21世纪高等学校应用型规划教材 计算机系列

Java程序设计实用教程

JAVA CHENGXUSHEJI SHIYONG JIAOCHENG

吴凤祥 杨 忠 主编

中国电力出版社
www.infopower.com.cn

21世纪高等学校应用型规划教材 计算机系列

Java程序设计实用教程

JAVA CHENGXUSHEJI SHIYONG JIAOCHENG

吴凤祥 杨 忠 主编
郭清华 刘雪清 刘丽华 路保慧 副主编

中国电力出版社
www.infopower.com.cn

内容提要

本书从语言特点、面向对象的方法、应用技术3个方面分4个层次全面介绍了Java语言规范、Java面向对象的机制、Java基本类库、GUI设计、Applet开发、Java的多线程机制、网络环境下的应用开发、Java与数据库的连接（JDBC）等。各部分内容均有大量的实例，每章后设有思考题和上机实验题。

本书在内容组织上遵循教学规律，内容由浅入深、循序渐进，讲解通俗易懂，条理清楚，非常适合于教学与自学。

本书可作为高等院校或其他各类学校的Java语言及技术方面的教材，也可供从事网络技术、软件开发的专业人员参考，或从事软件开发的初学者自学。

图书在版编目（CIP）数据

Java程序设计实用教程 / 吴凤祥等主编. —北京：中国电力出版社，2006

21世纪高等学校应用型规划教材. 计算机系列

ISBN 7-5083-4103-1

I. J... II.吴... HI. 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（2005）第158457号

丛 书 名：21世纪高等学校应用型规划教材·计算机系列

书　　名：Java程序设计实用教程

出版发行：中国电力出版社

地　　址：北京市三里河路6号　　邮政编码：100044

电　　话：（010）68362602　　传　　真：（010）68316497，88383619

本书如有印装质量问题，我社负责退换

服务电话：（010）88515918（总机）　　传　　真：（010）88518169

E-mail：infopower@cepp.com.cn

印　　刷：北京同江印刷厂

开本尺寸：185×260　　**印　　张**：17　　**字　　数**：406千字

书　　号：ISBN 7-5083-4103-1

版　　次：2006年2月北京第1版

印　　次：2006年2月第1次印刷

印　　数：0001—4000册

定　　价：25.00元

编 委 会

前　言

Java 是 Sun 公司 1995 年推出的一种编程语言，具有平台独立性、面向对象、多线程、安全性好等优点，很适合在网络环境中开发，是当今世界上最热门的编程语言。随着 Java 的发展，如今它已不仅仅是一门语言，而是一门技术，包括 Java 的芯片技术、Java 的编译技术、Java 的数据库连接技术、基于 Java 的信息家电联网技术（Jini）、企业信息服务的综合求解方案技术（Enterprise JavaBeans）等，都是当今计算机技术的热点。

Java 作为编程语言最突出的特点是平台独立性，在一个平台上编写的 Java 程序可以在任何其他平台上运行，而不需要将代码本地化。这种平台独立性一直是软件开发的需求和编程人员所追求的目标，到现在为止只有 Java 实现了这一目标。在网络环境下，编程非常需要这种跨平台的特性，因为计算机网络连接的是世界各地的各种计算机，要想使开发的程序能在各个站点运行，必须具有平台独立性。

用 Java 可以开发两种类型的程序，一是独立的桌面程序（Application），Java 系统的类库支持网络的各种协议，通过计算机网络，使用 URL 对象就可以方便地访问远程资源；二是 Java 小程序（Applet），这种程序内嵌于 Web 页中，在支持 Java 的浏览器上运行。

目前我国普通高校以及其他各类院校都先后开设了有关 Java 技术的课程，正在掀起使用 Java 的热潮，从网上购物、网上银行、远程教学、虚拟课程、嵌入技术、企业综合信息服务等都在使用 Java 技术。目前国内 Java 人材还很缺乏，而在国外具有 Java 程序员（Java Programmer Certification）和 Java 开发员（Java Developer Certification）资格证书的人也极易找到工作。

