21世纪高职高专规划教材·计算机系列

Java语言程序设计

杨有安 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.com.cn

第 1 章　Java 语言概述

本章要点：

1. Java语言的产生、发展及特点
2. Java语言的运行环境
3. Java语言程序的开发过程

本章目的：

1. 了解Java语言的全貌
2. 了解Java语言的作用
3. 了解Java语言的开发步骤

随着 Internet 在全世界的迅速发展，Java 语言应运而生，这种适用于 Internet 的网络语言是信息领域的一场新的技术革命，对信息世界愈来愈产生巨大的影响。本章主要介绍 Java 语言的产生、发展及特点，并详细介绍 Java 语言的运行环境及程序开发过程。

1.1　计算机和 Internet

从第一台计算机到现在的微型计算机，计算机经历了电子管、晶体管、中小规模集成电路、超大规模集成电路四代产品。1946 年的 ENIAC 计算机做乘法的运算速度为 500 次/秒，而现在最普通的微型计算机做乘法的运算速度都已达到百万次/秒以上。近年来，随着高档微型机的飞速发展以及数据库管理技术的进一步完善，加之多媒体技术的应用，使得计算机技术的应用向智能化的方向又迈进了一步。特别是 Internet 的实现，表明计算机技术的应用又上新台阶。计算机技术的应用现已从各个科学研究部门扩展到社会，进入到家庭。从科学计算、实时控制等方面逐步扩展到非数值计算的各个领域，在计算机辅助设计和制造、企业管理、办公自动化、网络通信、电子商务等方面发挥了巨大的作用，取得了可喜的效益。

Internet 在当今计算机界乃至整个世界都是最热门的话题，以至有人将它称为“信息高速公路”。Internet 是“国际计算机分组交换网”的缩写，是连接全球计算机的网络，其原型为美国国防部在 20 世纪 60 年代末所开发的 ARPANET。目前很难对 Internet 进行严格的定义，从纯技术的角度看，可以认为 Internet 是一个相互衔接的“IP 网”，即可对成千上万的局域网、企业网、校园网及全球性的计算机网络进行实时互连。某网络系统一旦连接到 Internet 上，则表明其是 Internet 的一个 Web 结点，即已经与 Internet 连接。在 Internet 上，广泛分布、各自独立的计算机用户能够相互稳定可靠地传输信息和数据，从而能实现各自的应用。

Internet 采用 TCP/IP(Transport Control Protocol/Internet Protocol)协议，它是一组通信协议和应用协议的集合，其中 TCP 是传输控制协议，IP 是互联网络协议。Internet 上的数百万台机器都是基于 TCP/IP 协议交换信息。TCP/IP 是开发性的协议，具有很多优点，它有力地支持许多协议，如可以使一台计算机通过网络交换二进制和文本文件的文件传输协议（FTP，

File Transfer Protoco1)。它还支持 Telnet 协议，Telnet 协议可以使一台计算机远程访问另一台计算机，并且可以就像在本地计算机上运行程序一样在那台计算机上执行程序。它还支持 Simple Mail Transfer Protoco1(SMTP，简单邮件传输协议)的高级协议，这种协议允许用户与其他的计算机交换含有声音、图像以及文本的电子邮件。

Internet 不仅信息资源广泛，而且为用户提供了以图、声、文并茂方式包罗万象的各种信息服务的应用功能，如快速方便地交换信息(通过电子邮件、FTP)，访问资深人士和专家(通过 USENET)，接收日常更新的专题信息(Mailing List)以及实时地在线交谈(BBS)等。中国 Internet 的主要国际出口结点有:

* 中国科学院中关村地区教育与科研网络
  名称: IHEP 和 NCFC
* 国家教委中国教育和科研计算机网
  名称: CERNET
* 邮电部中国公用计算机互联网
  名称: ChinaNet
* 电子部国家公用经济信息网
  名称: ChinaGBN

浏览 Web 是 Internet 上主要的服务功能，Web 是 WWW（World Wide Web）的简称，或称环球信息网、万维网。1989 年由欧洲粒子物理实验室(CREN)的 Tim Rerners-Lee 提出其原理，目的是为世界各地的物理学家交换彼此想法提供信息服务，同期还提出 HTML(HyperText Markup Language——超文本标识语言)。浏览 Web 的工作模式是通过客户机/服务器（Client/Server）进行的。用通俗的话来说，使用 WWW 时，用户首先要向远端的计算机发出申请，使本地计算机和远端的一台计算机建立联系，随后系统按用户请求发送过来一个网页文件，用户就可在异地使用远端的信息资源。在这个过程中，发出服务申请的计算机称作客户机，提供服务的远端计算机称为服务器。WWW 客户端浏览器负责处理用户对 Web 文档的请求，按一定方式连通服务器信息源，用点击鼠标方式取回所需信息，以多媒体方式显示在浏览器的窗口中。浏览器的软件产品有 Netscape Navigator、NCSA Mosaic、Internet Explorer、Sun HotJava 等。WWW 服务器功能则是负责向浏览器提供所需服务，监听浏览器发出的请求，并向浏览器传送所需信息。

WWW 工作过程为:

① WWW 浏览器通过 TCP/IP 向服务器传送客户对多媒体信息（文本、图像、声音）的访问要求;

