

21世纪高等院校计算机系列规划教材

Java

程序设计与实践

Java CHENGXU SHEJI YU SHIJIAN

主 编 王 薇 杜 威



华中科技大学出版社

<http://www.hustp.com>

21世纪高等院校计算机系列规划教材

Java

程序设计与实践

Java CHENGXU SHEJI YU SHIJIAN

主 编 王 薇 杜 威

常州大学图书馆
藏书章

内 容 简 介

本书主要从企业的实践技术需求出发,分两部分介绍 Java 核心技术。第一部分的内容是 Java 入门基础,主要包括:Java 简介及开发环境搭建,Java 基础语法,Java 程序流程控制,Java 面向对象编程基础与深入及 Java 输入/输出类操作。第二部分的内容是 Java 高级编程,主要包括:基于 AWT/Swing 的 GUI 图形用户界面技术,多线程编程技术,网络程序开发,数据库连接技术。

本书结构合理,语言简练,内容深入浅出,以案例驱动引述各章节,使读者学而知所用,体现 Java 编程语言的实战性特点。本书可以作为高等院校和培训班相关专业的教材,还可供从事计算机技术、电子商务、系统工程的人员和企业技术人员参考。

图书在版编目(CIP)数据

Java 程序设计与实践/王 薇 杜 威 主编. —武汉: 华中科技大学出版社, 2010. 9
ISBN 978-7-5609-6450-8

I. J… II. ①王… ②杜… 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(2010)第 144897 号

Java 程序设计与实践

王 薇 杜 威 主编

策划编辑: 何 赟

责任编辑: 何 赟

封面设计: 龙文装帧

责任校对: 周 娟

责任监印: 周治超

出版发行: 华中科技大学出版社(中国·武汉)

武昌喻家山 邮编: 430074 电话: (027)87557437

录 排: 华中科技大学惠友文印中心

印 刷: 武汉科利德印务有限公司

开 本: 787mm×1092mm 1/16

印 张: 21

字 数: 533 千字

版 次: 2010 年 9 月第 1 版第 1 次印刷

定 价: 39.80 元



华中出版

本书若有印装质量问题, 请向出版社营销中心调换
全国免费服务热线: 400-6679-118 竭诚为您服务
版权所有 侵权必究

前 言

Java 程序设计是计算机专业的一门专业必修课,Java 课程体系也成为各个高等院校计算机学院、软件学院学生学习的技術主线之一。由于学习 Java 语言具有入门快、开发快、就业快等特点,因此,目前学习 Java 语言的人越来越多。

本教材主要从企业的实践技術需求出发,分两部分介绍 Java 核心技术。第一部分的内容是 Java 入门基础,主要包括:Java 简介及开发环境搭建,Java 基础语法,Java 面向对象编程,Java 异常处理及 Java 输入/输出类操作。第二部分的内容是 Java 高级编程,主要包括:基于 AWT/Swing 的 GUI 图形用户界面技术,多线程编程技术,网络程序开发,数据库连接技术。

现在大多数 Java 程序设计教材单纯从程序设计语言的角度出发,纯粹介绍语言特点及语法规则,忽视了 Java 程序设计语言的应用性;而现在很多大中专院校计算机专业强调学生的实践动手能力,尤其是软件学院,对学生的实践动手能力要求更高,这就需要有相应的实践性强的教材。本书正是以这一市场需求为立足点,以理论要点为基础,以案例驱动引述各章节,使读者学而知所用,体现 Java 编程语言的实战性特点的一本教材。

编者在多年教学经验的基础上,结合企业实训要求,根据学生的认知规律精心组织了本教材的内容,并通过大量的案例,循序渐进地介绍了 Java 语言程序设计的有关概念和编程技巧。

综上所述,本书重项目实践,重理论要点,采用案例驱动式叙述方式,力求体现实战性教材特点,能使读者逐步具备利用 Java 来开发应用程序的能力。教材内容充实,结构合理,每章均配有理论练习题及上机实训题。本书力求在编写过程中突出知识点,强调实践操作,本书集知识性、实践性和操作性于一体,具有内容安排合理、层次清楚、图文并茂、通俗易懂、实例丰富等特点。

本书由长春大学软件学院王薇、杜威主编,全书由王薇统稿,第 1、6 章及附录部分由王薇完成,第 2、3、9 章由长春工业大学马承泽编写,第 4、5 章由杜威编写,第 7、10 章由长春大学计算机学院李晓辉编写,第 8 章由杨丽萍编写,第 11 章由刘玉宝编写。