本书是为高等院校及其他各类学校计算机相关专业编写的 Java 语言及技术教材，全书分为 12 章，从语言特点、面向对象的方法、应用技术 3 个方面，分 Java 语言基础、界面设计、网络环境下的应用与开发、高级应用 4 个层次全面介绍了 Java 语言规范、Java 面向对象的机制、Java 基本类库、GUI 设计、Applet 开发、Java 的多线程机制、网络环境下的应用开发、Java 与数据库的连接（JDBC）等。各部分内容均有大量与之相关的实例，每章后设有思考题和上机练习题以配合各章内容的学习。

本书融入了作者多年程序设计的教学经验和使用 Java 开发的经验，全书在内容的组织上遵循教学规律，内容由浅入深、循序渐进，讲解通俗易懂，条理清楚，非常适合于教学与自学。

本书可作为高等院校或其他各类学校相关专业 Java 语言课程的教材，也可供从事网络技术、软件开发的专业人员参考，或从事软件开发的初学者自学。

考虑到教学的需要，作者制作了本书的网络教学环境，以方便网络教学。有关本书的教学计划、教学分析、电子课件、网络教学环境、全部例题源代码、实验题源代码，可从 http://tch.hebau.edu.cn/computer 下载或与作者联系。

由于编者水平所限，书中疏漏和错误之处在所难免，敬请各位读者批评指正。

编者　于河北农业大学

2005 年 10 月

目　　录

第 1 章　Java 概述

Java 是 Sun 公司开发的一种编程语言，也是一门技术。作为编程语言，它具有平台独立性、面向对象、多线程、安全性好等优点，很适合在网络环境中开发，是当今最优秀的编程语言之一。作为一门技术，它包括 Java 的芯片技术、Java 的编译技术、Java 的数据库连接技术、基于 Java 的信息家电联网技术（Jini）、企业信息服务的综合求解方案技术（Enterprise JavaBeans）等，这些都是当今计算机技术的热点。

1.1　Java 的发展

1.1.1　发展历史

Sun 公司从 1991 年开始设计 Java，其最初目的不是用于网络环境下的开发，而是为消费用的电子器件设计一个通用环境，以此来开拓电子类消费产品市场。开始时它采用 C 语言编程，但由于 C 语言的编译过程与硬件相关，这样当硬件芯片更新时，软件必须适应芯片作相应的修改，而电子类产品的各类芯片在不断更新，于是在 C 语言的基础上产生了一种新的语言，以适应这种独立于平台的开发需要，这就是 Java。

由于商业上的种种复杂原因，这些电子产品当时并没能推向市场，Java 也差点夭折。

1993 年，Internet 的 WWW 由字符界面发展到图形界面，极大地加快了 Internet 的发展。从 1994 年起 Sun 决定将 Java 用在 Internet 的 WWW 开发中，并通过 Internet 让软件设计者免费使用 Java，这样就使越来越多的人了解了 Java，使 Java 受到了广泛的重视。

1995 年，Sun 推出了 Java 的第一个版本。由于 Java 适用于在网络环境中的开发，而计算机网络是当今发展最快的技术之一，这就给 Java 的发展创造了一个绝佳的机会。Java 刚推出不久，Netscape 公司宣布在其 Navigator 2.2 的设计中支持 Java，后来 IBM、Microsoft、Oracle、Novell、Borland 等著名公司也相继宣布支持 Java，但由于 Java 运行速度较慢，所以当时并未得到广泛的应用。计算机网络的发展依赖于微型计算机的普及，1997 年后以 PC 为主流的微型机发展到了 Pentium II、Pentium III 并成为微型机的主流，Java 也得到了广泛应用。

