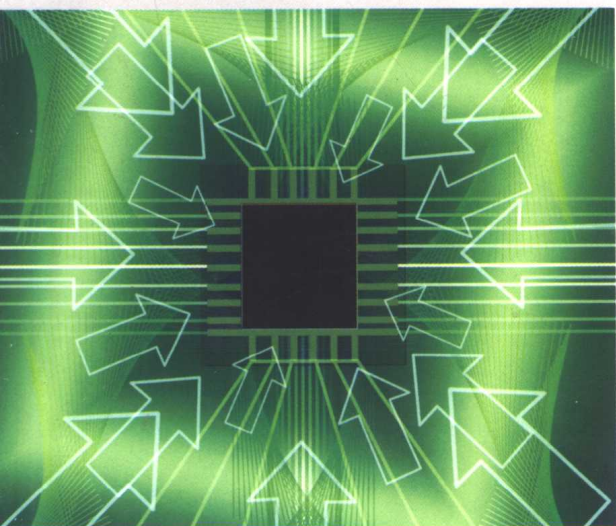




Java

程序设计简明教程

何 桥 李肃义 主 编

谢 昕 李 洋 副主编

张双才 刘 威 王永志 编 委

21 世纪高等院校计算机科学与技术规划教材

Java 程序设计简明教程

何 桥 李肃义 主 编

谢 昕 李 洋 副主编

张双才 刘 威 王永志 编 委

中国水利水电出版社

内 容 提 要

Java 语言是一门很优秀的语言, 具有面向对象、与平台无关、安全、稳定和多线程等优良特性, 是目前软件设计中极为强大的编程语言。本书从 Java 语言的基本特点入手, 逐步介绍了 Java 语言的基本概念和编程方法, 并深入介绍了 Java 的高级特性, 内容包括 Java 语言编程简介、面向对象编程、异常处理、Applet 编程、Java 中的图形图像、Java 中的 GUI、多线程、输入输出流和数据库应用等。

本书共分为 20 章, 内容详尽, 实例丰富, 通俗易懂。前 10 章为理论部分, 每章均附有习题, 后 10 章为实验部分, 是前 10 章的配套实验。

本教材可以作为高等学校本科生和专科生的教学用书, 也可以作为高等学校成人教育的培训教材和教学参考书, 还可以供从事微机应用开发工作的科技人员参考使用。

图书在版编目 (CIP) 数据

Java 程序设计简明教程 / 何桥等主编. —北京: 中国水利水电出版社, 2004
(21 世纪高等院校计算机科学与技术规划教材)

ISBN 7-5084-2313-5

I. J… II. 何… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 076481 号

书 名	Java 程序设计简明教程
主 编	何 桥 李肃义
副 主 编	谢 昕 李 洋
编 委	张双才 刘 威 王永志
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机) 68331835 (营销中心) 82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 15.75 印张 354 千字
版 次	2004 年 9 月第 1 版 2004 年 9 月第 1 次印刷
印 数	0001—5000 册
定 价	24.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

随着计算机技术的飞速发展和计算机应用的日益普及,对高等学校计算机教育提出了越来越高的要求。人们迫切要求掌握与计算机相关的新技术、新知识。Java 语言是众多新技术中的一个亮点,它适应了迅速发展的 Internet 的需要,开创了一种新的计算模式。

本教材主要介绍了 Java 语言的基本概念和编程方法,并深入介绍了 Java 的高级特性。全书共分为 20 章,前 10 章为理论部分,内容包括 Java 语言编程简介、面向对象编程、异常处理、Applet 编程、Java 中的图形图像、Java 中的 GUI、多线程、输入输出流和数据库应用等。后 10 章为实验部分,是前 10 章的配套实验。实验部分是本书的一大特色,强调“边学边做”,融合理论与实践,从实践中巩固所学知识。实验项目目的明确、针对性强,在内容的把握上以实用为原则,突出 Java 的特点和应用方向,充分考虑了人们的思维习惯、学习特点和教学规律。