② WWW 服务器产生应答数据，回送应答数据并关闭链接，完成一次基于 HTTP 协议（超文本传输协议）的会话;

③ WWW 浏览器对应答数据进行解释，并显示给用户。

在万维网中用户可以向多个服务器发出申请，观看其主页，获得所需的信息。WWW 的基本协议是 HTTP(HyperText Transport Protoc ol)协议，它基于 TCP/IP 之上，属于 Web 浏览器和 Web 服务器之间的应用层通信协议。依据该协议用户可在 Internet 上传送以 HTML 编写的网页内容，从而进行信息交流。HTML 可以控制文档的字型、字号大小、颜色等，还可嵌入其他实体，如正文、图形、图像、声音等，也可通过超链接引用其他 WWW 网络资源。

1.2 计算机程序设计语言与 Java

计算机只识别二进制数。一组由 0 和 1 组成的数值代码构成某台计算机的一条指令，可控制计算机进行相应的动作，称做机器指令。例如：用 11 表示做加法运算，需将计算机内存 0010 单元中的数加到 0001 单元中。该功能的机器指令可写为 1100010010。机器指令是控制计算机进行特定动作的二进制码，是用户与计算机能沟通信息的特定语言。某台计算机机器指令的集合称为该台计算机的机器语言。机器语言十分烦琐，枯燥无味，直观性差，而且编程工作量大。为了提高用户编程的效率，人们将机器指令用特定符号来编写，即形成符号语言，又称为汇编语言。例如，用汇编语言(符号语言)实现上例加法运算的语句为：

```
ADD A,B
```

该语句表示将 CPU(中央处理单元)中 A 寄存器的数值与 B 寄存器中的数值相加，其结果存放在 A 寄存器中。汇编语言中的每一条语句与相应计算机的机器指令一一对应。在计算机问世初期，其运算速度很慢，且内存容量又太小，汇编语言发挥了很大的作用。值得注意的是，机器指令随计算机类型的不同而不同，各种类型的计算机都有各自特定的机器指令系统，互不兼容。为此，要求具有与所用计算机无关，并能实现结构化、模块化，更接近人们日常应用、思维习惯的语言，于是就产生了高级程序设计语言。上例中的汇编语言语句用 Java 语言可以写成：

```
A = A + B;        或者    A += B;
```

显而易见，高级语言与人们习惯用的语言和数学公式极为相似。高级语言具有以下特点：

1. 用户不必学习机器指令，也不必懂得计算机内部的详细结构和工作原理，而能方便地利用这类语言在计算机上处理数据信息。

2. 高级语言适用于各种计算机系统。为完成这一功能，每种高级语言都配备了相应的编译系统，该系统能将高级语言编写的程序翻译成计算机所能接受的机器指令程序。高级语言在计算机上执行的情况如图 1-1 所示。

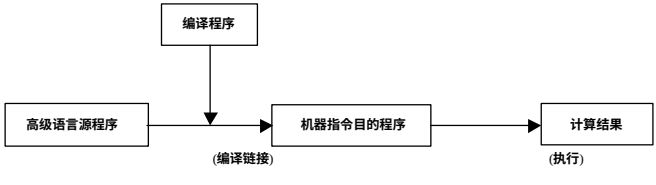

图 1-1　高级语言在计算机上执行的示例

迄今为止，世界上已有几千种高级语言。按其应用情况分，常用的高级语言大致可分为以下几类：

过程语言： FORTRAN、PL/1、COBOL、BASIC、Pascal、C

函数语言： Lisp、ML

逻辑语言： Prolog

面向对象语言：C++、Smalltalk-80、Ada、Java

高级语言解题的过程：

1. 根据实际问题构造数学模型，即归纳为数学公式。
2. 选择适当的计算方法，即将数学公式转换成适合计算机解题的方法。

例：求π值可用如下方法：

a. $\frac{22}{7}$

b. $\frac{\pi^2}{6}=\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}+\Lambda+\frac{1}{n^2}$

c. $\frac{\pi}{2}=\frac{2.2}{1.3}\times\frac{4.4}{2.5}\times\frac{6.6}{5.7}\times\Lambda$

3. 用高级语言编写程序(必要时设计程序语言框图)。
4. 上机调试程序，直到最后结果正确无误。

随着 Internet 以迅雷不及掩耳之势向全球的每个角落扩张，它将世界各地成千上万的计算机子网连成一个庞大的整体，而这些子网是由各种各样不同型号、不同规模、使用不同操作系统、具有不同应用软件平台的计算机构成的。但是，随之而来的问题是需要有一种能在各种不同的计算机环境上运行的语言，以便让 Internet 发挥更大作用。