1.1.2　编译和执行方式

Java 程序的运行速度今也是 Java 面临的一个问题，这是由它的编译执行方式决定的。一般高级语言的编译过程如图 1-1 所示。

Java 语言是半编译半解释型的语言，它的编译过程是由 Java 编译器将 Java 源程序编译成字节码。字节码不同于机器码，作为机器码，其操作码和操作数有确定的地址，而地址与平台有关，字节码采用“符号地址”，只有到具体平台上经 Java 解释器解释才有确定的地址；字节码不能直接运行，须由 Java 解释器解释执行。其编译过程如图 1-2 所示。

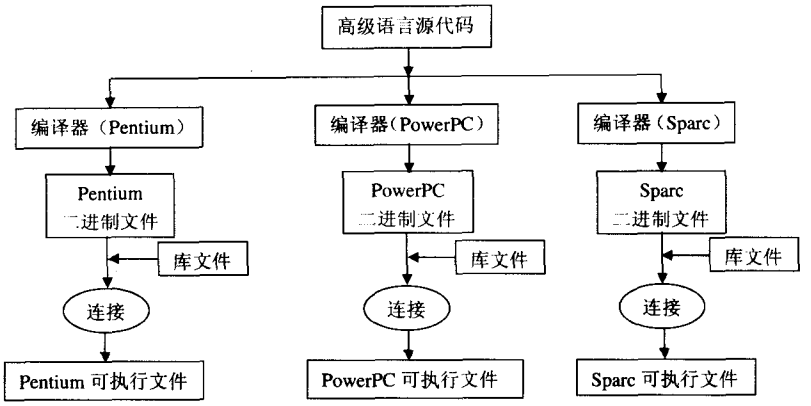

图 1-1　一般高级语言的编译过程

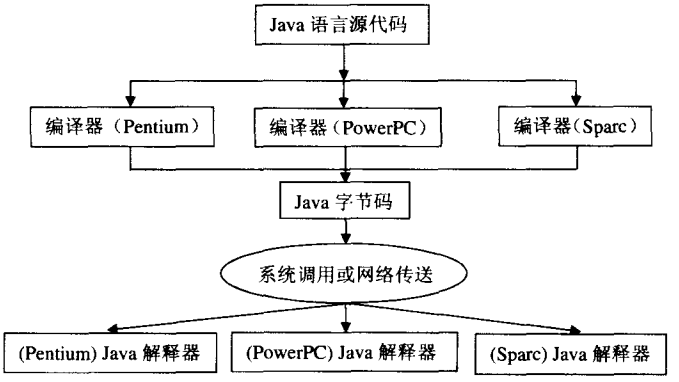

图 1-2　Java 语言的编译过程

从图 1-2 中可以看出，不论在什么平台上编写的 Java 程序，只要装有 Java 解释器，编译生成的字节码在任何一个平台上都能运行。从这个意义上讲，Java 是跨平台的，即相对于 Java 虚拟机而言，Java 程序具有平台独立性。

Java 程序的执行过程如图 1-3 所示。

1.2　Java 虚拟机

Java 虚拟机（Java Virtual Machine，JVM）是一种假想的计算机，它具有真正计算机的功能，是逻辑意义上的。在 Java 推出初期通过软件仿真的方法实现，后来设计出硬件实现的 JVM，主要产品有 Sun 的 Pico Java、Micro Java、Ultra Java 等。不论是软件还是硬件实现，规划一个虚拟机设计应对其逻辑部件作出规范。

从结构上看，JVM 的抽象部件包括如下 6 个部分，JVM 规范对这 6 个部件的功能要求及特点作了具体规定，在此不罗列其细节。

（1）指令系统。规定 Java 支持的基本数据类型、指令格式等，目前用了大约 200 条指令。

（2）寄存器组。作为 Java 虚拟机应能转换 Java 的中间码，使之在不同的机器上运行。为

此，JVM 使用尽可能少的寄存器，只包括程序计数器、堆栈栈顶指针、运行环境指针和局部变量指针。这样可使得只配有少量寄存器的计算机也能建立 Java 运行环境，而对配置较多寄存器的计算机通过优化技术可充分利用资源、提高性能。