本书中的所有实例源程序都在 JCreator 开发环境和 J2SDK1.4.0 中编译通过,每个 Java Application 程序都能在安装了 J2SDK1.4.0 的 Windows 环境下正常运行,每个 Java Applet 程序都能在 IE 6.0 浏览器中正常浏览。

本书由何桥、李肃义主编,谢昕、李洋任副主编,另外参加本书编写的还有张双才、刘威、王永志等。

出版本教材,是在计算机教学改革上的一次尝试,虽力图做好,但由于作者水平和经验有限,难免有不足之处,敬请广大同仁和读者批评指正,以便使本教材的质量得到进一步的提高。

编者

2004 年 5 月

目 录

前言

第 1 章 Java 编程简介	1
1.1 概念简介	1
1.1.1 机器语言与高级语言简介	1
1.1.2 操作系统的角色	2
1.1.3 问题的提出	2
1.1.4 解决方案——解释	2
1.2 Java 简介	2
1.2.1 诞生	2
1.2.2 主要特点	3
1.2.3 开发过程	4
1.3 Java 虚拟机 (Java Virtual Machine)	4
1.4 Java 编程基础	5
1.4.1 一个简单的 Java 应用程序	5
1.4.2 一个简单的 Java 小应用程序 (Applet)	7
1.5 小结	9
习题 1	9
第 2 章 Java 基本语法	10
2.1 Java 的标识符和关键字	10
2.2 注释	10
2.3 数据类型	11
2.4 运算符、表达式和语句	13
2.4.1 运算符与表达式	13
2.4.2 语句	14
2.5 字符串和数组	19
2.5.1 字符串	19
2.5.2 数组	22
2.6 小结	24
习题 2	24
第 3 章 面向对象编程	25
3.1 面向对象问题求解的提出	25

3.2	面向对象的分析与设计	26
3.2.1	面向对象的分析 (Object-Oriented Analysis)	26
3.2.2	面向对象的设计 (Object-Oriented Design)	27
3.3	面向对象程序设计方法的优点	28
3.3.1	可重用性	28
3.3.2	可扩展性	29
3.3.3	可管理性	30
3.4	Java 中类的定义和使用	30
3.4.1	类声明	30
3.4.2	类体	30
3.5	对象	36
3.5.1	创建对象	36
3.5.2	对象的内存模型	37
3.5.3	使用对象	37
3.5.4	对象是引用型变量	38
3.6	访问权限	38
3.6.1	方法和变量的访问控制符	38
3.6.2	类的访问控制符	40
3.6.3	访问权限小结	40
3.7	继承	41
3.7.1	创建子类	41
3.7.2	成员变量的隐藏和方法的重写	41
3.7.3	this 和 super 关键字	42
3.8	多态	44
3.8.1	什么是多态	44
3.8.2	多态带来的优越性	46
3.8.3	多态的机制只能用在实例方法上	46
3.9	final 和 abstract 关键字	48
3.9.1	final	48
3.9.2	abstract	48
3.10	接口	50
3.10.1	接口的定义	50
3.10.2	接口的使用	50
3.10.3	接口的多重继承	52
3.10.4	接口的嵌套	52
3.11	内部类	53
3.11.1	非静态内部类	53

3.11.2 静态内部类.....	54
3.12 包 (package)	56
3.12.1 什么是包	56
3.12.2 访问包成员	56
3.12.3 导入包	56
3.12.4 导入子包	57
3.12.5 创建包	57
3.12.6 如何产生 jar 文件	59
3.12.7 Java API (Application Programming Interface) 简介	59
3.13 小结	60
习题 3	62
第 4 章 异常处理	63
4.1 异常简介	63
4.2 基本的异常处理	65
4.3 获得异常的有关信息	68
4.4 抛出异常	69
4.5 在方法声明处抛出异常	70
4.6 自定义异常类	71
4.7 小结	73
习题 4	73
第 5 章 Applet 编程	74
5.1 超文本标记语言 HTML	74
5.1.1 HTML 出现的背景	74
5.1.2 HTML 简介	74
5.2 Java Applet	75
5.2.1 Applet 与应用程序的比较	75
5.2.2 对 Applet 的限制	75
5.3 Applet 的结构	75
5.3.1 Applet 类	76
5.3.2 Applet 的生命周期	76
5.4 Applet 的运行	77
5.5 向 Applet 传递参数	77
5.6 小结	78
习题 5	78
第 6 章 Java 与图形、图像	79
6.1 绘制字符串、字符及字节	79
6.1.1 绘制字符串	79