1990 年 SUN 公司上马了一个叫 Green 的项目，要求实现一种与平台无关、可靠性强、小而灵活的编程语言，用来控制有线电视和小型家用电器，这就是 Java 项目组的前身。1994 年，当全球信息网的热潮席卷全球时，Java 这种中性平台的语言恰恰是全球信息网所期待的编程语言。随后，Java 在基于网络环境的基础上进行了一系列的改进，融合了 C 和 C++等传统程序语言的优点，形成了与众不同的面向实体的通用程序设计语言 Java。用 Java 开发出来的软件可以不用修改或重新编译就可直接用于任何计算机上，也不管它用什么操作系统或哪个版本，Java 程序都能安全运行。Internet 的迅猛发展和环球信息网的快速增长更促进了 Java 语言的研制，使之成为 Internet 上深受欢迎的开发与编程语言。1995 年 5 月 Java(JDK1.0)正式发布后，一些著名的计算机公司纷纷购买了 Java 语言的使用权，如 MicroSoft、IBM 等。1997 年 2 月 18 日 JavaSoft 公布了 JDK1.l 版，3 月 11 日又公布了 JDK1.1.l版，JDK1.1.1 去掉了 JDK1.1 中的一些错误。随后的 JDK1.1.2 以及 JDK1.1.3 再次修改了一些错误，不过并没有增加新的特性。现在使用的是 JDK1.2.2、JDK1.3 版本，其标准基本上没有改变。JavaBeans 规范以及新的 AWT 使得 JDK 将相对稳定。

Java 是一种编程语言，同时又是一种开发环境和一种应用环境。它是一个面向实体的通用程序设计语言，Java 不但适用于网上程序设计，也适用于一般的大规模软件工程项目。在 Java 之前，全球信息网上的网页基本上是静态的，一个网站不能执行另一个网站上的程序，因为这两个网站可能用的是不同的计算机、不同的操作系统，或者只是不同版本的软件。即便这些全一样，也还有一个很重要的安全性问题。Java 的出现迅速改变了这种状况。Java 的小应用程序(Applet)可以被下载到任何用户机器上安全运行，这使得用户与网页的互动对话、动画等成为现实。

Java 的诞生对整个计算机产业发生了深远的影响，对传统的计算模型提出了新的挑战，倾倒了无数用户。微软总裁比尔·盖茨通过一段时间的观察的后，不无感慨地说“ Java 是长时间以来最卓越的程序设计语言”，并确定微软整个软件开发的战略从 PC 单机时代向着以网

络为中心的计算机时代转移，且把购买 Java 作为他的重大战略决策。SUN MicroSystem 公司的总裁 Scott McNealy 认为：Java 为 Internet 和 WWW 开辟了一个崭新的时代。Dta 咨询公司的高级软件工程师 Rich Kadel 说："Java 不仅仅是一种程序设计语言，更是现代化软件再实现的基础；Java 还是未来新型 OS 的核心；将会出现 Java 芯片；将构成各种应用软件的开发平台与实现环境，是人们必不可少的开发工具，……"。

Java 为全球信息网带来了生命和一场真正意义上的技术革命，必将对未来产生如下的影响：

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，Java 是最纯、最优秀的面向对象的语言；

2．计算过程的可视化、可操作化的软件的开发；

3．可视化动态画面的设计，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)及各种功能模块设计；

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)和发展提供了强大的支持；

7．在各种管理系统中与各类数据库连接查询的 SQL 语句实现给人以一种新的景象；

8．为分布式对象技术提供了良好的环境；

9．为开放系统在各个计算机领域的运用和发展开辟了道路；

Java 是一个广泛使用的网络编程语言，它是一种新的计算概念。首先，作为一种程序设计语言，它简单、面向对象、不依赖于机器的结构，具有可移植性、健壮性、安全性，并且提供了并发的机制、具有很高的性能。其次，它最大限度地利用了网络，Java 的小应用程序(Applet)可在网络上传输而不受 CPU 和环境的限制。另外，Java 还提供了丰富的类库，使程序设计者可以很方便地建立自己的系统。归纳起来 Java 有以下几个特点：

1．通用性强

Java 源程序可在任何一台安装了 Java 虚拟机的机器上运行，这一点对网络应用至关重要。Java 解释器生成与体系结构无关的字节码指令，只要机器上安装了 Java 运行系统，Java 程序就可以运行。这些字节码指令对应于 Java 虚拟机中的代码，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

2．解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息，使连接过程更加简单。

3．简单性

Java 语言通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 来源于 C++，但删除了其中的多继承、指针操作、GOTO 语句、全局变量及头文件、宏语句、struct 和 union 模块、模板、自动类型转换、以及与 C 的(向后)兼容性等功能。Java 略去了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作，因此不必做释放内存空间的工作。

4．面向对象

Java 语言是一个面向对象的高级语言，除定义整数、浮点数、布尔值、字符的基本类型

不是类，属非面向对象的部分外，其他均是类。Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

5. 健壮性

Java 是一个强类型的语言，在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时容易产生的错误。它通过集成的面向对象的例外处理机制，在编译时提示出可能出现但未被处理的例外，帮助程序员正确地进行选择以防止系统的崩溃。另外，Java 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

6. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针操作，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊木马”等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

7. 支持多线程

多线程机制使应用程序能够并行执行，即允许多个应用程序并行操作，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易地实现网络上的实时交互行为。

8. 动态性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量，且不会影响用户程序的执行。Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式和扩展性。

9. 分布性

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

10. 可移植性

Java 与平台无关，程序可以方便地被移植到网络上的不同机器，字节码可直接被执行。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语言实现的，Java 运行时系统由标准 C 实现，这使得 Java 系统本身也具有可移植性。

11. 高性能

和其他解释执行的语言如 BASIC、C 不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。能编写多线程的应用程序。