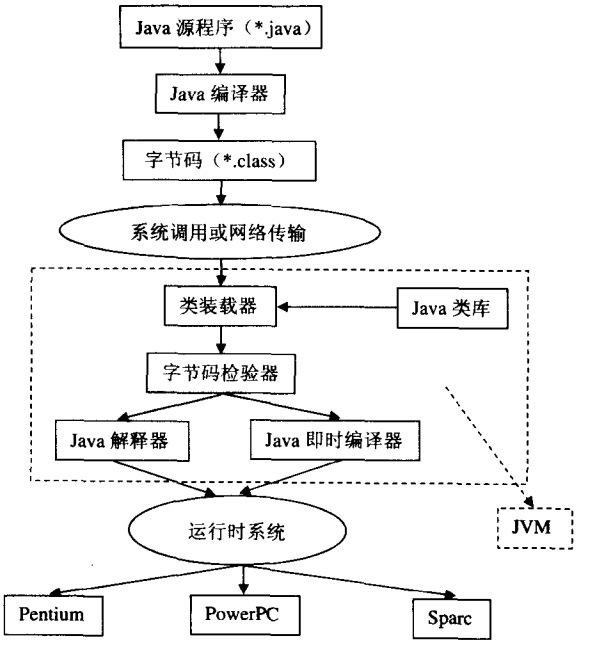

图 1-3　Java 程序的执行过程

（3）类文件格式规定。JVM 的程序代码放在类文件（.class）中，它由编译后的字节码构成，与平台无关。类文件的结构由 4 部分组成：①类文件标识和版本信息；②常数表（包括字符串、域名、类名等）；③所用类及方法的说明；④程序主体。

（4）堆栈。JVM 采用面向堆栈的机制，这为只设置少量寄存器建立了前提。在 JVM 中许多指令的操作数来自堆栈，操作结果也放在堆栈中。

（5）内存垃圾收集器。它是嵌入运行系统的一个附加程序，是系统级的一个线程。其功能是不断地对内存进行扫描，及时释放不再使用的内存区。这种技术使得 Java 程序员在编程时不必关心内存的分配，也不用留心内存的释放问题，由此使程序编写变得容易，而安全性却得到了提高。

（6）存储区。任何一种计算机必备。JVM 的存储区用于存放字节码及各种表格，也为程序翻译和运行提供环境。

凡是符合 JVM 规范的计算机，便可作为 JVM 来运行 Java 软件。所以，凡是符合 JVM 规范的计算机就可以看成是一种 Java 平台，尽管这种平台的实际构成可能各不相同，但都能运行 Java 程序，这就使得 Java 具有了跨平台的特性。

从工作过程上看，JVM 包括以下 3 个部分。

（1）Java 类装载器。Java 源程序经编译生成字节码，再经过系统调用或网络传输到 Java 运行系统。这时首先由类装载器完成字节码的装载（其过程见图 1-3）。运行程序所涉及的所有代码都被装载，在字节码中，不指定操作码和操作数的具体地址，而只包含着地址的“符

号引用”信息，装载时，运行系统通过建立符号引用信息和内存地址之间的对照表，来确定内存的分配。由此可见，Java 是动态分配内存，实际上它采用面向对象的机制，用 new 关键字为每个对象分配内存空间，而且实际内存还会随程序运行情况而改变。

（2）Java 字节码检验器。字节码检验器对字节码进行安全性检查。通过检查可以排除字节码中可能存在的问题，如违反访问权限问题、不规范的数据类型、错误指针堆栈操作、非法调用等。

（3）Java 解释器。Java 程序执行时一般采用解释方式，通过 Java 解释器将字节码翻译成机器码，由即时运行部件立即送往硬件执行。

因为 Java 程序执行时需经装载、校验、解释的过程，所以 Java 在运行速度上并不占优势。若想提高 Java 的执行速度，可采用编译方式，通过 Java 即时编译器，将字节码一次生成适用于本机系统的机器码，然后再送硬件多次执行。只有对软件运行速度要求高的软件才采用这种方式。

