

全国高职高专规划教材

Java程序设计

Programming Java

欧阳江林 主 编
汪志达 马荣飞 副主编



科学出版社

www.sciencep.com



全国高职高专规划教材

Java 程 序 设 计

欧阳江林 主 编

汪志达 马荣飞 副主编

科 学 出 版 社

北 京

内 容 简 介

本书以通俗易懂的语言,全面介绍了 Java 语言的功能和特点,通过大量实例介绍 Java 的编程技巧,解释了 Java 面向对象的编程思想。主要内容包括 Java 语言基础,Java 面向对象编程基础,Java 常用工具,Applet 与图形用户界面,Java 网络设计及相关技术,J2ME 无线应用编程,Java 与 Oracle 数据库。

本书可作为高职高专计算机及相关专业程序设计课程的入门教材,也可作为各学校程序设计公共选修课的教材,还可用于职业培训,供从事软件开发读者学习使用。

图书在版编目(CIP)数据

Java 程序设计/欧阳江林主编. —北京:科学出版社,2003

(全国高职高专规划教材)

ISBN 7-03-012015-9

I. J... II. 欧... III. JAVA 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 067429 号

策划编辑:李振格/责任编辑:韩 洁

责任印制:吕春珉/封面设计:东方人华平面设计部

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2003年8月第 一 版 开本:787×1092 1/16

2003年8月第一次印刷 印张:14 3/4

印数:1—5 000 字数:347 000

定价:20.00 元

(如有印装质量问题,我社负责调换<长虹>)

全国高职高专规划教材编委会名单

主 任 俞瑞钊

副主任 陈庆章 蒋联海 周必水 刘加海

委 员 （以姓氏笔画为序）

王 雷 王筱慧 方 程 方锦明 卢菊洪 代绍庆

吕何新 朱 炜 刘向荣 江爱民 江锦祥 孙光弟

李天真 李永平 李良财 李明钧 李益明 余根墀

汪志达 沈凤池 沈安衢 张 元 张学辉 张锦祥

张德发 陈月波 陈晓燕 邵应珍 范剑波 欧阳江林

周国民 周建阳 赵小明 胡海影 秦学礼 徐文杰

凌 彦 曹哲新 戚海燕 龚祥国 章剑林 蒋黎红

董方武 鲁俊生 谢 川 谢晓飞 楼 丰 楼程伟

鞠洪尧

秘书长 熊盛新

本书编写人员名单

主 编 欧阳江林

副主编 汪志达 马荣飞

撰稿人 杨昱曷 潘修强 杨林 杨端荣 朱锦晶 陈小冬

前 言

Java语言作为Internet编程的重要工具，具有其他编程语言无可比拟的优点。首先，Java语言具有面向对象性、平台无关性、安全性和健壮性等特点，内置了多线程和网络支持能力，是网络编程的首选语言；其次，Internet的迅速发展为Java的推广和普及拓展了广阔的空间。目前，许多大中专院校都把“Java程序设计”作为计算机及其相近专业的必修课或选修课。

本书的作者有多年的Java教学经验和应用、开发经验。作者力求从实用的角度阐述Java面向对象的编程思想，通过实例介绍了Java的编程技巧，注重理论和实践的有机结合。在内容的安排上，按照循序渐进、由浅入深的原则，兼顾系统性、实用性和新颖性。第1章介绍了Java语言基础知识，包括Java开发工具、Applet和Application、基本数据类型、结构化程序设计等。第2章介绍了类、包、接口等的声明和使用方法。第3章介绍了Java的类库、数组及向量类和字符串类。第4章介绍了Applet及图形用户界面的标准组件等。第5章介绍了流式输入输出、多线程机制、网络应用程序的实现方法。第6章介绍了J2ME无线应用编程的基础知识，包括J2ME平台及体系结构、J2ME的用户界面编程和J2ME数据库编程等。第7章介绍了Oracle数据库基础、SQL语言基础、Java程序对Oracle数据库的访问与操作等。

