

计算机、程序和Java概述

学习目标

- 回顾计算机的基本组成、程序和操作系统 (1.2~1.4节)。
- 探究Java与万维网 (World Wide Web) 之间的关系 (1.5节)。
- 区分术语API、IDE和JDK (1.6节)。
- 编写一个简单的Java程序 (1.7节)。
- 在控制台上显示输出 (1.7节)。
- 解释Java程序的基本语法 (1.7节)。
- 创建、编译和运行Java程序 (1.8节)。
- (GUI) 使用JOptionPane输出对话框显示输出结果 (1.9节)。

1.1 引言

人们使用字处理程序编写文档,使用Web浏览器在互联网中探索,使用电子邮件程序发送电子邮件。这些字处理程序、Web浏览器以及电子邮件程序都是在计算机上运行的软件实例。软件是用程序设计语言开发出来的。程序设计语言有很多种——那么为什么要选择Java这种程序设计语言呢?答案就是Java能够让用户开发和部署可用于Internet上的服务器、台式电脑和小型手持设备的应用程序。互联网将会对计算技术的未来产生深远的影响,而Java肯定会在未来计算技术中占很大的比重。所以说,Java是一种真正的互联网程序设计语言。

我们即将开始一段激动人心的旅行,学习一门功能强大的程序设计语言。在开始这一奇幻之旅之前,很有必要回顾一下计算机的基本组成、程序和操作系统,熟悉一下数字系统。如果你已经很熟悉CPU、内存、磁盘、操作系统以及程序设计语言等术语,那么你可以跳过1.2~1.4节中对这些内容的回顾。

1.2 什么是计算机

计算机是存储和处理数据的电子设备。它包括硬件(hardware)和软件(software)两部分。一般来说,硬件包括计算机中可以看得见的物理部分,而软件提供看不见的指令,这些指令控制硬件并且要求硬件完成特定的任务。编写指令让计算机来完成的过程就称为计算机程序设计。要学习一门程序设计语言,并不一定要了解计算机硬件知识,但是如果你了解一些硬件知识的话,它的确可以帮助你更好地理解程序中指令的功效。本节介绍计算机硬件组件及其功能。

一台计算机是由以下几个主要的硬件组件构成的(如图1-1所示):

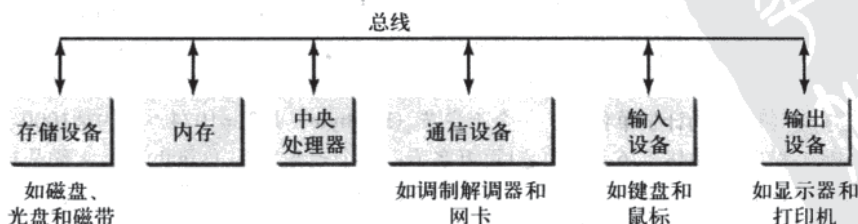


图1-1 计算机由中央处理器、内存、存储设备、输入设备、输出设备和通信设备组成

2 · 第 1 章 计算机、程序和Java概述

- 中央处理器 (CPU)
- 内存 (主存)
- 存储设备 (例如, 磁盘、光盘、磁带)
- 输入和输出设备 (例如, 显示器、键盘、鼠标、打印机)
- 通信设备 (例如, 调制解调器和网络接口卡 (NIC, 简称网卡))

这些组件通过一个称为总线 (bus) 的子系统连接, 总线负责在这些组件之间传输数据和电信号。

1.2.1 中央处理器

中央处理器 (Central Processing Unit, CPU) 是计算机的大脑。它从内存中获取指令然后执行这些指令。CPU通常由两部分组成: 控制单元 (control unit) 和算术/逻辑单元 (arithmetic/logic unit)。控制单元用于控制和协调除CPU之外其他组件的动作。算术/逻辑单元用于完成数值运算 (加法、减法、乘法、除法) 和逻辑运算 (比较)。