1.3 Java 的特点

1. 跨平台

Java 在源程序级确保基本数据类型对各种平台大小一致，所以，基本类型数据的长度与具体平台无关。源程序编译后产生的中间码是一种与具体机器指令无关的指令集合，这种代码通过 JVM 可以在各种平台上运行；Java 提供的类库实现了与不同平台的接口，使得 Java 系统本身也具有可移植性。

2. 面向对象

面向对象技术是当今软件开发采用的先进技术，Java 是一种新型的面向对象的程序设计语言，有以下面向对象的特性。

（1）封装性。Java 以类为基础，类对象为基本组成单元，对象中封装了它的状态变量及相关的方法，很好地体现了模块化和信息隐藏等优良的程序设计思想。

（2）继承性。子类继承父类的数据成员，在此基础上添加新的成员。类继承有利于程序的扩展和代码重用。Java 只提供类的单继承机制，即一个父类可以有多个子类，但一个子类只能有一个父类。在 Java 中引入“界面”的概念，利用界面可以实现多继承的绝大部分功能，也克服了多继承的复杂性。

（3）多态性。多态性指一个类中不同的方法具有相同的名字。Java 通过方法“重载”和方法“重写”实现多态性。方法重载指多个方法具有相同的名字，但参数的个数或类型不同，调用重载方法时，根据传递的参数个数和类型决定调用哪一个方法；方法重写指在继承过程中，子类重新定义父类的方法，实现子类中所需要的功能。利用多态性对一些方法只需定义其方法体，不再取新的名字，既简化方法的实现和调用，又便于记忆。

3. 多线程

多线程程序是指程序的多个逻辑部分可同时运行。

如果一个程序的多个逻辑部分能被同时运行，则每个运行的逻辑部分称为一个线程，如果一个程序在执行过程中按顺序执行各个单个的逻辑部分，即执行完一个方法再调用另外的方法，然后继续运行，直到完成和程序退出，即为单线程，一个单线程的任务就是一个进程；

如果一个程序中有多个线程同时执行不同的任务，则称为多线程，如图 1-4 所示。

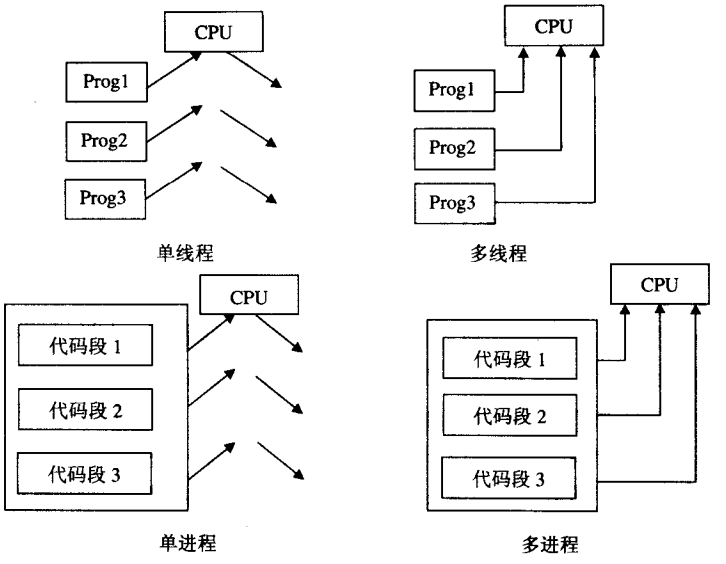

图 1-4　进程和线程

多线程程序可提高系统的输入输出速度、有效利用系统资源、改善计算机通信功能。比如，一个程序运行时需要调用一个子程序来完成一项耗时的操作，在单线程程序中，只有等待该项操作完成，程序的其余部分才能继续运行，而在多线程程序中，可以把这样的操作放到其自身的线程中去，使程序的其他部分继续独立运行。如在网页上嵌入一个 Applet 用于显示图像或播放声音时，由于图片文件和声音数据较大，下载时间较长，要想做到使程序调用方法后立即返回，可启动一个线程在后台进行数据的下载工作，而用另外的线程并发地执行程序的其他部分。