1.3 Java 语言的开发及环境

Java 程序的工作原理与其他高级语言不同，Java 语言是解释执行的。用户写好 Java 源程序后，通过 Java 编译器(Javac)将源程序编译成字节码，然后在任何装有 Java 虚拟机的平台上运行。Java 虚拟机运行时，系统对字节码进行解释执行。字节码文件内容与具体计算机机型无关，无论什么机型，只要安装了 Java 虚拟机都可执行。

Java 开发工具 JDK 是许多 Java 专家最初使用的开发环境。尽管许多编程人员已经使用了另外的开发环境，但 JDK 仍被当做 Java 开发的重要工具。JDK 开发工具可以在网上找到，SUN 公司的网站 http://Java.sun.com/提供了开发工具、Hotjava 浏览器以及关于 Java 语言的各个方面的说明。还可从国内的一些 ftp 站点中获得，这样要比从国外站点上下载快一点，如：在 ftp.pku.edu.cn 站点的 pub\Javasoft\pub 目录中下载。ftp.pku.edu.cn 站点的内容相当全面，它提供了 JDK 以及其他各种 Java 的相关产品，且提供下载。如下载 JDK1.1.3 可以用浏览器直接访问下面的网页

http://java.sun.com/products/jdk/1.1/index.html，

或者访问具有相似内容的网页

http://www.javasoft.com/products/jdk/1.1/index.html。

Java 开发环境有多种不同的版本，用户可以根据具体的机器情况下载相应版本的 JDK。Windows95 或 Windows NT 的用户可以下载 Win95/NT 版的，UNIX 的用户可以下载 UNIX 版的。

对于 Win95/NT 的用户，下载后的 JDK 文件可为 jdk113.exe，对于 UNIX(Solaris 系统)用户，下载后的 JDK 文件可为 dk1.1.3_solaris2_sparc.bin。Java 开发工具下载后还要进行安装才能使用。

对于 Win95/NT 的用户，若将 Java 开发环境安装在 C 盘的根目录下，可在 Windows 资源管理器中直接运行 jdk113.exe 文件或者在 MSDOS 窗口中执行下面命令行：

```
jdkll3.exe
```

这是个自解压文件，执行后，在 C 盘的根目录下会生成一个名叫 jdk1.1.3 的目录，然后在 AUTOEXEC.BAT 文件中设置路径。AUTOEXEC.BAT 文件中的路径可设置如下：

```
PATH=C:\windowS;C:\windowS\command;C:\dos;C:\jdk1.1.3\bin；
```

也可将 JDK 安装在其他逻辑盘上，对此，必需修改相应的路径。

另外还要设置环境变量：

```
set CLASSPATH=.;C:\jdk1.1.3\lib\classes.zip
```

若使用缺省值这一步可以跳过。环境变量告诉 Java 虚拟机以及 jdk1.1.3\bin 目录下的应用程序到哪里去找类库。比如寻找 lib 目录下的 classes.zip 文件，如果你改变了 classes.zip 文件的位置，或者你想引入自己开发的类库，那么你就必须设置环境变量。设置工作完成后，应重新启动计算机，以使 JDK1.1.3 生效。

值得注意的是：不能将压缩文件 classes.zip 解压，否则系统无法工作。

对于 UNIX(Solaris 系统)用户，可将下载后的 JDK 文件安装在/usr/local/目录下。在 shell 窗口中执行下面的命令行：

```
%chmod a+ x jdk1.1.3_solaris2_sparc.bin
%./jdk1.l.3_solaris2_sparc.bin
```

执行后会生成一个名叫 jdk1.1.3 的目录。然后设置路径，在 C shell（csh 用户）中修改.cshrc 文件：

```
set Path=($ Path /usr/local/jdk1.1.3/bin)
```

必要时可设置环境变量：

```
%setenv CLASSPATH .: /usr/local/jdk1.1.3/lib/classes.zip
```

在 JDK 开发环境的工具库中提供了 7 个主要的实用程序：

Javac	Java 编译器，将 Java 源代码转换成字节码。
Java	Java 解释器，直接从类文件执行 Java 应用程序字节代码。
appletviewer	小程序浏览器，一种执行 HTML 文件上的 Java 小程序的 Java 浏览器。
Javadoc	根据 Java 源码及说明语句生成 HTML 文档。
Jdb	Java 调试器，可以逐行执行程序，设置断点和检查变量。
Javah	产生可以调用 Java 程序的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件。
Javap	Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义。

当 Java 支持环境准备完毕后，就可建立和运行 Java 程序。Java 程序有两种：独立执行的应用程序（Application program）和小应用程序（Applet）。独立执行的应用程序是一个独立存在与执行的程序，小应用程序则是一个只能在网络浏览器的环境上运行的程序。下面将分别介绍这两种程序的开发过程。

一、独立执行的应用程序(Application program)开发过程

开发简单的 Java 独立执行的应用程序，首先要选一个用户较熟悉的编辑器，如 UNIX 系统中的 VI 或者 DOS 的 EDIT 文本编辑器，然后编辑独立执行的应用程序内容。

例如，可用文本编辑器(如 Edit)编写 Java 源程序如下内容:

```
public class HelloWorld{                          //定义一个 HelloWorld 的类
    public static void main(String args[ ]){       //定义一个 main( )方法
        System.out.println("Hello World!");        //输出字符串"Hello World!"
    }
}
```