本书在编写过程中得到了李纯莲教授的大力支持,在此一并表示感谢。由于时间仓促、水平有限,不足之处在所难免,恳请读者批评指正。

编 者

2010 年 4 月

目 录

第一部分 Java 入门基础

第 1 章 Java 简介及开发环境搭建	(3)
1.1 Java 语言简介	(3)
1.2 Java 语言发展历史	(5)
1.3 Java 技术体系架构	(6)
1.4 Java 的工作方式	(8)
1.5 Java 的相关术语	(9)
1.6 Java 开发环境的搭建	(10)
1.7 Java 程序的编写与运行	(18)
1.8 常见问题	(25)
1.9 本章小结	(26)
理论练习题	(26)
上机实训题	(27)
第 2 章 Java 语法基础	(29)
2.1 Java 程序的结构	(29)
2.2 Java 程序的一些特殊语句	(31)
2.3 常量	(34)
2.4 变量和数据类型	(34)
2.5 Java 标识符和关键字	(41)
2.6 运算符和表达式	(42)
2.7 本章小结	(54)
理论练习题	(55)
上机实训题	(56)
第 3 章 程序流程控制	(57)
3.1 程序的流程控制	(57)
3.2 选择结构	(58)
3.3 循环结构	(66)
3.4 跳转语句	(71)
3.5 数组	(73)
3.6 多维数组	(76)

3.7 本章小结	(78)
理论练习题	(79)
上机实训题	(79)
第4章 Java 面向对象编程基础	(80)
4.1 面向对象的基本概念	(80)
4.2 类的定义	(82)
4.3 案例分析: 定义银行账户类	(84)
4.4 对象的定义和引用	(85)
4.5 案例分析: 银行账户对象的创建	(90)
4.6 静态成员与实例成员	(91)
4.7 方法的重载	(95)
4.8 案例分析: 银行账户类构造方法的重载	(96)
4.9 类的封装和访问控制	(97)
4.10 案例分析: 简单的银行账户管理程序	(101)
4.11 本章小结	(107)
理论练习题	(108)
上机实训题	(111)
第5章 Java 面向对象编程深入	(112)
5.1 类的继承	(112)
5.2 案例分析: VIP 银行账户类的创建	(118)
5.3 多态	(121)
5.4 抽象类	(123)
5.5 内部类与外部类	(126)
5.6 包	(127)
5.7 案例分析: 改进的银行账户管理程序	(130)
5.8 接口	(135)
5.9 Java 异常处理机制	(138)
5.10 本章小结	(144)
理论练习题	(145)
上机实训题	(146)
第6章 Java 常用类	(148)
6.1 Java 类库	(148)
6.2 字符串类	(152)
6.3 Math 类与 Random 类	(164)
6.4 Date(日期)类与 Calendar(日历)类	(166)
6.5 Java 基本类型包装器与泛型	(171)
6.6 集合类	(174)
6.7 案例分析: 利用 ArrayList 类存储银行账户类	(182)
6.8 本章小结	(185)

理论练习题	(186)
上机实训题	(189)
第 7 章 Java 输入/输出系统	(190)
7.1 概述	(190)
7.2 字节 I/O 流	(191)
7.3 字符 I/O 流	(198)
7.4 文件访问	(201)
7.5 案例分析: 多种流的实现	(206)
7.6 本章小结	(207)
理论练习题	(208)
上机实训题	(209)

第二部分 Java 高级编程

第 8 章 GUI 图形用户界面	(213)
8.1 AWT 概述	(213)
8.2 常用容器	(215)
8.3 常用 AWT 组件	(221)
8.4 GUI 布局管理器	(235)
8.5 事件处理模型	(245)
8.6 用 Swing 生成图形化用户界面	(255)
8.7 本章小结	(260)
理论练习题	(260)
上机实训题	(262)
第 9 章 线程	(263)
9.1 线程概念	(263)
9.2 线程的创建和使用	(264)
9.3 线程的状态	(270)
9.4 线程的优先级调度	(271)
9.5 本章小结	(273)
理论练习题	(273)
上机实训题	(274)
第 10 章 网络编程	(275)
10.1 网络开发基础	(275)
10.2 URL 编程	(276)
10.3 Socket 通信机制	(280)
10.4 案例分析: C/S 模式下的文件内容传递	(288)
10.5 本章小结	(290)
理论练习题	(291)