本书的第1章由陈小冬和朱锦晶编写，第2章由汪志达编写，第3章由杨昱曷编写，第4章由马荣飞编写，第5章由杨端容和杨林编写，第6章由潘修强编写，第7章由欧阳江林编写，全书由欧阳江林和潘修强统稿。

由于时间仓促，编者水平有限，书中可能存在疏漏和不妥之处，敬请广大读者批评指正。

编 者

目 录

第 1 章 Java 语言基础.....	1
1.1 Java 语言概述.....	1
1.1.1 Java 语言的特点和用途.....	1
1.1.2 Java 开发工具.....	3
1.1.3 Java 程序开发步骤.....	4
1.1.4 用于 Web 的 Applet.....	6
1.1.5 Application.....	6
1.1.6 一个实用的 Applet.....	7
1.2 Java 程序的输入输出.....	8
1.2.1 Applet 输入输出.....	9
1.2.2 Application 输入输出.....	11
1.3 变量与数据类型.....	14
1.3.1 基本数据类型.....	14
1.3.2 常量.....	15
1.3.3 变量.....	16
1.4 运算符与表达式.....	18
1.4.1 运算符.....	18
1.4.2 表达式与运算符的优先级.....	24
1.5 算法与结构化程序设计.....	25
1.5.1 Java 程序结构.....	25
1.5.2 Java 语句.....	26
1.6 分支控制结构.....	27
1.6.1 if 语句.....	27
1.6.2 嵌套的 if 语句.....	28
1.6.3 switch 开关语句.....	29
1.6.4 一个实例.....	30
1.7 循环控制语句.....	31
1.7.1 for 循环语句.....	31
1.7.2 while 循环.....	31
1.7.3 do-while 循环.....	32
1.7.4 循环语句嵌套.....	32
1.7.5 跳转语句.....	33

1.7.6 一个实例.....	34
1.8 定义和使用 Java 的方法.....	35
1.8.1 方法的定义.....	35
1.8.2 方法的调用.....	36
1.8.3 一个实例.....	36
小结.....	36
习题.....	37
第 2 章 Java 面向对象编程基础	38
2.1 定义类.....	38
2.1.1 类的修饰.....	38
2.1.2 类体.....	40
2.1.3 类的构造方法.....	40
2.1.4 成员变量.....	41
2.1.5 成员方法.....	41
2.1.6 消息传递.....	43
2.2 创建类的实例对象.....	43
2.2.1 创建对象.....	43
2.2.2 使用对象.....	44
2.2.3 清除对象.....	44
2.2.4 一个应用实例.....	44
2.3 构造函数.....	45
2.3.1 构造函数的特点.....	45
2.3.2 定义构造函数.....	46
2.4 访问控制修饰符.....	47
2.4.1 public.....	47
2.4.2 private.....	47
2.4.3 protected.....	48
2.5 静态属性、静态方法和静态初始化器.....	48
2.5.1 静态属性.....	48
2.5.2 静态方法.....	48
2.5.3 静态初始化器.....	50
2.6 抽象类与抽象方法.....	50
2.6.1 抽象类.....	50
2.6.2 抽象方法.....	51
2.7 最终类、最终属性、最终方法和终结器.....	53
2.7.1 最终类.....	53
2.7.2 最终属性.....	53
2.7.3 最终方法和终结器.....	53
2.8 其他修饰符及修饰符的混合使用.....	54