程序内容中的保留字 class 是关键字，它定义了一个 HelloWorld 类，关键字 public 说明它是公共类(public)，每个程序只有一个公共类。整个类用大括号“{}”括起来。在该类中定义了一个 main( )主方法，程序运行时首先执行 main( )方法，该方法必须有，而且只能有一个。定义时必须按照如上的格式，其负责调用其他函数或过程。main( )方法定义时使用关键字 public，表示其访问权限，指明所有类都可以使用该方法。其中关键字 void 的使用指明该 main( )方法不返回任何值。关键字 static 指明该方法是个类方法，可通过类名直接调用。String args[ ]是传递给 main( )方法的参数，参数名叫 args，这是个标准参数，必须有，否则语法错。args 是类 String 的一个实例，参数可以为 0 个或多个，每个参数用“类名参数名”来指定，多个参数间用逗号分隔。整个方法的内容也用大括号“{}”括起来。该 main( )方法中只有一条语句：System.out.println("Hello World!")，是程序体，其功能是用来输出“Hello World!”字符串。Jave 程序中可定义多个类，每个类中可定义多个方法。程序中各关键字的详细功能将陆续地在以后的章节中加以解释。

程序中双斜杠的部分是 Java 提供的一种注释格式，用来注释程序。按软件工程的要求，程序都应该有注释，以便让别人看得懂，也便于自己查阅。Java 中的注释格式有三种：

第一种以“/*”开始，以“*/”结束，在“/*”和“*/”之间写入要注释的内容，即/*[<注释内容>]*/。注释的内容可以有多行组成。例如：

```
/* 这是一个独立执行的应用程序
   该应用程序中只定义了一个类。*/
```

第二种是以双斜杠开始，在双斜杠的后面加上注释内容，即//[<注释内容>]。这种格式只能用于单行说明，不能换行。例如：

```
System.out.println("Hello World!");//在屏幕上输出“Hello World!”字符串
```

第三种格式是以“/**”开始，以“*/”结束，中间为注释说明，即/**[<注释内容>]*/。这是一种 Java 特有的 doc 注释方式，主要是为支持 JDK 工具 Javadoc 而采用的，用以生成 doc 格式的文档说明文件。一般用得比较少。

键入程序内容后，应检查输入是否正确，然后就可考虑存盘。存盘时选用文件名最好和类的名字相同，因为类名是用 public 关键字修饰的。Java 解释器要求公共类必须放在与其同名的文件中，因此，文件名必须与类名相同。Java 源文件名后缀名(扩展名)只能为.java，该例文件名可为 HelloWorld.java。请务必注意字母的大小写，Java 文件名中字母大小写是有区别的。

HelloWorld.java 是 ASCII 码源文件，还应对其编译，生成字节码文件。编译时要用到编译器 Javac，命令格式如下：

```
javac HelloWorld.java
```

编译成功的结果是在同一个目录中生成 HelloWorld.class 字节码文件，该文件带有许多编译时产生的信息。若源程序存盘时使用了与类名不同的名字，则形成的字节码文件名为类名加上后缀名.class。若源程序有错，编译器会提示第几行可能有哪种错误。用户应重新进入编辑器，按显示的信息要求修改源程序，再编译，直至正确无误。

最后用 java 解释器来解释执行该字节码文件，格式如下：

```
java HelloWorld
```

执行结果，在屏幕上显示：

```
Hello World!
```

请务必注意，在这个 java 解释功能的命令行中，文件名标识中不能带.class 扩展名。还要注意的是 Java 编译器(Javac)和 Java 解释器(Java)不是一回事。编译器（Javac）是用来编译 Java 源文件生成字节码的，而解释器（Java）则是用来执行字节码文件输出结果的。

二、小应用程序(Applet)开发过程

首先要选一个较熟悉的编辑器，用来编辑小应用程序的源代码。例如，可用文本编辑器(如 Edit)编制 Java 小应用程序的源代码内容如下：

```
import java.applet.*;
import java.awt.Graphics;
public class HelloWorldApplet extends java.applet.Applet{
    {public void paint(Graphics g){
        g.drawString ("Hello World!",20,50);
    }
}
```

该小应用程序中的 import 语句引入 JDK 的 java.applet.*和 java.awt.graphics 下所有的包，使得该程序可能使用这些包中所定义的类。然后声明一个公共类 HelloWorldApplet，用 extends 指明它是 Applet 的子类。在类中，重写父类 Applet 的 paint( )方法，其中参数 g 为 Graphics 类，表明当前作画的内容。在 paint( )方法中，调用 g 的方法 drawString( )，在坐标(20,50)处输出字符串“Hello World!”，其中坐标参数是用像素点表示的。在这个程序中没有用到 main( )

方法，这是小应用程序 Applet 与独立执行的应用程序 Application 的区别之一。

对该例 Java 小应用程序的源代码检查无误后，就以文件名 HelloWorldApplet.java 存入磁盘，然后按下面格式对其进行编译。

```
javac HelloWorldApplet.java
```

编译成功后，在同一个目录中生成字节码文件 HelloWorldApplet.class。接下来的操作应是解释执行字节码文件。由于小应用程序 Applet 中没有 main( )方法作为 Java 解释器的入口，故无法用解释器 Java.exe 解释执行。