6.1.2	绘制字符	79
6.1.3	绘制字节	79
6.2	绘制线条和椭圆形	80
6.2.1	绘制线条	80
6.2.2	绘制椭圆	81
6.2.3	填充椭圆	81
6.3	绘制矩形和圆角矩形	82
6.3.1	绘制矩形	82
6.3.2	填充矩形	82
6.3.3	绘制圆角矩形	83
6.3.4	填充圆角矩形	83
6.4	绘制弧和多边形	84
6.4.1	绘制弧	84
6.4.2	填充弧	84
6.4.3	绘制多边形	85
6.4.4	填充多边形	85
6.5	颜色控制	86
6.6	字体控制	87
6.7	绘制图像	88
6.7.1	Image 类	88
6.7.2	绘制图像	88
6.8	小结	90
习题 6		90
第 7 章	Java 中的 GUI	91
7.1	AWT 简介	91
7.1.1	组件和容器	91
7.1.2	组件类的层次	91
7.2	文本框和文本区	92
7.2.1	文本框	92
7.2.2	处理文本框事件	93
7.2.3	文本区	96
7.2.4	处理文本区事件	97
7.3	标签	98
7.4	按钮	98
7.5	选择框	102
7.6	下拉列表框和滚动列表	105
7.7	面板和画布	111

7.8 建立窗口和菜单	114
7.9 对话框	125
7.10 布局设计	129
7.10.1 FlowLayout 布局	129
7.10.2 BorderLayout 布局	130
7.10.3 GridLayout 布局	131
7.11 小结	134
习题 7	134
第 8 章 Java 多线程	135
8.1 线程入门	135
8.2 线程的产生及 Thread 类的方法	136
8.2.1 线程的产生	136
8.2.2 Thread 类的方法	138
8.3 线程的优先级与调度	141
8.4 线程同步	142
8.5 wait-notify 机制	146
8.6 垃圾收集器	148
8.7 小结	148
习题 8	149
第 9 章 输入输出流	150
9.1 java.io 包简介	150
9.2 InputStream 类和 OutputStream 类	150
9.2.1 InputStream 类	151
9.2.2 OutputStream 类	152
9.3 File 类	153
9.4 RandomAccessFile 类	154
9.5 Reader 类和 Writer 类	156
9.5.1 Reader 类	156
9.5.2 Writer 类	157
9.6 流的种类和层次	160
9.6.1 Node stream	160
9.6.2 Processing stream	161
9.7 System 类提供的输入和输出功能	161
9.8 小结	162
习题 9	162
第 10 章 数据库应用	163
10.1 JDBC 简介	163

10.2	设置数据源	163
10.3	数据库连接的建立	166
10.4	执行基本的 SQL 语句	166
10.4.1	Statement 对象与 ResultSet 对象	166
10.4.2	记录的增、删、改、查操作	167
10.5	小结	175
习题 10		175
第 11 章	运行环境及简单程序实验	176
11.1	安装 JDK	176
11.2	设置环境变量	176
11.3	编写 Java 程序	177
第 12 章	基本语法实验	181
12.1	接收命令行参数的程序	181
12.2	接收键盘输入的程序	182
12.3	数据的排序	183
第 13 章	面向对象编程实验	185
13.1	接口的编写	185
13.2	多态在工资系统中的应用	186
13.3	包的建立与使用	190
第 14 章	异常处理实验	193
14.1	异常的捕获	193
14.2	自定义异常	194
第 15 章	小应用程序实验	196
15.1	小应用程序生命周期	196
15.2	带参数的小应用程序	198
第 16 章	图形图像实验	199
16.1	绘制图形	199
16.2	实现动画效果	200
第 17 章	GUI 实验	202
17.1	制作一个简单的画板	202
17.2	菜单的编写	203
17.3	使用 Dialog 实现消息对话框和一般对话框	206
第 18 章	多线程实验	210
18.1	使用 Runnable 接口的方法实现多线程	210
18.2	实现简单动画	211
第 19 章	输入输出流实验	213
19.1	读取文本文件内容	213