现在的CPU都是内嵌在一块小小的硅半导体芯片里, 这块芯片上有数百万个晶体管。每台计算机都有一个内部时钟, 该时钟以固定速度发射电子脉冲。这些脉冲用于控制和同步各种操作的步调。时钟速度越快, 它在给定时间段内执行的指令就越多。时钟速度的计量单位是赫兹 (hertz, Hz), 1赫兹相当于每秒1个脉冲。计算机的时钟速度通常是以兆赫 (MHz) 来表示的 (1MHz就是100万Hz)。CPU的速度在不断地提高。Intel公司的奔腾3处理器的运行速度是500MHz左右, 而奔腾4处理器的运行速度大约是3GHz (1GHz是1000MHz)。

1.2.2 内存

为了存储和处理信息, 计算机利用电的两种状态: 关 (off) 和开 (on), 习惯上认为它们分别是0和1。这些0和1被解释为二进制数字系统中的数, 并且将它们称为比特 (bit, 二进制数)。各种类型的数据, 例如, 数字、字符和字符串, 都被编码为比特序列。CPU要执行的数据和程序指令都以一组比特或字节的形式存储在计算机内存 (memory) 中, 每个字节由8比特构成。内存单元是由字节 (byte) 构成的有序序列, 如图1-2所示。

程序员不需要关心数据的编码和解码, 这些都是系统根据编码表来自动完成的。例如, 在流行的ASCII编码表中, 字符 'J' 是用一个字节01001010来表示的。

字节是最小的存储单元。像3这样的小数字就可以存储在单个字节中。为了存储一个单个字节放不下的大数字, 计算机需要使用几个相邻的字节。任何两个数据都不能共享或分割同一个字节。

内存中字节的内容永远非空, 但是它的原始内容可能对于你的程序来说是毫无意义的。一旦新的信息被放入内存字节, 该字节的当前内容就会丢失。

程序及其所需数据必须在它们被执行前放入内存。

每个字节都有一个唯一的地址。使用这个地址确定字节的位置, 以便于存储和获取数据。因为可以按任意顺序存取字节, 所以内存也被称为随机访问存储器 (Random-Access Memory, RAM)。现在的个人电脑通常至少有1GB的RAM。计算机的存储大小是以字节、千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 和万亿字节 (TB) 为计量单位的。一个千字节 (kilobyte) 是 $2^{10}=1024$, 大约是1000字节; 一个兆字节 (megabyte) 是 $2^{20}=1\,048\,576$, 大约是1百万字节; 一个千兆字节 (gigabyte) 大约是10亿字节; 而一个万亿字节 (terabyte) 大约是1000千兆字节。同CPU一样, 内存也是内置在一个表面上嵌有成千晶体管的硅半导体芯片上。与CPU芯片相比, 内存芯片更简单、更低速, 也更便宜。

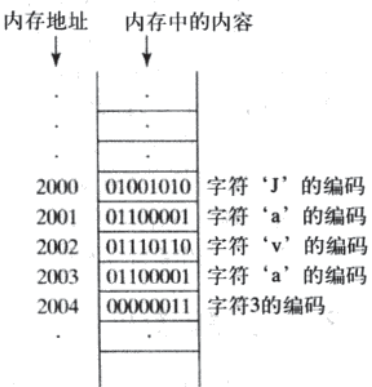


图1-2 内存存储数据和程序指令

1.2.3 存储设备

内存是不能长久保存数据的，因为断电时信息就会丢失。程序和数据都会被永久地存放在存储设备上，当计算机确实要使用它们时再移入内存，因为从内存读取比从存储设备读取要快得多。

存储设备主要有以下四种类型：

- 磁盘驱动器
- 光盘驱动器（只读光盘（CD-R）、可擦写光盘（CD-RW）和数字化视频磁盘（DVD））
- 磁带驱动器
- USB闪存驱动器

驱动器（drive）是对存储介质进行操作的设备，例如，磁盘、光盘和磁带等都是存储介质。

1. 磁盘

每台计算机至少有一个硬盘驱动器。硬盘（hard disk）是用来永久地存储数据和程序的。在最新的个人电脑上，硬盘容量一般为80GB~250GB。磁盘驱动器通常都安装在计算机内。此外，还可以用移动硬盘。