例如: java HelloWorldApplet

屏幕显示:

```
In class HelloWorldApplet:void main(string argv[ ]) is undefined
```

解释执行的结果告诉用户没有定义 main( )方法，无法执行。解释器将该程序当做独立执行的应用程序来操作，main( )方法是独立执行的应用程序的入口，没有 main( )方法，独立执行的应用程序就无法执行。小应用程序 Applet 没有 main( )方法，它必须嵌入 HTML 文件，用支持它的浏览器去运行它。JDK 的用户还可用 HotJava 或用相应版本的 Java 开发工具包自带的 appletviewer 运行工具。由于建立小应用程序 Applet 所需要的这一特殊的规则，所以建立一个小应用程序 Applet 要比建立一个独立执行的应用程序稍微复杂一点。

为了能运行该小应用程序，应事先编写一个 HTML 文件，把该 Applet 嵌入其中。该例相应的 HTML 文件内容可按如下编写:

```
<HTML>
<HEAD>
<TITLE> My first applet: </TITLE>
</HEAD>
<BODY>
<APPLET CODE="HelloWorldApplet.class" WIDTH=200 HEIGHT=40>
</APPLET>
</BODY>
</HTML>
```

其中，用<APPLET>标记来启动 HelloWorldApplet.code 指明字节码文件所在位置；WIDTH 和 HEIGHT 的参数指明 applet 输出所占范围大小。把这个文件用 HelloWorldApplet.html 名字存盘，文件名中 HelloWorldApplet 字符串用户可自定，但扩展名一定是.html。然后就可用 appleviewer 或浏览器来解释执行该小应用程序 Applet。用 appleviewer 可按如下格式运行:

```
appleviewer HelloWorldApplet.html
```

这时屏幕上弹出一个窗口，其中显示信息:

```
Hello World!
```

从上述例子中可以看出，Java 程序是由类构成的，对于一个独立执行的应用程序来说，必须有一个类中定义 main( )方法；而对 applet 来说，它必须作为 Applet 的一个子类。在类的定义中，一般应包含类变量的声明和类中方法的实现。在以后的各章中，会详细介绍。本节只是使大家对 Java 程序有一个初步的了解。

Java 语言的特性使它可以最大限度地利用网络。小应用程序适应于互联网上运行，它是动态、安全、跨平台的网络应用程序，通过主页发布到 Internet，网络用户访问服务器的 Applet

时，这些 Applet 从网络上进行传输，然后在支持 Java 的浏览器中运行。小应用程序使用户的网页(Homepage)更加丰富，网页不再只是静态的，而可以是交互式的，还可以加上逼真的动画甚至美妙动听的音乐。小应用程序无 main( )方法的引导，必须嵌入在 HTML 文件中使用，用浏览器或 appletviewer 运行工具运行。由于 Java 语言的安全机制，用户一旦载入 Applet 就可以放心地来生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。虽然 Applet 可以像图像、声音、动画等一样从网络上下载，但它并不同于这些多媒体的文件格式，它可以接收用户的输入，动态地进行改变，而不仅仅是动画地显示声音的播放。

值得注意的是，用浏览器运行小应用程序要下载到用户系统上，为安全起见应做到以下几点:

① 小应用程序不能对用户系统进行写操作;
② 不能在用户系统上运行任何程序;
③ 不能用其他服务器通信(存放小应用的服务器除外);
④ 不能将服务器上的程序传送到用户系统上;
⑤ 解释执行时对字节码作安全检查。

独立执行的应用程序是用 Java 语言编写的一种普通程序，不需浏览器支持运行，起先由编译程序编译，再由解释器解释执行。程序必须有且只有一个 main( )方法作为程序的入口，程序执行时先执行 main( )方法。HotJava 浏览器就是用 Java 语言编写的。

不编程是学不会任何计算机语言的。子曰:“学而时习之，不亦乐乎？”你必须输入、修改、编译、运行每一个范例程序，多做习题。每运行一个程序，都可能给你一些新的启发。最重要的是，这样能够帮助你理解和记忆。这是学习计算机语言必由的“捷径”。

1.4 HTML 超文本标识语言

HTML 是一种标识语言,可以用以生成文本档案，Web 页面利用 HTML 来组织文档和链接文档。HTML 定义了文档(包括文本、图像、动画和声音)的结构、标识及属性，也可定义文档结构、字体、字型、版面设计、链接等。以下是一些 HTML 标记及功能:

<HTML>..≤HTML>	标记整个 HTML 文件，标记之间是文件内容
<HEAD>..≤/HEAD>	标记文件头部信息，标记之间是文件的总标题等信息
<BODY>..≤/BODY>	标记正文的内容，即这对标记之间是文件的正文
<HR>	产生一个分格线
 	在当前位置上换行
<H1>..≤/H1>	标记之间为第一层标题，字号较大
<H2>..≤/H2>	标记之间为第二层标题，字号比上一级小
<H3>..≤/H3>	标记之间为第三层标题，字号比上一级小
<H4>..≤/H4>	标记之间为第四层标题，字号比上一级小
<H5>..≤/H5>	标记之间为第五层标题，字号比上一级小
<H6>..≤/H6>	标记之间为第六层标题，字号比上一级小
<I>..≤/I>	标记之间内容为斜体字
<B>..≤/B>	标记之间内容为粗体字
<U>..≤/U>	文字下面画线