上机实训题	(291)
第 11 章 Java 数据库操作	(292)
11.1 数据库基础知识	(292)
11.2 SQL 语言基本语法	(293)
11.3 JDBC 访问数据库	(300)
11.4 案例分析：创建 JDBC-ODBC 连接	(303)
11.5 预处理语句	(306)
11.6 可滚动结果集的操作	(307)
11.7 本章小结	(311)
理论练习题	(312)
上机实训题	(314)
附录 A Tomcat 的安装	(315)
附录 B 在 MyEclipse 8.5 中添加 Tomcat 服务器	(318)
附录 C 二级 Java 试题解析	(322)
参考文献	(326)

第一部分

Java 入门基础

第 1 章 Java 简介及开发环境搭建

第 2 章 Java 语法基础

第 3 章 程序流程控制

第 4 章 Java 面向对象编程基础

第 5 章 Java 面向对象编程深入

第 6 章 Java 常用类

第 7 章 Java 输入/输出系统



第 1 章 Java 简介及开发环境搭建

【教学目标】

- (1) 认识 Java 并了解其发展历史。
- (2) 掌握安装和配置 Java 开发环境的方法。
- (3) 掌握 Java 中 Path 及 classpath 属性的作用。
- (4) 能熟练编写并运行一个简单的 Java 程序。
- (5) 掌握 Java 的开发流程。

【教学重点】

Java 是一门程序设计语言，其本身的发展经过了一系列的过程。本章首先介绍 Java 的发展历程，让读者对 Java 有一个基本的认识，了解 Java 的主要作用，接着详细讲解 Java 的体系架构、工作方式，让读者明确 Java 的相关术语，最后介绍 Java 程序从编写到运行的过程，包括 MyEclipse 8.5 开发环境的搭建。

1.1 Java 语言简介

当 1995 年 Sun 推出 Java 语言之后，全世界的目光都被这个神奇的语言所吸引。Java 正在进入企业、家庭和学校，它正在像 Internet 一样成为普遍存在的技术之一。EarthWeb 公司董事长兼首席执行官 Jack Hidary 曾说过：“Java 正在改变我们使用 Internet 的方法，改变我们在公司内部存取信息的方法，甚至改变我们使用蜂窝电话的方法。它正在从根本上改变制作信息、销售信息和分配信息的方法。”

目前，Java 语言仍然是世界上最受欢迎的编程语言之一，而且是一种面向对象的高级编程语言。

1.1.1 什么是面向对象编程

面向对象程序设计(Object-Oriented Programming, 简称 OOP)思想起源于 20 世纪 60 年代的 Simula 语言，发展至今已经将近 50 年，其自身理论已经十分完善，并被多种面向对象程序设计语言(Object-Oriented Programming Language, 简称 OOPL)实现。

传统的结构化编程语言中，编程代码与数据分开，而面向对象编程则把代码与数据结合在一起。如以“人”为例编程，在传统的编程方法中是在数据定义部分声明人的身高、体重等，再声明函数，如走路、吃饭等，它们每一个只是“人”类的某一个方面，这就把这个“人”分开成几部分去说明了。面向对象编程则把身高、体重等数据部分与吃饭、走路等代码部分包装在一起，形成一个“类”，就如同平时所说的人类一样，具有身高、体重这些特征，同时也具有吃饭、走路这些行为动作。再比如在结构化编程语言中，一辆车被描述为一组函数

用来说明启动、刹车、停车、加速等，用一组独立的变量来定义车的颜色、门的号码、制造日期、型号等。只要初始化这些变量并调用一些函数来操作这些变量，就相当于创建了一辆车。而 OOP 语言则将车看成由它的行为(功能)和保持车状态的变量(数据值)所构成的一个整体，即为“车”这个类，可以简单的创建这个类的对象，即具体化“车”类的那些变量。

本质上，面向对象编程是以一组对象(如：车)，而不是将功能(如：停车)和变量(如：门的数目)分开来考虑的。这就使得程序员可以从现实世界客观存在的事物(即对象)出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。有关面向对象的知识将在第 4 章作详细介绍。