2. 光盘和数字化视频磁盘

CD的全称是致密的盘片。光盘驱动器的类型有两种：只读光盘（CD-R）和可擦写光盘（CD-RW）。只读光盘只能用于读取那些永久存储在光盘上的信息，内容一旦记录到光盘上，用户是不能修改它们的。可擦写光盘可以像硬盘一样使用，对这类光盘可以进行读取和重写操作。一张光盘的容量可以达到700MB。多数软件都是存储在只读光盘上发布的。大多数新型的个人电脑都安装了可擦写光驱，它既支持只读光盘也支持可擦写光盘。

DVD的全称是数字化多功能碟片或者是数字化视频磁盘。DVD和CD看起来很像，可以使用任何一种来存储数据。一张DVD上可以保存的信息要比一张CD保存的信息多。一张标准DVD的存储容量是4.7GB。

3. 磁带

磁带（tape）主要用于备份数据和程序。不同于磁盘和光盘，磁带是顺序存储信息的。计算机必须按照信息被存储的顺序来获取信息。磁带速度非常慢。备份1GB的硬盘需要一两个小时。新的趋势是使用闪存或者外挂硬盘来备份数据。

4. USB闪存驱动器

USB闪存驱动器（flash drive）是用于存储和传输数据的设备。闪存驱动器很小——大约就是一包口香糖的大小。它就像移动硬盘一样，可以插入计算机上的USB端口。USB闪存驱动器目前可用的最大存储容量为32GB。

1.2.4 输入和输出设备

用户是通过输入设备和输出设备与计算机进行通信的。通常，输入设备是指键盘（keyboard）和鼠标（mouse），而输出设备是指显示器（monitor）和打印机（printer）。

1. 键盘

计算机键盘（keyboard）很像一个附带完成某些特定功能的特殊键的打字机键盘。

功能键（function key）位于键盘的最上边，而且都是以F为前缀。它们的功能取决于软件的设置。

修饰符键（modifier key）是特殊键（例如，Shift、Alt和Ctrl），当它和另一个键同时按下时，会改变另一个键的常用功能。

数字小键盘（numeric keypad）位于键盘的右下角，是为了快速输入数字而独立出来的一套按键集合。

方向键（arrow key）位于主键盘和数字小键盘之间，用于上下左右地移动光标。

插入键（Insert）、删除键（Delete）、向上翻页键（Page Up）和向下翻页键（Page Down）都位于方向键的上方，分别用来在字处理过程中完成插入、删除、向上翻页和向下翻页的功能。

4 • 第1章 计算机、程序和Java概述

2. 鼠标

鼠标 (mouse) 是定点设备, 是用来在屏幕上移动光标的电子指针, 或者用于点击屏幕上的对象来触发它以响应这个动作。

3. 显示器

显示器 (monitor) 显示信息 (文本和图形)。屏幕分辨率和点距决定显示的质量。

屏幕分辨率 (screen resolution) 是指每平方英寸的像素数。像素 (“图像元素” 的简称) 就是在屏幕上构成图像的小点。对于一个17英寸的屏幕, 分辨率一般为宽1024像素、高768像素。分辨率可以手工设置。分辨率越高, 图像越锐化、越清晰。

点距 (dot pitch) 是指像素之间以毫米为单位的距离。点距越小, 显示效果越好。

1.2.5 通信设备

计算机可以通过通信设备进行联网, 例如, 拨号调制解调器 (modulator/demodulator, 调制器/解调器)、DSL、电缆调制解调器、网络接口卡以及无线通信设备。拨号调制解调器使用的是电话线, 传输数据的速度可以高达56 000bps (bps表示每秒比特)。DSL (Digital Subscriber Line, 数字用户线) 使用的也是电话线, 但是传输数据的速度比拨号调制解调器快20倍。电缆调制解调器利用电缆公司维护的有线电视电缆进行数据传输, 速度与DSL一样快。网络接口卡 (NIC) 是将计算机接入局域网 (LAN) 的设备。局域网通常用于大学、商业组织和政府组织。一种称为10BaseT的典型NIC能够以10Mbps (Mbps表示每秒百万比特) 的速度传输数据。无线通信正在变得越来越流行。现在, 每台手提电脑都配有无线适配器, 计算机可以通过无线适配器连接到Internet上。