<P>	段落分界标记
<AHREF=http://.>.</A>	链接 URL
<AHREF="mailto.">.</A>	链接电子邮件
<AHREF="fileName">.</A>	链接另一个文件
<IMG SRC="xxx.GIF">./IMG>	显示文件名为 xxx.GIF 的图片

1.5 Java 虚拟机

Java 是在一个虚拟的环境下工作和运行的。从底层看，Java 虚拟机就是以 Java 字节码为指令组的软 CPU。其工作过程为：首先，将源程序经过编译后形成不依赖特定平台的字节码指令集，该文件扩展名为.class，存放在 Web 服务器上。这样，Java 程序就已作为 Internet 或 Intranet 资源存放在 Web 服务器上，随时可让客户使用。在客户端，用户使用 WWW 浏览器，通过 Internet/Intranet 将 Web 服务器上含有 Java 程序的主页下载，再依赖本地 Java 虚拟机对.class 文件解释执行。解释器在运行中安排内存布局，确定相应的地址后，运行产生结果。这样，内容丰富的 Java 应用资源便由服务器传送到客户端，并在用户浏览器上显示。

Java 虚拟机体系结构包括以下部分：数据类型、指令组、寄存器组、方法区、无用单元回收堆。Java 虚拟机规定了统一的数据类型以及类型长度。近 250 个字节码指令，每个完成一个基本运算。Java 虚拟机的寄存器用于保存机器运行状态，同微处理器的寄存器很类似。Java 虚拟机寄存器有四种：程序计数器(PC)；操作数栈端指针 Optop；当今执行方法的执行环境指针 Frame；指向当今执行方法的局部变量指针 Var.Java，其不定义或使用寄存器来传递或接受参数，保证指令集的简捷性和实现时的高效性。Java 虚拟机是栈式的，包括局部变量区、执行环境区、操作数区。Java 虚拟机的方法区是编译后的代码区域，它包括语法代码、符号表等。无用单元回收堆，作为动态分配对象的存储区。由于在当前主机操作系统上加上了 Java 虚拟机层，Java 字节码执行速度目前要比本地机器慢 10 到 20 倍。究其原因有如下几条:

首先，验证过程要花费时间，读入的类要在运行时验证，而传统程序在程序编译时即完成验证工作；其次，Java 指令都是字节码，由于大多数操作对象超过一个字节长，因此必须读多个字节码来取得操作符和不同操作数；再次，由于 Java 完全采用堆栈机理，运算操作都在堆栈上执行，而传统编译器在编译时进行多种优化工作，很多计算操作可直接在寄存器中完成，大大提高程序执行速度；最后，在程序执行期间，系统要进行无用内存单元回收工作，在回收过程中，程序将停止执行，这无疑也会影响性能。

习题

1. Java 语言对未来有何影响?
2. Java 语言有什么特点?
3. 请简述 Java 程序的运行过程。
4. 编写一个 Java 小应用程序，要求在屏幕上输出如下信息：

```
**********************************
            Hello World!
**********************************
```

5. 编写一个 Java 小应用程序，要求在屏幕上输出信息“Welcome to Java World.”。
6. 编写一个 HTML 文件，将第 5 题生成的 Applet 字节码嵌入其中，并用 WWW 浏览器观看这个 HTML 文件规定的 Web 页面。
7. 应用程序与小应用程序有什么区别?
8. 请简述 Java 虚拟机的体系结构。

第 2 章　数据类型与表达式

本章要点:

1. 语句、关键字和标识符的概念
2. 常量、变量、变量的初始化与值的输出等概念
3. 变量值的范围及精度
4. 算术运算符及算术表达式，赋值运算符和复合赋值操作符
5. 关系运算符及关系表达式
6. 逻辑运算符及逻辑表达式
7. 条件运算符和条件表达式
8. 运算符的优先级和结合规则
9. 表达式中数据类型的转换

本章目的:

合理地书写各类表达式。

2.1　语句

一个计算机程序由若干条语句组成，每条语句都能实现相应的功能：有的语句要求计算机完成特定的指令功能，有的则给计算机程序运行提供相关的信息。让计算机完成某个特定操作的语句称之为可执行语句，该语句经编译后产生二进制代码。例如，下面对两个变量 A 和 B 做加法的操作语句就属于可执行语句。

```
A=A+B;
```

或

```
A+=B;
```

还有许多种可执行语句，在后续章节中将陆续介绍。

还有一类语句，其在计算机程序运行之前为程序提供相关信息，该类语句称之为非执行语句。该语句仅为编译系统提供相关信息，编译后不产生机器指令。例如:

声明i变量为整型量的语句：　　int i;

声明数组a的语句：　　type [ ] a; 或 type a[ ];

以上两条语句也称之为声明语句。声明语句还有许多种，在后续章节中将陆续介绍。

不难看出，Java语言和C语言相似，程序的语句均以“;”号结尾。例题语句中的A、B称之为变量，供保存数据用，A=A+B称之为表达式。Java语言程序的变量及表达式都必须具有各自的数据类型(type)。如果计算机不知道A和B里存的是整数还是浮点数的话，就无法把A和B加起来，因为整数加法和浮点数加法的操作过程在计算机中是不同的。变量、表达式、语句常量、标识符、原始类型等均属于Java语言里的基本组成单元。