19.2 读取图像文件	215
19.3 将内容写入文件	216
第 20 章 数据库应用实验	218
20.1 Java 应用程序连接数据库	218
20.2 Java 小应用程序连接数据库	223
附录 A Java 关键字	226
附录 B Java 编程规则	228
附录 C String 类方法汇总	232
附录 D Thread 类方法汇总	236
附录 E InputStream 类方法汇总	239
附录 F PrintStream 类方法汇总	240

第 1 章 Java 编程简介

本章要点

- 理解 Java 语言的平台无关性
- 掌握 Java 程序的开发过程
- 熟悉简单的 Java 应用程序及小应用程序的编写

1.1 概念简介

人与计算机之间交流信息要使用计算机语言，机器语言是计算机惟一能接受和执行的语言。针对计算机的硬件组成，计算机的设计者按照数字“1”和“0”的不同排列顺序构成了该类型计算机内的各种指令，这些指令用以指挥 CPU 来完成特定的任务，被称为特定计算机的“机器语言”。由于机器语言程序是一些数码序列，不容易看懂，也不便于查错，因此人们又发明了助记符表达的计算机语言，这类语言称为汇编语言。汇编语言仅仅是符号化了的机器语言，仍然是面向机器的，但可移植性差，所以人们又开发出了面向用户的、类似于简单自然语言的各种程序设计语言，称为高级语言。无论是汇编语言还是各种高级语言，它们最终都要转变（即汇编、编译或解释）成 CPU 能识别并有一定格式的机器语言的形式才可运行。

1.1.1 机器语言与高级语言简介

不同种类的计算机有各自的机器语言。因此，特定的命令在被发送到某个特定计算机上时，它所表示的含义与发送到另外一台计算机上时可能完全不同。换言之，在某台计算机上有效的命令可能在另外一台计算机上无效。

所以，为了对计算机进行编程，用户必须知道它所支持的机器语言。每台计算机都基于特定的芯片。例如，PC 机基于 Intel 芯片，Macintosh 基于 Motorola 设计的芯片。每类芯片都有惟一的用于确定它如何转换成特定指令的机制。

由于机器语言因计算机而异。因此，为了能够对特定计算机进行编程，程序员必须学习其机器语言。如果程序员必须使用多种计算机，就会非常麻烦。并且，用机器语言进行编码十分困难，这就导致了高级语言（如 BASIC、FORTRAN 等）的出现。高级语言与用户日常熟悉的自然语言及数学语言很接近，用高级语言编写的程序易学、易读、易修改、通用性好、不依赖于机器，但不能在机器上直接运行。用高级语言编写的程序叫做高级语言源程序，必须翻译成机器语言目标程序才能被计算机执行。高级语言的翻译有两种方式：编译方式和解释方式。

（1）编译方式：先由编译程序把高级语言源程序翻译成目标程序，执行时运行目标

程序。

(2) 解释方式：在运行高级语言源程序时，由解释程序对源程序边翻译边执行。必须经过“编译器”这样的语言处理程序的翻译后才可以被机器接受。用户可以用高级语言编写一次代码，然后使用不同机器的编译器将其翻译成不同机器的机器语言。

1.1.2 操作系统的角色

操作系统是负责管理、控制和维护计算机的系统资源，为用户提供高效、方便、友好的界面的软件。它使用户能够通过键盘、鼠标之类的设备与计算机进行交互。

操作系统是针对不同制造商生产的计算机的类型而开发的软件。因此，操作系统只对它所面向的特定的计算机有效。例如，面向 Intel 计算机的 UNIX 操作系统与面向 Sun 计算机的 UNIX 操作系统不同。

所有程序都使用操作系统的内置函数。例如，某个程序是为使用 Windows 98 操作系统的 Pentium 计算机设计的，则运行时一定会调用 Windows 98 操作系统所提供的 API 函数，如果将此程序不加改动，则肯定不能在使用 UNIX 等不同类型操作系统的计算机上运行。