1.1.2 为什么学习 Java

1. Java 的特点

Java 语言作为一种广泛使用的程序设计语言，具有以下八大优势。

(1) 广泛性。Java 是目前使用最为广泛的网络编程语言之一。

(2) 简单性。Java 语言简单是指这门语言既易学又好用。学习过 C++ 语言的人会感觉 Java 很眼熟，因为 Java 中许多基本语句的语法和 C++ 一样，但不要误解为 Java 是 C++ 的增强版。Java 和 C++ 是两种完全不同的语言，它们各有各的优势，将会长期并存下去，Java 语言和 C++ 语言已成为软件开发应当掌握的基本语言。如果从语言的简单性方面看，Java 要比 C++ 简单，C++ 中许多容易混淆的概念，或者被 Java 弃之不用了，或者以一种更清楚更容易理解的方式实现，例如，Java 不再有指针的概念。

(3) 面向对象性。Java 语言与其他面向对象语言一样，引入了类的概念，类是用来创建对象的模板，它包含被创建的对象的状态描述和方法的定义。

(4) 平台无关性。与平台无关是 Java 语言最大的优势。其他语言编写的程序面临的一个主要问题是操作系统的变化，处理器升级以及核心系统资源的变化，这些都可能导致程序出现错误或无法运行。Java 虚拟机成功地解决了这个问题，Java 编写的程序可以在任何安装了 Java 虚拟机的计算机上正确运行，Sun 公司实现了自己的目标，即“一次写成，处处运行”。

(5) 解释性。Java 不像 C++，它不针对特定的 CPU 芯片进行编译，而是把程序编译为一个字节码文件。字节码是一种中间代码，很接近于机器码。字节码文件可以在提供了 Java 虚拟机的任何系统上被解释执行。Java 被设计成为解释执行的程序，即翻译一句，执行一句，不产生整个的机器代码程序。翻译过程如果不出现错误，就一直进行到完毕，否则将在错误处停止执行。同一个程序，如果是解释执行的，那么它的运行速度通常比编译为可执行的机器代码的运行速度慢一些。Sun 公司在 Java 2 发行版中提供了这样一个字节码编译器——JIT(Just In Time)，它是 Java 虚拟机的一部分。Java 的字节码经过仔细设计，很容易便能使用 JIT 即时编译方式，将字节码直接转化成高性能的本地机器码，因而“高效且跨平台”对 Java 来说不再矛盾。如果把 Java 的程序比做“汉语”，那么字节码就相当于“世界语”，世界语不和具体的“国家”相关，只要这个“国家”提供了“翻译”，就可以快速地把世界语翻译成本地语言。

(6) 多线程。Java 的特点之一就是内置对多线程的支持。多线程允许同时完成多个任务。实际上多线程使人产生多个任务在同时执行的错觉，因为目前计算机的处理器在同一时刻只

能执行一个线程,但处理器可以在不同的线程之间快速地切换,由于处理器速度非常快,远远超过了人接收信息的速度,所以给人的感觉好像多个任务在同时执行。C++没有内置的多线程机制,因此必须调用操作系统的多线程功能来进行多线程程序的设计。

(7) 安全性。当用户从网络上下载一个程序时,最大的担心是程序中含有恶意的代码,比如试图读取或删除本地机上的一些重要文件,甚至该程序是一个病毒程序等。当使用支持 Java 的浏览器时,则可以放心地运行 Java 的小应用程序 Java Applet,不必担心病毒的感染和恶意的企图,Java 小应用程序将限制在 Java 运行环境中,不允许它访问计算机的其他部分。

(8) 动态性。Java 程序的基本组成单元就是类,有些类是自己编写的,有一些是从类库中引入的,而类又是运行时动态装载的,这就使得 Java 可以在分布环境中动态地维护程序及类库,而不像 C++那样,每当其类库升级之后,相应的程序都必须重新修改、编译。

2. Java的应用范围

Java 语言有着广泛的应用前景,大体上可以从以下几个方面来考虑其应用。

- (1) 所有面向对象的应用开发,包括面向对象的事件描述、处理、综合等。
- (2) 计算过程的可视化、可操作化的软件开发。
- (3) 动态画面的设计,包括图形图像的调用。
- (4) 交互操作的设计(选择交互、定向交互、控制流程等)。
- (5) Internet 的系统管理功能模块的设计,包括 Web 页面的动态设计、管理和交互操作设计等。
- (6) Intranet(企业内部网)上的软件开发(直接面向企业内部用户的软件)。
- (7) 与各类数据库连接查询的实现。
- (8) 嵌入式软件的开发。
- (9) 其他应用类型的程序的开发。