用其他一些语言也能实现多线程程序，但必须靠外部的线程软件包来实现，而 Java 本身是使用线程的思想开发的，它的所有类都是在多线程的思想下定义的。也就是说，Java 从语言级提供对多线程的支持，使在应用程序中创建和使用线程非常容易。

4. 安全性

现在的计算机技术中安全性很重要。Java 从多个方面考虑到了安全性，并设计了安全措施。

（1）从语言级取消了指针操作，为开发人员提供动态内存机制。在一般语言中利用指针可申请或访问内存，虽使编程灵活，但也最容易出问题。Java 不是通过指针而是通过引用访问内存，内存的释放通过内存垃圾收集器实现。

（2）在 JVM 中设置了两道安全屏障：一是类装载器，在装载字节码时，本地类与外来类独立存放，本地类可引用外来类，而外来类不能访问本地类，从而保证了本地数据的安全。二是类检验器，一般来说经编译器编译得到的虚拟机代码符合安全性的规定，但在大型的程序中，可能从其他地方引入代码，另外来自网络的病毒也可能生成字节代码，这两种代码不是由编译器产生的，为了保证这些代码的安全，在进行解释执行之前，再进行安全性检查，以确保虚拟机安全、迅速地执行。

（3）在 Java 的类库中多数较为完善的类都定义为最终的（final），最终类不能派生子类，

这在一定程度上可有效地防止病毒侵入。有些病毒破坏数据的途径是从一些处理关键信息的类派生子类，再用子类代替原来的类，从外部特征看，这些子类可能和通常的类一样，但功能却完全不同了。而最终类不能派生子类，这对安全性是有力的支持。

（4）在多线程程序中，由于多个线程共享系统的数据资源，往往出现数据访问过程中的同步问题。比如下面的代码段：

```
if(currentValue>0){
    ... ...    // 作一些操作
    currentValue - = 1;
}
```

先判断 currentValue 的状态，进行一些操作后再重新赋值，由于这些操作不是一步完成，因而可能出现这样的问题：在前一个线程未完成赋值之前，新的线程又调用了该代码段。譬如，这是一个飞机订票系统，currentValue 表示空余座位数，那么每个售票点都先读出座位数，判断是否满员，不满员时售票后把空余座位数减 1。如果一个售票点在未完成操作之前，另一个售票点也调用了该代码段，则两个售票点就可能售出两张相同的航班票，而最后出现超员现象。这对于用其他语言编写的程序来说，由于自身没有控制同步问题的机制，问题就复杂了，而在 Java 程序中，每个 Java 类对线程都是安全的，Java 设有线程保险箱，对这样的问题可通过在其线程“保险箱”内建代码，一次只允许一个线程使用该代码段，其他线程必须等待，直到这个线程完成。

1.4 应　　用

每门语言都有其最适用的领域，作为 21 世纪的编程语言，Java 的最大特点是跨平台，因而最适用于在不同机型、不同操作系统之间的数据交换和通信、协调控制、综合管理等。具体来讲有如下几个方面。

（1）嵌入技术。Java 嵌入技术是基于 Java 应用于嵌入设备开发的技术。它把计算机制引入设备，如电视、电话、数控机床、PDA 等，从而使之具有可编程的特性。在 Java 嵌入技术中，采用类似标准 Java 的系统结构，由 Java 的核心结合嵌入 JavaAPI，组成新的 Java 程序运行环境，运行在 Java 虚拟机上。

嵌入 JavaAPI 是 Java 嵌入技术的核心，是标准 Java 子集的扩展类库，由 Sun 公司公开发布的嵌入 JavaAPI 包括 PersonalJava、EmbeddedJava、JavaCard、JavaOgibe、JavaTV 等。这些扩展类库覆盖了家用电器、工业设备、金融保险、通信网络等领域，为嵌入 Java 的开发提供了自由的选择机会。