1.1.3 问题的提出

用高级语言编写程序的第一步是确定运行程序的机器具有何种操作系统、何种处理器芯片（即何种平台）。因为用高级语言编写的源代码经编译后，生成的目标代码将只能在特定的平台上运行。因此，目前的需求是拥有可运行于任何平台上的目标代码。

1.1.4 解决方案——解释

对于上述问题的一个解决方案就是使用解释器。解释器是一种软件，它接受用高级语言编写的代码并逐行执行它。因此，解释器不是将用高级语言编写的整个代码转换为机器语言，而是在阅读代码时一次执行一行。

- 优点：程序编写一次，即可在不同计算机上进行解释。
- 缺点：解释程序的运行速度非常慢。
- 限制：要执行程序的计算机上必须拥有解释器，并且每发布一次新版本的语言，所有用户都必须获得新的解释器。这是由于新版本的语言都会增加一些新的语言特色，这些特色老的解释器是无法解释的。

1.2 Java 简介

Java 是美国 Sun 公司开发的语言，它使用解释器执行代码，因此，无需对源代码进行任何更改即可在不同计算机上运行，是真正跨平台的编程开发语言。

1.2.1 诞生

大约在 1990 年左右，人们发现用 C++ 开发家电产品的软件很不实用，于是开发者以

C++为基础设计了一种新的程序设计语言，命名为 Oak。Oak 先被用于家用电器控制系统中，后来又被用于 VOD 视频点播控制系统中，慢慢地 Oak 成熟起来，不过人们却发现 Oak 这个名字已经被注册了，于是 Oak 的开发者 Sun 公司就起了 Java 这个名字。Java 这个名字是几个技术人员在一起品尝爪哇（Java）岛出产的咖啡时想起来的，这个岛的咖啡非常有名，也许是希望 Java 语言同样出名吧。随着 Java 的不断完善，Sun 公司在 1996 年前后，正式发布了 Java 语言的版本。目前，Java 以其强大的功能成为网络时代的首选编程语言。

Java 编写的软件能在执行码上兼容，只要计算机提供了 Java 解释器，Java 编写的软件就能在其上运行。

1.2.2 主要特点

1. 简单（Simple）

制定 Java 的原则之一，是要建立一种结构简单而且使用容易的系统，可以让用户不必接受很深的训练就可以开始设计程序，所以 Java 的语法尽可能与在当前许多程序设计师都采用的 C 及 C++ 语言相似。并且，Java 删除了 C 及 C++ 许多极少使用、不易理解或常被混淆的功能，如多重继承、指针等。

2. 面向对象（Object-Oriented）

面向对象是近年来信息界广为使用的概念和技术。它有许多良好的特性。对象的封装性可以使对象的接口定义明确；继承性可以增加软件的可重用性，有助于分类及模板设计等。实现面向对象的概念及其各种良好的特性是 Java 的设计理念之一。

3. 分布式（Distributed）

计算机网络的发展使得信息应用朝着分布式的环境发展，所以现代的信息开发语言及环境要有能配合分布式的特性及功能。Java 具有一个网络功能的程序库，其中包含与如 HTTP 和 FTP 等 TCP/IP 网络通信协议整合的能力。

4. 强壮性（Robust）

由 Java 所编写的程序要能在各种情况下运行，而且必须具有高的稳定性。Java 在制定时即加入了能防止存储器被覆写和数据损坏的相关处理机制。

5. 安全性（Secure）

Java 是被设计用于网络及分布式环境中的，所以安全性是一个很重要的考虑。Java 拥有数个从简单到复杂的安全保护措施，能有效地防止病毒的侵入和破坏行为的发生。

6. 结构中立性（Architecture Neutral）

在网络上存在许多不同类型的计算机，从中央处理器到操作系统结构均有很高的差异性。因此要使应用程序在每一种机器上均能运行是相当困难的。针对这个目的，Java 的编译器可以产生一种结构中立的目标码文件格式——字节码（Byte Code）。这种字节码可以在许多种不同的计算机上运行。