1.2 Java 语言发展历史

Java 语言最早诞生于 1991 年,来自于 Sun(全称为 Stanford University Network,1982 年成立,Sun 公司的 Logo 见图 1-1)公司的一个叫 Green 的项目,其研发目的是为家用消费电子产品开发一个分布式代码系统,通过把 E-mail 发给电冰箱、电视机等家用电器,对它们进行控制和信息交流。开始时准备采用 C++来开发该项目,但 C++太复杂,安全性差,最后基于 C++开发了一种新的语言 Oak(Java 的前身)。Oak 是一种用于网络的精巧而安全的语言,Sun 公司曾以此投标一个交互式电视项目,但结果被 SGI 打败。可怜的 Oak 几乎无家可归,恰巧这时 Mark Andreesen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员,他们用其编制了 HotJava 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,触发了 Java 进军 Internet。Java 语言于 1995 年 5 月 23 日正式发布推广。

Java 的取名也有一则趣闻,有一天,几位 Java 项目组的成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,有一个人灵机一动说就叫 Java 怎样,这得到了其他人的赞赏,于是,Java 这个名字就这样传开了,Java 的标志如图 1-2 所示,也

表示了其名字的来历。



图 1-1 Sun 公司 Logo



图 1-2 Java 标志

到 2009 年年中为止, Java 已经发布了一系列的版本, 以下列举其中比较重要的 Java 历史事件。

1995 年 5 月 23 日, Java 语言诞生。

1996 年 1 月, 第一个 JDK—JDK1.0 诞生。

1996 年 4 月, 10 个最主要的操作系统供应商申明将在其产品中嵌入 Java 技术。

1996 年 9 月, 约 8.3 万个网页应用了 Java 技术来制作。

1997 年 2 月 18 日, JDK1.1 发布。

1997 年 4 月 2 日, JavaOne 会议召开, 参与者逾 1 万人, 创下当时全球同类会议规模的最高纪录。

1997 年 9 月, JavaDeveloperConnection 社区成员超过 10 万。

1998 年 2 月, JDK1.1 被下载超过 200 万次。

1998 年 12 月 8 日, Java2 企业平台 J2EE 发布。

1999 年 6 月, Sun 公司发布 Java 的三个版本: 标准版、企业版和微型版(J2SE、J2EE、J2ME)。

2000 年 5 月 8 日, JDK1.3 发布。

2000 年 5 月 29 日, JDK1.4 发布。

2001 年 6 月 5 日, NOKIA 宣布, 到 2003 年将出售 1 亿部支持 Java 的手机。

2001 年 9 月 24 日, J2EE1.3 发布。

2002 年 2 月 26 日, J2SE1.4 发布, 自此 Java 的计算能力有了大幅提升。

2004 年 9 月 30 日, J2SE1.5 发布, 这是 Java 语言的发展史上的又一里程碑事件。为了表示这个版本的重要性, J2SE1.5 更名为 J2SE5.0。

2007 年 1 月, J2SE1.6 发布, 且更名为 Java SE 6.0, 正在开发中的版本是 J2SE1.7 版本。

1.3 Java 技术体系架构

人们很容易把 Java 当作是仅仅开发各种应用程序的编程语言——用来编写源文件并将它们编译为字节码。而事实上, 作为编程语言只是 Java 的众多用途之一, 而形成 Java 众多优点(包括平台独立性)的是其底层架构。

完整的 Java 体系结构实际上是由如下 4 个组件组合而成的。

(1) Java 编程语言。

(2) Java 类文件格式。

(3) Java API(Java Application Program Interface, 即 Java 应用程序编程接口)。

(4) JVM(Java Virtual Machine, 即 Java 虚拟机)。

使用 Java 开发程序,就是用 Java 编程语言编写代码,然后将代码编译为 Java 类文件,接着在 JVM 中执行类文件。

JVM 与核心类 API 共同构成了 Java 平台,也称为 JRE(Java Runtime Environment, Java 运行时环境),该平台可以建立在任意操作系统上。图 1-3 所示的是 Java 不同功能模块之间的相互关系,以及它们与应用程序、操作系统之间的关系。