（2）电子商贸。如零售业、网上银行、网上购物、金融支付、税务征收等。

（3）企业综合信息服务。基于服务器的信息服务功能，如销售系统、售后服务等。

（4）教育领域。如远程教学，教学管理等。

（5）可视化图形软件和动画软件的设计。Java 可设计质量很高的活动图形软件，将对计算机图形学、多媒体通信提供良好的支持。

（6）网页制作。利用 Applet 可非常方便地将动画和各种信息嵌入网页。

Java 应用程序有两种类型。一是普通的应用程序（Application），这种程序经编译器编译

后生成字节码文件，然后由解释器解释执行。既可以访问本地文件，也可以访问远程资源。它必须有一个主方法 Main()，作为应用程序运行的起始位置。另一类是 Java 小程序（Applet），这种程序也是先由编译器编译为虚拟机代码，但不能独立运行，必须内嵌于 Web 页中，然后在支持 Java 的浏览器上控制执行，或用 appletviewer 执行。它不允许访问本地系统资源，没有 Main()方法，但有多个其他入口点[如 init()、start()等]，以适应在浏览器上显示时可能发生的各种事件。

1.5 运行环境与开发工具

Java 语言可以在任何硬件环境下运行，如 IBM 的 PC、Sun 的 Sparc、苹果公司的 PowerPC 等，它的操作系统可为 Windows 3.x/9x/NT、UNIX 等。

Sun 的 Java 开发工具包 JDK（Java Developer's Kit）是开发 Java 语言的一个综合性工具集，有多个版本，包括 for Windows、for UNIX 等。随着 Java 的不断发展，JDK 的内容不断增加，每个版本也不断升级。可通过查询 Sun 的 Java Web 站点（http://www.javasoft.com/）来寻找 Java 的最新信息和 JDK 的最新版本。

JDK 包括如下主要开发工具。

（1）编译器（javac.exe）。将 Java 源代码（*. java）编译成可在 JVM 上执行的字节码类文件（*.class）。

命令格式：javac Option filename

Option：

-classPath <pathList>：用于设置路径表（多个路径以分号分隔），在该路径表中 Java 寻找需被调用的类。

-sourcePath <path>：用于指定被编译源文件的路径。

-d dir：指定经编译产生的类文件存放目录，缺省表示当前目录。

-g：使编译器为 Java 类文件生成调试表。有该选项时调试表包含局部变量和行号，缺省时只生成行号。

-nowarn：关闭编译时的错误提示。

-verbose：使在编译过程中显示附加信息，使用户清楚地了解源文件被编译的过程和每一步所需的时间。

（2）解释器（java.exe）：解释执行类文件。

命令格式：java Option classname

Option：

-debug：在调试模式下启动解释器。带此选项时，Java 提示输入口令，然后才进入调试阶段。

-checksource：先比较源文件和类文件中的数据，如源文件修改过，则自动重新编译类文件，然后解释执行。

-classpath path：指定类路径名，若指定多个路径，则用分号分隔。

-verify：检验字节代码，缺省时只检验类装入器装入系统中的代码。

-noverify：关闭所有代码检验。

（3）Applet 观察器（appletviewer.exe）：用于在简单的环境下测试 Applet 或解释执行 Java 的类文件。

一般情况下，要运行一个 Applet 需在 WWW 浏览器上进行，但在测试一个 Applet 时使用浏览器效率低、不方便。为此在开发制作过程中可用 Applet 观察器观察程序执行结果。Applet 观察器可在 WWW 以外的环境下运行 Applet。

命令格式：appletviewer Option filename

Option：

-debug：使在调试过程中启动 Applet 观察器。

其中 filename 是内嵌有 Applet 的 HTML 文件。

（4）调试器(Jdb.exe)：调试 Java 程序的命令行调试器。

（5）类文件反汇编器（javap.exe）：用于反汇编一个类文件。在只有类文件而没有源代码的情况下有用。也有一些选项用于指明反汇编的条件。

（6）文档生成器（javadoc.exe）：直接从 Java 源代码生成 API（Application Program Interface）文档。文档生成器通过对 Java 源文件进行分析，生成有声明内容的 HTML 页。