7. 多线程（Multithreaded）

多线程是开发复杂和功能强大的程序所必须的手段之一，Java 同样支持这个重要功能。

1.2.3 开发过程

Java 程序的开发过程可以用图 1.1 来表征, 首先应编写 Java 的源程序, 然后经编译器编译生成字节码文件, 最后交给 Java 虚拟机执行 (Java 虚拟机将在下一节中介绍)。

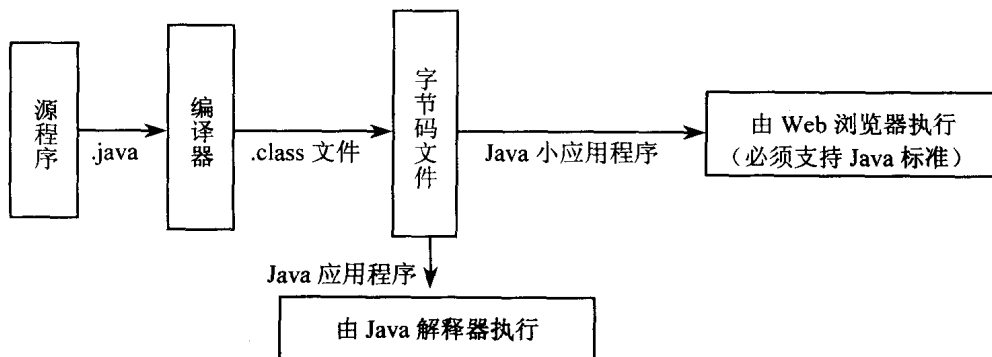


图 1.1 Java 程序的开发过程

1. 源程序

只要使用能够编辑文字文件的编辑器 (例如记事本) 就可以编写 Java 的源程序。但注意不要使用会在文件中加入控制信息的编辑工具 (如 Word)。源程序的内容是依照 Java 的语法编写的, 保存时文件的扩展名为 .java。

2. 编译器

源程序要经过编译器 (javac.exe) 的编译生成扩展名为 .class 的字节码文件。字节码文件是由与平台无关的二进制码组成的, 执行时由解释器解释成本地机器码。

3. 运行

Java 程序分为两大类——Java 应用程序 (Application) 和 Java 小应用程序 (Applet)。Java 应用程序必须通过 Java 解释器 (java.exe) 来解释执行其字节码文件; Java 小应用程序可通过支持 Java 标准的浏览器来解释执行。

1.3 Java 虚拟机 (Java Virtual Machine)

Java 之所以能够“编写一次, 随处运行”就在于有 JVM (Java 虚拟机)。JVM 位于 Java 程序和用户的计算机操作系统之间。

虚拟机只不过是一个基于虚拟计算机设想的软件概念, 是计算机的一种抽象设计, 是由计算机程序来实现的。Java 虚拟机, 即一种知道 Java 语言的各种指令, 并能够将这些指令翻译成用户计算机所能理解的指令程序, 也就是一种“抽象计算机”。

用户编写的 Java 程序首先被编译成能在 JVM 上运行的虚拟指令, 编译后的 Java 程序是可执行文件。不同的是, 这种可执行文件必须运行于 JVM, JVM 为不同的平台解释 Java 程序, 所有与移植有关的问题都在 JVM 中得到解决, 这样用户的 Java 程序就可以不加修