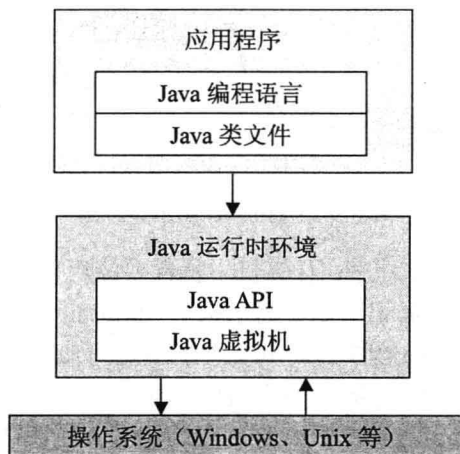


图 1-3 Java 各组件功能概述

Java API 是预先编写的代码,并按相似主题分成多个包。Java API 主要分为三大平台。

(1) J2SE(Java 2 Platform, Standard Edition): 该平台中包含核心 Java 类和 GUI 类。

(2) J2EE(Java 2 Platform, Enterprise Edition): 该包中包含开发 Web 应用程序所需的类和接口,有 Servlet、JavaServer Page 以及 Enterprise JavaBean 类等。

(3) J2ME(Java 2 Platform, Micro Edition): 该包体现了 Java 的传统优势,为消费类产品提供了一个已优化的运行环境,用于如传呼机、手机或汽车导航系统相关软件的设计。

根据三类 Java API 形成了 Java 语言的三个发展方向,但是这三门技术中核心的部分是 J2SE,而 J2ME 和 J2EE 是在 J2SE 基础之上发展起来的,三种技术的关系如图 1-4 所示。另外要提醒读者注意的是,在 2005 年 Java 十周年大会之后这三门技术又重新更名:J2SE 更名为 Java SE;J2ME 更名为 Java ME;J2EE 更名为 Java EE。

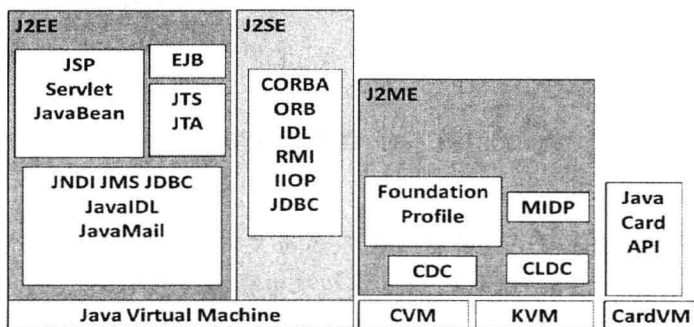


图 1-4 三种技术的关系示意图

1.4 Java 的工作方式

计算机高级语言类型主要有编译型和解释型两种, Java 是两种类型的集合, 在 Java 中处理代码的过程如图 1-5 所示。

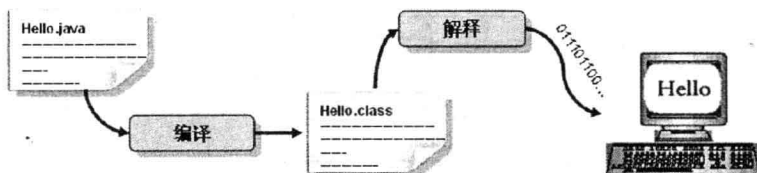


图 1-5 Java 程序的运行机制

从图 1-5 中可以看出, Java 程序在计算机中执行要经历以下几个阶段。

- (1) 使用文字编辑软件(例如记事本、写字板、UltraEdit 等)或集成开发环境(JCreator、Eclipse、MyEclipse 等)编辑 Java 源文件, 其文件扩展名为 .java。
- (2) 通过编译使 .java 的文件生成一个同名的 .class 文件。
- (3) 通过解释方式将 .class 的字节码文件转变为由 0 和 1 组成的二进制指令执行。