2.2 关键字和标识符

关键字是具有某种特定的含义的一批专用词汇，或称为保留字，只能用于特定位置。例如: CLASS、PUBLIC、IF。

Java中所有关键字的情况如下。

(1) 与C兼容的关键字21个

break	case	char	continue	default	do	double
else	float	for	goto	if	int	long
return	short	static	switch	void	volatile	while

(2) 与C++兼容的关键字11个

catch	class	const	new	operator	
private	protected	public	this	throw	try

(3) Java新增的关键字27个

abstract	boolean	byte	byvalue	cast	extends
false	final	finally	future	generic	implements
import	instanceof	inner	interface	native	null
outer	package	rest	super	synchronized	throws
transient	true	var			

标识符(identifier)是用来标识常量、变量、数据类型、类、方法等名字的字符串。Java 中的关键字就是典型的标识符。用户也经常定义和使用标识符，例如在本章前面的例子里，变量的名字 A 和 B 都是标识符；在第 1 章的 Java 应用程序中，类名 HelloWorld 也是一个标识符，变量名字 args 也是一个标识符。Java 规定标识符必须以字母、下划线“_”、美元符号“$”开头，之后可以任意跟数字和字母，不限长度。Java 的字母包括英文字母，也可包括中文字符。值得注意的是：Java 中的保留运算符 +、-、*、/、%、# 等是不能作为标识符的。例如#max、%X、A+B 都是不合法的。另外，Java 程序中的字母大小写在标识符中是有区别的，例如 IF、if、If、iF 为不同标识符。用户定义标识符时不能与 Java 关键字同名，标识符的命名应该有利于程序的可读性，这都是用户应该遵守的。请看下面几个正确及错误标识符的例子。

x、X、HelloWorld、Cat、dogs5、pai、$10 都是正确的标识符，其中 x 和 X 是不同的标识符。

\u03c0 是 π 的 Unicode 字符，因此可以用做标识符。

5dogs、%10、A+5、for 都是不正确的标识符，第一个不是以字母、美元符号、下划线开头，中间两个含有运算符，而 for 则是关键字。

为了使程序清晰易懂，给变量起名时应尽量做到见名知义。对于变量、类以及方法等的命名，世界范围内的 Java 程序员都采用“Smalltalk”的命名规则：

(1) 每个名字可以由几个单词连接而成。

(2) 对于类名，每个单词的开头字母应该大写。如：MyClass，GroupEmployee。

(3) 对于方法名和变量名，类似类名的命名规则。但第一个字母不用大写。如：

方法名 myMethod( ) 变量名 myVariable

(4) 表示私有的或局部的变量的标识符全部用小写字母。例: next_value

(5) 表示常量值的标识符，字母应该全部大写。如：LIMIT，RED。

(6) 包名应该全部小写。如：mypackage。

(7) 标识符中不用或少用美元符号，其在链接代码时用于链接库例程。

2.3 基本数据类型

类型(type)是程序设计语言里的基本概念。它决定了数据值的操作行为和表示行为，是一种类属标志。定义一个数据值的类型，就是要定义该数据值的取值范围和在该值上的操作这两部分。例如，整型定义一个整数的类型，它的值域是：…，-2，-1，0，1，2，…，在其上的操作就是常用的加减乘除等。在一个纯面向实体的语言中，类型这一概念被类取代了。类是一个比类型更广义的概念，除了定义取值范围和在数值上的操作这两部分之外，还允许有扩展、继承、修改等。这样的语言(如 Smalltalk)使用起来很不方便，定义与使用比较复杂。若把所有的类型都当成类时，即便使用一个整数 1，也要颇费周折地生成一个实体 1。为了避免这种情况，Java 采取了折中的办法，引入整数、浮点数、字符及布尔值四种基本数据类型，不放入类之中，称之为基本数据类型(primitive type)。另外还包括数组类型、类、接口三种。下面将一一介绍基本数据类型的内容。

基本数据类型分为整数类型(byte、short、int、long 的正负整数)、字符型类型(char 表示字符或正负整数)、浮点型类型(float 和 double 表示带小数点的正负数)、布尔型类型(boolean 表示逻辑值)。

整数型类型

根据值域的大小，Java有四种有关整数的基本类型。

基本类型	所占位数	取值范围	操作符
byte	8位	2^{-7} ~$-2^{7}-1$	+、-等
short	16位	2^{-15}~$-2^{15}-1$	+、-等
int	32位	2^{-31}~$-2^{31}-1$	+、-等
long	64位	2^{-63}~$-2^{63}-1$	+、-等

其中，byte类型特别适用于网络或文件中的字节流；short类型是一种高位在前的16位整型数据，而IBM-PC是低位在前的16位整数，故不常用；int类型是常用的，当该类型表示的范围不能满足需要时就用long类型。

字符型类型

基本类型	所占位数	取值范围	操作符
char	16位	所有Unicode字符	==、++、--等

因为Java的Unicode字符集包括了汉字、希腊、罗马、阿拉伯、数学符号等几乎所有语言的字符，故用16比特位表示一个字符。操作符==是比较相等，操作符++是找到某字符在Unicode代码集中的下一个字符。字符型数可以当做整型数进行操作，它既可以字符格式输出显示字符本身，也可以整型形式输出显示Unicode码值。

浮点型类型

浮点型是用来近似表示数学上的实数，其基本类型如下:

基本类型	所占位数	取值范围	操作等
float(单精度)	32位	3.4e-38~3.4e+38	+、-等