2.8.1 其他修饰符.....	54
2.8.2 修饰符的混合使用.....	55
2.9 继承与重载	56
2.9.1 类的继承.....	56
2.9.2 类的多态.....	57
2.10 包	59
2.10.1 包的声明.....	59
2.10.2 包的引用.....	59
2.10.3 Java 的系统程序包.....	59
2.11 接口	61
2.11.1 接口的声明.....	61
2.11.2 接口的继承关系.....	62
2.11.3 在类中实现一个接口	62
2.11.4 在类中实现多个接口	63
2.12 错误、异常及其处理	64
2.12.1 错误与异常.....	64
2.12.2 异常类	65
2.12.3 异常的处理.....	67
2.12.4 自定义异常.....	68
2.12.5 抛出异常.....	70
小结.....	71
习题.....	72
第3章 常用工具.....	73
3.1 Java 的类库	73
3.1.1 语言基础类库.....	73
3.1.2 Object 类.....	74
3.2 数组与向量类.....	74
3.2.1 数组	74
3.2.2 向量类	75
3.3 字符串类.....	77
3.3.1 字符串类的构造方法.....	77
3.3.2 String 类的应用.....	78
3.3.3 StringBuffer 类的应用.....	80
3.4 其他常用工具.....	81
3.4.1 标准输入输出.....	81
3.4.2 数学函数类 Math	82
3.4.3 日期类	84
3.4.4 随机数类 Random	85
小结.....	87

习题.....	88
第 4 章 Applet 与图形用户界面	89
4.1 Applet 基本工作原理.....	89
4.1.1 Applet 简介.....	89
4.1.2 Applet 的生命周期.....	89
4.2 使用 Applet 类.....	90
4.2.1 Applet 类的主要方法.....	90
4.2.2 应用实例.....	90
4.3 Applet 与 HTML 文件的配合.....	92
4.3.1 Applet 标签属性.....	92
4.3.2 HTML 和 Applet 的参数传递.....	93
4.3.3 Applet 与 Application 的合并运行.....	94
4.4 Java 图形用户界面的标准组件.....	96
4.4.1 标签.....	96
4.4.2 按钮.....	97
4.4.3 选项框.....	98
4.4.4 复选框和选项按钮.....	99
4.4.5 列表框.....	101
4.4.6 文本框.....	102
4.4.7 文本区.....	103
4.4.8 滚动条.....	105
4.5 组件布局管理.....	106
4.5.1 顺序布局.....	106
4.5.2 边界布局.....	107
4.5.3 卡片布局.....	108
4.5.4 网络布局.....	109
4.5.5 网格包布局.....	110
4.5.6 面板的使用.....	112
4.5.7 手工布局.....	113
4.6 事件处理.....	114
4.6.1 Java 的事件处理机制.....	114
4.6.2 事件处理实例.....	117
4.7 窗口、菜单和对话框.....	121
4.7.1 窗口.....	121
4.7.2 菜单.....	122
4.7.3 对话框.....	128
4.8 图形处理.....	131
4.8.1 基本图形.....	131
4.8.2 画布.....	136

4.8.3 文字输出.....	137
4.8.4 绘图模式控制.....	140
4.9 多媒体编程.....	141
4.9.1 图像.....	141
4.9.2 动画处理.....	147
4.9.3 数字音频.....	149
小结.....	151
习题.....	151
第5章 Java 网络程序设计及相关技术.....	151
5.1 流式输入输出与文件处理.....	151
5.1.1 流的分类.....	151
5.1.2 数据流的应用.....	151
5.1.3 文件类.....	154
5.1.4 实例.....	155
5.2 Java 多线程机制.....	156
5.2.1 创建线程.....	157
5.2.2 线程的生命周期.....	159
5.2.3 线程的优先级.....	160
5.2.4 线程同步.....	160
5.2.5 多线程问题.....	163
5.2.6 实例.....	164
5.3 网络应用的层次和结构.....	169
5.3.1 Java 网络基础知识.....	169
5.3.2 URL 编程.....	171
5.3.3 实例.....	174
5.4 用 Java 实现底层网络通信.....	174
5.4.1 利用 Socket 和服务器交互.....	174
5.4.2 利用数据报和服务器交互.....	177
5.4.3 实例.....	178
小结.....	181
习题.....	181
第6章 J2ME 无线应用编程.....	183
6.1 Java 2 平台微缩版.....	183
6.1.1 J2ME 平台.....	183
6.1.2 J2ME 体系结构.....	183
6.1.3 K 虚拟机.....	184
6.2 J2ME 开发环境.....	184
6.2.1 J2ME Wireless Toolkit 的硬件和软件需求.....	184
6.2.2 J2ME Wireless Toolkit 的安装.....	184