从以上阶段可以看出, Java 程序的执行包括了编译和解释两种方式。Java 程序执行的具体过程如图 1-6 所示。

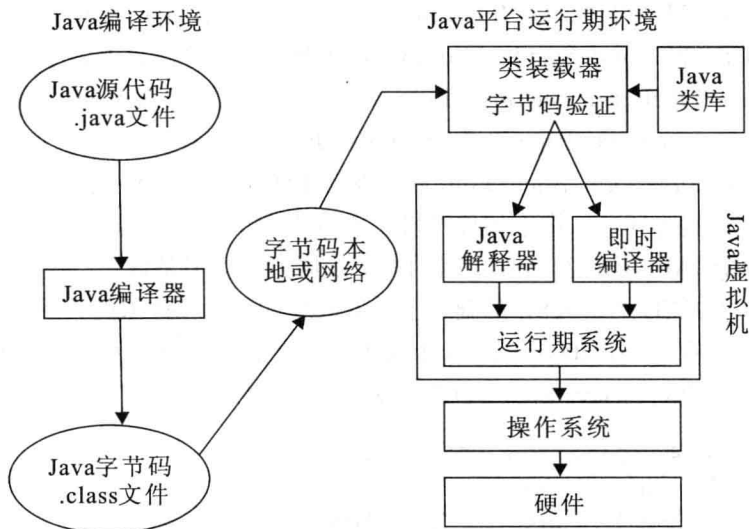


图 1-6 Java 程序的运行机制

知识提示

因为在 Web 应用中 JSP 文件需要转换成 Servlet 文件, 这个 Servlet 文件还需要编译成可以在 JRE 上执行的 .class 文件, 因此在 Java 虚拟机中要有即时编译器。

Java 的 .class 文件是在一个称为 Java 虚拟机(Java Virtual Machine—JVM)的上面运行的。JVM 是在一台计算机上由软件或硬件模拟的计算机, JVM 可以实现 Java 程序的跨平台运行,

即运行的操作平台各不相同。有 JVM 的存在, 就可以将 Java 的.class 文件转换为面向各个操作系统的程序, 再由 Java 解释器执行。这就如同有一个中国富商, 他同时要和美国、韩国、俄罗斯、日本、法国、德国等几个国家洽谈生意, 可是他不懂这些国家的语言, 所以他针对每个国家请了一个翻译, 他说的话就只对翻译说, 不同的翻译会将他说的话翻译给不同国家的客户, 这样富商只需要说一句话给翻译, 那么就可以同几个国家的客户沟通了。可见 JVM 的作用是读取并处理经编译过的、与平台无关的字节码.class 文件; 而 Java 解释器是负责将 JVM 的代码在特定的平台上运行。JVM 的基本原理如图 1-7 所示。

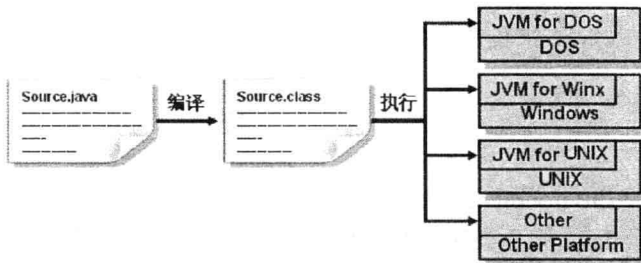


图 1-7 JVM 基本原理

从图 1-7 中可以发现, 所有的.class 文件都是在 JVM 上运行的, 即.class 文件只需要认识 JVM, 由 JVM 再去适应各个操作系统。如果不同的操作系统安装上符合其类型的 JVM, 那么以后程序无论到哪个操作系统上都是可以正确执行的。

1.5 Java 的相关术语

1.5.1 JDK

JDK(Java Development Kit) 称为 Java 开发包或 Java 开发工具, 是一个编写 Java 的 Applet 小程序和应用程序的程序开发环境。JDK 是整个 Java 的核心, 包括了 Java 运行环境(Java Runtime Environment), 一些 Java 工具和 Java 的核心类库(Java API)。不论什么 Java 应用服务器实质都是内置了某个版本的 JDK。主流的 JDK 是 Sun 公司发布的 JDK, 除了 Sun 之外, 还有很多公司和组织都开发了自己的 JDK, 例如, IBM 公司开发的 JDK, BEA 公司的 Jrocket, 还有 GNU 组织开发的 JDK, 等等。其中 IBM 的 JDK 包含的 JVM 运行效率要比 Sun JDK 包含的 JVM 的效率高出许多。

作为 JDK 实用程序, 工具库中有如下七种主要程序。

- (1) javac: Java 编译器, 将.java 源代码文件转换成.class 字节码文件。
- (2) java: Java 解释器, 直接解释执行 Java 字节码文件。
- (3) appletviewer: 小应用程序浏览器, 一种执行 HTML 文件上的 Java 应用小程序的 Java 浏览器。
- (4) javadoc: 根据 Java 源码及说明语句生成 HTML 文档。
- (5) jdb: Java 调试器, 可以逐行执行程序, 设置断点和检查变量。