1.3 程序

计算机程序 (program) 通常称为软件 (software), 是发给计算机的指令, 告诉计算机该做什么。因为计算机不理解人类的语言, 所以, 需要在计算机程序中使用计算机语言。程序设计 (programming) 就是创建一个可以让计算机执行并完成所需任务的程序。

计算机本身的语言因计算机类型的不同而有差异, 计算机本身的语言就是它的机器语言 (machine language) ——最初植入计算机的一套原始指令集。因为这些指令都是以二进制代码的形式存在, 所以, 为了告诉机器该做什么, 必须输入二进制代码。用机器语言进行程序设计是非常单调乏味的过程, 而且, 所编的程序也非常难以读懂和修改。例如, 为进行两数的相加, 可能必须写成如下的二进制形式:

```
1101101010011010
```

汇编语言 (assembly language) 是一种低级的程序设计语言, 它用助记符表示每一条机器语言指令。例如, 为进行两数相加, 用汇编代码所编写的指令形式如下:

```
ADD3 R1, R2, R3
```

汇编语言的出现降低了程序设计的难度。然而, 由于计算机不理解汇编语言, 所以需要使用一种称为汇编器 (assembler) 的程序将汇编语言程序转换为机器代码, 如图1-3所示。

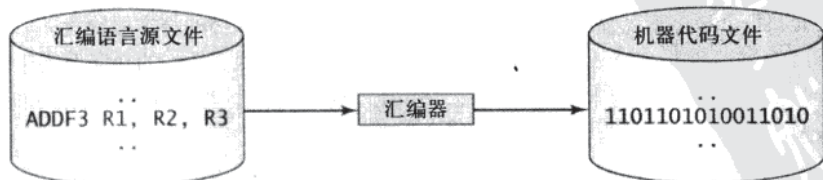


图1-3 汇编器将汇编语言指令翻译为机器码

汇编程序是用易于记忆的助记符形式的机器指令编写的。因为汇编语言具有机器依赖性, 所以汇编程序只能在某种特定的机器上执行。为了克服平台依赖性问题以及降低程序设计难度, 开发了高级语言。

高级语言 (high-level language) 很像英语, 易于学习和编写程序。例如, 下面是计算半径为5的圆面积的高级语言语句:

```
area = 5 * 5 * 3.1415;
```

在一百多种高级语言中, 以下几种是很著名的:

- COBOL (面向商业的通用语言)
- FORTRAN (公式翻译)
- BASIC (初学者通用符号指令代码)
- Pascal (以Blaise Pascal命名)
- Ada (以Ada Lovelace命名)
- C (由B的设计者开发)
- Visual Basic (Microsoft公司开发的类似Basic的可视化语言)
- Delphi (Borland公司开发的类似Pascal的可视化语言)
- C++ (基于C语言的一种面向对象程序设计语言)
- C# (Microsoft公司开发的类似Java的语言)
- Java

这里的每一种语言都是为了特定目的而设计的。COBOL是为商业应用而设计的, 现在主要用于商业数据处理。FORTRAN为数学运算而设计的, 主要用于数值计算。BASIC是为了易学易用而设计的。Ada是为美国国防部开发的, 主要用于国防项目。C语言具有汇编语言的强大功能以及高级语言的易学性和可移植性。Visual Basic和Delphi用于开发图形用户界面, 还可以进行快速应用开发。C++非常适合开发系统软件项目, 例如, 编写编译器和操作系统。Microsoft公司的Windows操作系统就是用C++编写的。C# (读作: C Sharp) 是由微软开发出来的新语言, 用来开发基于微软.NET平台的应用程序。Java是由Sun公司开发的, 广泛用于开发一些独立于平台的互联网应用程序。

用高级语言编写的程序称为源程序 (source program) 或源代码 (source code)。由于计算机不能理解源程序, 所以, 要使用称为编译器 (compiler) 的程序将源程序翻译成机器语言程序。然后, 这个机器语言程序再与其他辅助的库代码进行链接, 构成可执行文件, 该文件就可以在机器上运行, 如图1-4所示。在Windows平台上, 可执行文件的扩展名是.exe。