改地运行于任何拥有 JVM 的平台上。如图 1.2 所示为 Prog.java 的开发与执行过程。

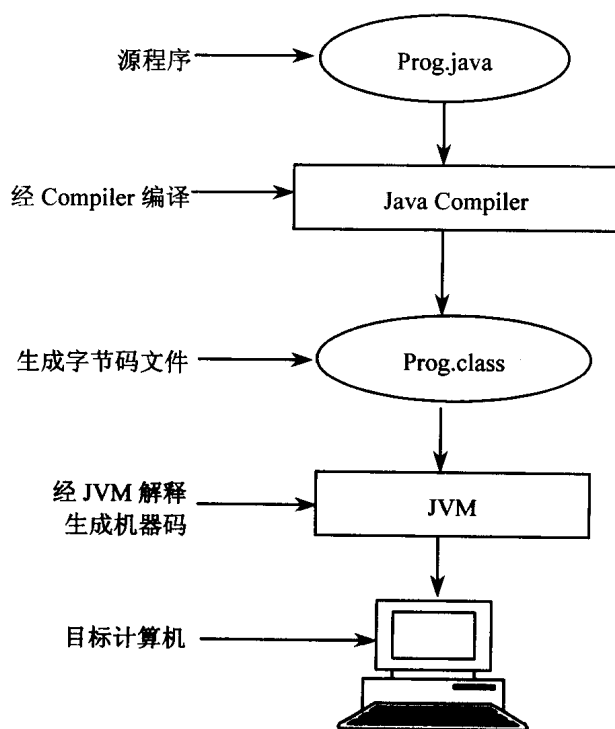


图 1.2 Prog.java 的开发与执行过程

注意：JVM 是平台相关的，不同平台的机器需要不同版本的 JVM。JVM 类似于一个精通各国语言的翻译，它可以相同的一句话翻译成不同国家的语言，而在用户看来，他的语言所有国家的人都可以听懂。

1.4 Java 编程基础

同其他的语言一样，Java 也可用来建立各种程序。下面以一个简单的应用程序和一个简单的小应用程序为例，在屏幕上输出“Hello World!”。

1.4.1 一个简单的 Java 应用程序

1. 编写源文件

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

(1) 应注意的几个问题。

1) 一个 Java 源程序是由若干个类组成的, 上面的这个 Java 应用程序只有一个类, 类名为 Hello。

2) class 是用来定义类的关键字 (注意使用小写)。第一个大括号和最后一个大括号以及它们之间的内容被称为类体。public 是用来修饰类的使用范围的关键字 (在第 3 章中会详细地介绍)。

3) public static void main(String args[]) 是类体中的一个方法, 之后的一对大括号以及之间的内容被称作方法体。main 方法是 Java 应用程序开始执行的入口。在一个类中有且只能有一个 main 方法, 并且 main 方法必须被说明为 public static void, 参数也必须是 String 类型的数组。否则, Java 解释器将无法运行此程序。

在运行程序时, 可以在命令行中给出一些参数, 这些参数可传递到 main() 方法中的 String 类型的名为 args 的数组里, 在此程序中没有使用命令行参数。

4) System.out.println("Hello World!") 的作用是在屏幕上输出双引号中的字符串。Out 是 System 类的一个静态变量, 它的类型是 PrintStream。PrintStream 是 java.io 包中的一个类, println 方法是此类中的一个方法。

5) 保存为 D:\java\Hello.java, 不可以保存成 hello.java 或 Hello.Java。因为 Java 语言是区分大小写的, 且 Java 文件名必须与文件中标记为 Public 的类的类名完全一致。

(2) 源文件的命名规则。若源文件中有多类, 那么只能有一个类是 public 类。如果有一个类是 public 类, 那么源文件的名称必须与这个类的名称完全相同, 扩展名是 .java; 如果源文件没有 public 类, 那么源文件的名称可以任意。但一般情况下要和某个类的名称相同。

在一个 Java 源程序中, 如果有多个类, 那么就可以有多个 main() 方法, 但一个类中有且只能有一个 main() 方法。

2. 编译

创建了 Hello.java 这个源文件后, 可以使用 java 编译器 (javac.exe) 来进行编译, 如下所示:

```
D:\java\>javac Hello.java
```

编译通过后将生成 Hello.class 文件, 它被存放在与源文件相同的目录中。若源程序中包含了多个类, 则会生成多个 .class 文件。

注意: 若源程序被修改了, 则需要重新编译。

3. 运行

使用 Java 解释器 (java.exe) 运行这个应用程序, 如下所示:

```
D:\java\>java Hello
```

注意: 当运行时, 一定要保证运行的类里含有 main 方法。

屏幕将显示如下信息:

```
Hello World!
```

图 1.3 以 Hello.java 为例, 给出了 Java 应用程序的编译 (使用 javac 命令)、执行 (使用 java 命令) 过程, 以及通过使用不同版本的 JVM 达到跨平台的目的。