6.2.3	运行例子程序.....	186
6.2.4	创建 J2ME Wireless Toolkit 应用程序.....	186
6.3	CLDC 和 MIDP API 简介.....	188
6.3.1	CLDC API 简介.....	188
6.3.2	MIDP API 简介.....	188
6.4	J2ME 的用户界面编程.....	188
6.4.1	如何使用 Display 类.....	188
6.4.2	使用 Screen 类.....	189
6.4.3	使用命令按钮响应用户操作.....	189
6.4.4	使用文本框输入信息.....	190
6.4.5	一个简单实例.....	190
6.5	J2ME 数据库编程.....	191
6.5.1	RMS 的表——Record Store.....	192
6.5.2	在 Record Store 中操作记录.....	193
6.5.3	创建多列表格.....	196
6.5.4	记录的排序.....	198
6.5.5	使用 RecordFilter 查找记录.....	199
6.5.6	一个简单实例.....	199
6.6	J2ME 网络编程.....	202
6.6.1	J2ME 的网络连接框架.....	202
6.6.2	HTTP 通信概述.....	203
6.6.3	用 HTTP 发送接收数据.....	204
6.6.4	一个简单实例.....	205
小结		210
习题		210
第 7 章	Java 与 Oracle 数据库.....	211
7.1	关系数据库简介.....	211
7.1.1	关系数据库.....	211
7.1.2	Oracle9i 简介.....	212
7.2	SQL 语言简介.....	212
7.2.1	SQL 语言简介.....	212
7.2.2	SELECT 语句.....	213
7.3	Java 程序对数据库的访问与操作 (JDBC).....	217
7.3.1	数据库应用模型.....	217
7.3.2	JAVA 通过 JDBC 访问 Oracle 数据库.....	218
7.3.3	实例.....	220
7.4	Java 网络操作中的安全性限制.....	222
小结		222
习题		222

第 1 章 Java 语言基础

本章要点

- Applet 输入输出
- Application 输入输出
- 基本数据类型
- 常量及变量
- 运算符与表达式
- 分支控制结构
- 循环控制结构

本章难点

- Java 程序开发步骤
- 用于 Web 的 Applet
- Application
- 定义和使用 Java 的方法

1.1 Java 语言概述

1.1.1 Java 语言的特点和用途

Java 语言是当今流行的新兴网络编程语言，Java 语言特别适合于用来开发网络上的应用程序；另外，作为一种面世较晚的语言，Java 也集中体现和充分利用了若干当代软件技术新成果，如面向对象、多线程等，这些也都在它的特点中有所反映。

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：简单性、面向对象、分布式、解释型、可靠、安全、平台无关、可移植、高性能、多线程、动态性等。下面我们将重点介绍 Java 语言的面向对象、平台无关、分布式、多线程、可靠和安全等特性。

1. 平台无关性

Java 是平台无关的语言是指用 Java 写的应用程序不用修改就可可在不同的软硬件台上运行。平台无关有两种：源代码级和目标代码级。C 和 C++具有一定程度的源代码级平台无关，表明用 C 或 C++写的应用程序不用修改只需中心编译就可以在不同平台上运行。Java 主要靠 Java 虚拟机（JVM）在目标代码级实现平台无关性。JVM 是一种抽象

机器，它附着在具体操作系统上，本身具有一套虚拟机指令，并有自己的栈、寄存器组等。但 JVM 通常是在软件上而不是在硬件上实现的。JVM 是 Java 平台无关的基础，在 JVM 上，有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完软件后，通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配置了 Java 解释器，就可以运行这个程序，而不管这种字节代码是在何种平台上生成的。另外，Java 采用的是基于 IEEE 标准的数据类型。通过 JVM 保证数据类型的一致性，也确保了 Java 的平台无关性。