Java 源代码的编辑可在任何文字处理器下进行。另外 Jpad 是 Sun 公司的 Java 程序开发平台，不仅可以编辑源代码，也可编译、调试、运行。

在安装 JDK 时，如指定安装路径为 C:\Java\JDK，则开发 Java 程序的工具文件应在 C:\Java\JDK\bin 目录下。为了编译运行方便，可在 Autoexec.bat 的 PATH 行加上该路径。

Microsoft 公司 1996 年发行了 Java 交互开发环境 Visual J++，目前常用的版本是 Visual J++6.0，其功能强大，操作简便，是开发 Java 程序的编程平台，包括编辑器、编译器、调试器及大量的在线文件。

Borland 公司开发的 Jbuilder 也是一个可视化的 Java 程序开发环境，可自动生成基于 Windows 窗体的 Java 程序以及 Applet，集编辑、编译、调试、运行于一体。

1.6 Java 程序实例

本节举例说明 Java 两类程序的运行过程。

例 1-1 运行一个简单的 Application。在屏幕上输出“Hello World!”。

```
//Sample Hello World Application              // 注释行
//filename:HelloWorldApp.java
public class HelloWorldApp{                   // 定义类
public static void main(String args[]){       // 定义主方法
    System.out.println("Hello World!");      // 调用 System.out 对象的 println 方法
    }   // 结束方法体
}       // 结束类定义
```

Java 作为面向对象的编程语言，程序由类构成，所有的数据和处理数据的方法封装在类体中，通过生成类的实例构成应用程序的单元。

本例只有一个类，一般情况下，一个 Java 程序可有多个类，但至多有一个 Public 类；每个类中可定义多个方法，但所有类中只有一个主方法，Java 解释器在没有生成任何实例的情况下，以 main()为入口方法解释执行该程序。

编译该程序，在 DOS 提示符下将 Java 源程序文件存放的位置设置为当前目录，键入命令如图 1-5 所示。

图 1-5　编译 Java 文件

若代码有误，则提示错误信息，否则生成字节码文件：HelloWorldApp.class。

运行，在 DOS 提示符下键入如图 1-6 所示命令，在屏幕上显示的运行结果如图 1-6 所示。

图 1-6　执行字节码文件及运行结果

例 1-2　运行一个简单的 Applet。在浏览器的 Applet 的显示区中显示“Hello World!”。

```
//Sample Hello World Applet
//filename:HelloWorldApplet.java
import java.awt.Graphics;              // 引入 Graphics 类
public class HelloWorldApplet extends java.applet.Applet{
   public void init(){                 // 重写 Applet 的 init()方法
      resize(100,50);
   }
   public void paint(Graphics g) {  // 重写父类 Applet 的 paint()方法
      g.drawString("Hello World!",40,40);//画出字符串“Hello World!”
   }
}
```

可在任何文字处理器中建立该文件，编译生成字节码文件。不能由解释器直接运行，必须将 HelloWorldApplet.class 嵌入到一个 HTML 文件中，才能在支持 Java 的 WWW 浏览器上运行，或用 appletviewer 运行。下面是调用 Applet 的 HTML 文件。

```
<html>
<! filename:HWApplet.html>
<head>
<title>Hello World Applet</title>
</head>
<body>
<applet code="HelloWorldApplet.class" width=150 height=80>
</applet>
</body>
</html>
```

用浏览器打开该 HTML 文件时，HelloWorldApplet 便会自动运行，并在浏览器上显示字符串“Hello World!”。

下面简单总结 Java 源程序的构成。

一个 Java 源程序由一个或多个编译单元组成，每个编译单元只能由下列部件构成。

（1）一个包语句（如 Package java.Simple）。可选，选择时必须放在程序的最前面，此时，将程序中定义的类编译后放在 java.Simple 包中。

（2）多个引入语句（如 import java.awt.Graphics）。表示编译时从 java.awt 包中引入 Graphics