Java 语言独特的运行机制使得它具有良好的二进制级的可移植性，利用 Java，开发人员可以编写出与具体平台无关，普遍适用的应用程序，大大降低了开发、维护和管理开销。

2. 面向对象

Java 是面向对象的编程语言。面向对象技术较好地适应了当今软件开发过程中新出现的种种传统面向过程语言所不能处理的问题，包括软件开发的规模扩大、升级加快、维护量增大以及开发分工日趋细化、专业化和标准化等，是一种迅速成熟、推广的软件开发方法。面向对象技术的核心是以更接近于人类思维的方式建立计算机逻辑模型，它利用类和对象的机制将数据与其上的操作封装在一起，并通过统一的接口与外界交互，使反映现实世界实体的各个类在程序中能够独立、自治、继承；这种方法非常有利于提高程序的可维护性和可重用性，大大提高了开发效率和程序的可管理性，使得面向过程语言难于操纵的大规模软件可以很方便地创建、使用和维护。C++也是面向对象的语言，但是为了与 C 语言兼容，其中还包含了一些面向过程的成分；Java 去除了 C++中非面向对象的部分，其程序编写过程就是设计、实现类，定义其属性、行为的过程。

3. 分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上，操作分布是指把一个计算分散在不同主机上处理。

Java 支持 WWW 客户机/服务器计算模式，因此，它支持这两种分布性。对于前者，Java 提供了一个叫做 URL 的对象，利用这个对象，你可以打开并访问具有相同 URL 地址上的对象，访问方式与访问本地文件系统相同。对于后者，Java 的 Applet 小程序可以从服务器下载到客户端，即部分计算在客户端进行，提高系统执行效率。

Java 提供了一整套网络类库，开发人员可以利用类库进行网络程序设计。

4. 安全稳定

在网络上的另一个需求是要有较高的安全可靠。用户通过网络获取并在本地运行的应用程序必须是可信赖的，不会是病毒或其他恶意操作的传播者攻击用户本地的资源；同时它还应该是稳定的，轻易不会产生死机等错误，使得用户乐于使用。Java 特有的“沙箱”机制是其安全性的保障，同时它去除了 C++易造成错误的指针，增加了自动内存管理等措施，保证了 Java 程序运行的可靠性。

5. 支持多线程

线程是操作系统的一种新概念，它又被称为轻量进程，是比传统进程更小的可并发执行的单位。C 和 C++采用单线程体系结构，而 Java 却提供了多线程支持。

多线程是当今软件技术的又一重要成果，已成功应用在操作系统、应用开发等多个领域。多线程技术允许同一个程序有两个执行线索，即同时做两件事情，满足了一些复杂软件的需求。

Java 在两方面支持多线程。一方面，Java 环境本身就是多线程的。若干个系统线程运行负责无用单元回收、系统维护等系统级操作；另一方面，Java 语言内置多线程控制，可以大大简化多线程应用程序开发。Java 提供了一个类 Thread，由它负责启动运行，终止线程，并检查线程状态。Java 的线程还包括一组同步原语。这些原语负责对线程实行并发控制。利用 Java 的多线程编程接口，开发人员可以方便地写出支持多线程的应用程序，提高程序执行效率。必须注意的是，Java 的多线程支持在一定程度上受运行时支持平台的限制。例如，如果操作系统本身不支持多线程，Java 的多线程特性可能就表现不出来。

6. 简单易学

如前所述，自 C++衍生的 Java 语言，出于安全稳定性的考虑，去除了 C++中不容易理解和掌握的部分，如最典型的指针操作等，降低了学习的难度；同时 Java 还有一个特点就是它的基本语法与 C 语言几乎一模一样，这样，无论是学过 Java 再学 C，还是已经掌握了 C 语言再来学 Java，都会感到易于入门。

Java 的上述种种特性不但能适应网络应用开发的需求，而且还体现了当今软件开发方法的若干新成果和新趋势。

1.1.2 Java 开发工具

Java 不仅提供了一个丰富的语言和运行环境，而且还提供了一个免费的 Java 开发工具集 (Java Developers Kits, 简称 JDK)，它是 Java 的发明者 SUN 公司免费发行的开发工具。它短小简单，可以很方便地从 Sun 的站点 <http://www.sun.com/product/JDK/> 中获取。JDK 有多个版本，总体来说有 1.02, 1.1.x 和 1.2 三个系列，其中高级版本可以对低级版本实现向下兼容。

JDK 包括以下工具：Java 语言编译器 javac，输出结果为 Java 字节码；Java 字节码解释器 java；Java 字节码分解程序 javap Disassembler，本程序返回 Java 程序的成员变量及方法等信息；资源分析工具 javaprof，用于分析 Java 程序在运行过程中调用了哪些资源，包括类和方法的调用次数和时间，以及各数据类型的内存使用情况等；代码处理工具 javah C，用于从 Java 类调用 C++代码；小应用程序浏览工具 java AppletViewer，用于测试并运行 Java 小应用程序。

Java 开发环境还包括 Java 类库（包括 I/O 类库、用户界面类库、网络类库等）和 HotJava WWW 浏览器，其中，HotJava 浏览器提供了在 WWW 环境下运行 Java 代码的一个运行系统，而且还为 WWW 开发人员提供了一个 Java 开发框架。Java 解释器是面

向 Java 程序的一个独立运行系统,它可以一种稳定、高性能方式运行那些独立于平台的 Java 字节码,Java 编译器则用于生成这些字节码。

前面介绍的 JDK 软件包是一种命令行开发工具,在开发 java 程序,尤其是大型 Java 程序时会使开发人员感到很不方便,所以我们来认识另一种集成化的 Java 开发工具 Visual J++ 6.0,简称 VJ++。所谓集成化开发环境 (IDE) 是指将源代码编辑、字节码编译、调试和运行集中在同一个工具软件中完成的环境,它是现在使用十分广泛并且较为成熟的一种软件开发环境。VJ++ 是微软公司的系列开发工具 Visual Studio 98 中的一员,它不但与其他的 Visual Studio 工具有着相同的前端编辑和编译环境,而且其内置的 Java 虚拟机也以极快的运行速度在所有的 Java 开发工具中名列前茅。VJ++ 是一种比较实用的 Java 集成开发工具,它特有的向导 (Wizard) 功能和可视化开发环境可以帮助开发人员自动生成大量的源代码,提高开发效率、准确性和标准化程度;它包含的对话框编辑器、菜单编辑器和位图编辑器等工具能够大大方便开发工作;强大的调试、排错功能和在线帮助在开发过程中更是不可缺少的;此外,VJ++ 拥有 Visual Studio 系列工具共有的丰富的后台支持,可以方便地生成 COM、ADO 等组件,完成与数据库连接,等等,适合于 Windows 平台上大型网络应用项目的前台开发,尤其适合于后台采用 Microsoft 网络服务器 (NT)、Microsoft Internet 服务器 (IIS) 和 Microsoft 数据库 (SQL Server) 的项目。

1.1.3 Java 程序开发步骤

根据结构组成和运行环境的不同,Java 程序可以分为两类,Java Application 和 Java Applet。简单地说,Java Application 是完整的程序,需要独立的解释器来解释运行;而 Java Applet 则是嵌在 HTML 编写的 Web 页面中的非独立程序,由 Web 浏览器内部包含的 Java 解释器来解释运行。Java Application 和 Java Applet 各自使用的场合也不相同。

Java 程序的开发通常需要编写源代码、编译生成字节码和运行几个步骤。

1. 编写源代码

Java 源程序是以 .java 为后缀的简单的文本文件,可以用各种 Java 集成开发环境中源代码编辑器来编写,也可以用其他文本编辑工具,如 Windows 98 中的记事本或 DOS 中的 Edit 软件等,编程者利用这个文字编辑器将源代码录入到计算机中,形成一个文件保存起来。

在编写 Java 源程序定义类的时候,需要注意下面几个问题:

Java 是区分大小写的语言,关键字的大小写不能搞错,如果把 class 写成 Class 或者 CLASS,都会导致错误。

在一个类的内部不能定义其他的类,即类和类之间是平行的,而非嵌套的关系。

一个程序中可以有多个类,但是只能有一个主类。不同类型的 Java 程序,其主类的标志是不同的。

源程序编写好之后,应该以文件的形式保存在硬盘或软盘上,称为源程序文件,或称为源文件。这个源文件的名称不是随便取的,它必须与程序的主类名一致,并且以 java 为后缀。

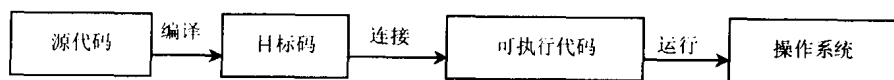
2. 编译生成字节码

高级语言程序从源代码到目标代码的生成过程称为编译。在 Java 程序中源代码经编译所得的目标代码称为字节码。字节码是二进制文件，编程人员无法直接读懂，由 Java 语言的解释器来解释执行。编译字节码需要使用专用的 Java 编译器，在继承化的 Java 开发环境（如 VJ++）中，只要选择一个菜单或单击某一个按钮就可以完成这个编译过程，而在 JDK 这样的命令行开发工具中则需要运行独立的编译程序 `javac.exe`。

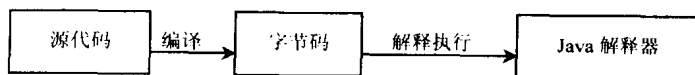
高级语言按照执行模式可以划分为编译型和解释型两种。编译型的高级语言，如 C、C++ 等，生成的字节码经链接后就成为可以直接执行的可执行代码；而解释型语言，如 Basic、Java 等，其程序不能直接在操作系统级运行，需要一个专门的解释器程序来解释执行。

一般说来，解释型的语言比较简单，执行速度也较慢，但在网络应用平台中，却有着一个重要的优势。编译型语言是直接作用在操作系统的，所以对运行它的软硬件平台有着较强的依赖性，在一个平台上可以正常运行的编译语言程序在另一个平台上可能完全不能工作，而必须在这个特定平台上将源代码重新编译，从而生成适合这个特定平台的可执行代码。这种可移植性上的不足对于以网络为支撑平台的应用程序将是很大的麻烦，因为网络是由不同软硬件平台的计算机组成的，为了使这些机器都能够顺利运行编译型应用程序，就必须专门为各种不同的平台开发出不同版本的应用程序，同时对于版本升级和维护的工作量也将非常大。

解释型语言为解决这个问题提供了一个全新的思路，Java 就是遵循这个思路设计而成的。由 Java 源代码编译生成的字节码不能直接运行在一般的操作系统平台上，而必须运行在一个称为 Java 虚拟机的操作系统之外的软件平台上。在运行 Java 程序时，首先应该启动这个虚拟机，然后由它来负责解释执行 Java 的字节码。这样，利用 Java 虚拟机就可以把 Java 字节码程序跟具体软硬件平台分隔开来，只要在不同的计算机上安装针对特定具体平台特点的 Java 虚拟机，就可以把这种不同软硬件平台的具体差别隐藏起来。使得 Java 字节码程序在不同的计算机上能够面对相同的 Java 虚拟机，而不必考虑具体的平台差别，从而实现了真正的二进制代码级的跨平台移植。



(a) 编译型语言的运行机制



(b) Java 语言的运行机制

图 1.1 编译型语言与 Java 的不同运行机制

3. 运行

Java 程序可以分为 Java Application 和 Java Applet 两类，这两类程序的运行方式有