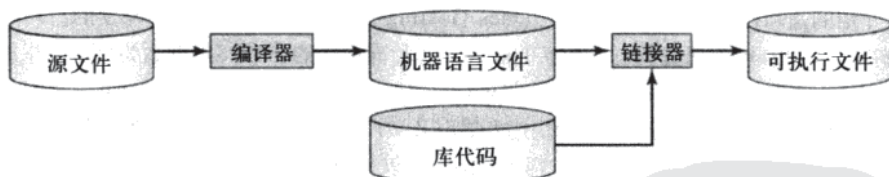


图1-4 源程序被编译为机器语言文件, 然后和系统库链接形成可执行文件

1.4 操作系统

操作系统 (Operating System, OS) 是运行在计算机上最重要的程序, 它可以管理和控制计算机的活动。流行的操作系统有Microsoft Windows、Mac OS以及Linux。如果没有操作系统, 像Web浏览器或者字处理程序这样的应用程序就不能运行。硬件、操作系统、应用软件和用户之间的关系如图1-5所示。

操作系统的主要任务有:

- 控制和监视系统的活动

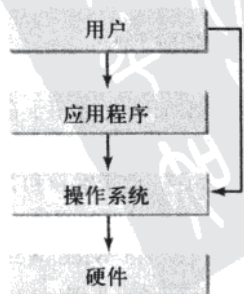


图1-5 操作系统是控制和管理系统的软件

6 • 第1章 计算机、程序和Java概述

- 分配和调遣系统资源
- 安排操作的顺序

1.4.1 控制和监视系统的活动

操作系统完成基本的功能，例如，识别来自键盘的输入，向显示器发送输出结果，保存磁盘中文件和目录的路径，控制类似硬盘驱动器和打印机这样的外部设备。操作系统还要确保不同的程序 and 用户同时使用计算机时不会相互干扰，而且要对安全负责，以确保未经授权的用户无权访问系统。

1.4.2 分配和调遣系统资源

操作系统负责确定一个程序需要使用哪些计算机资源（例如，CPU、内存、磁盘、输入和输出设备），并进行资源分配和调遣以运行程序。

1.4.3 安排操作的顺序

操作系统负责确定程序的执行顺序，以便有效地利用系统资源。为了提高系统的性能，目前许多操作系统都支持像多道程序设计（multiprogramming）、多线程（multithreading）和多处理（multiprocessing）这样的技术。

多道程序设计允许多个程序通过共享CPU同时运行。CPU的速度比其他组件快得多，这样，多数时间它都处于空闲状态，例如，在等待数据从磁盘或其他资源传入时。多道程序设计操作系统利用这一特点，允许多个程序同时使用CPU，一旦CPU空闲就让别的程序使用它。例如，在Web浏览器下载文件的同时，可以用字处理程序来编辑文件。

多线程允许在一个程序内部进行并发处理，这样，程序的子任务就可以同时运行。例如，字处理程序允许用户在编辑文本的同时，将其保存到文件。在这个例子中，编辑和保存是同一个应用程序的两个不同任务。这两个任务是并发运行的独立线程。

多处理也称为并行处理（parallel processing），是指使用两个或多个处理器共同完成同一个任务。它就像在外科手术中多名医生同时给一个病人做手术一样。

1.5 Java、万维网以及其他

本书介绍的是Java程序设计。Java是James Gosling在Sun公司领导的小组开发的。Java最初被称为Oak（橡树），是1991年为消费类电子产品的嵌入式芯片而设计的。1995年更名为Java，并重新设计用于开发Internet应用程序。关于Java的历史，参见www.java.com/en/javahistory/index.jsp。

近年来，Java变得非常流行。Java的快速发展以及被广泛接受都应归功于它的设计特性，特别是它的承诺：一旦编写了一个程序，在任何地方都可以运行。就像Sun公司声称的，Java是简单的（simple）、面向对象的（object oriented）、分布式的（distributed）、解释型的（interpreted）、健壮的（robust）、安全的（secure）、结构中立的（architecture neutral）、可移植的（portable）、高性能的（high performance）、多线程的（multithreaded）和动态的（dynamic）。关于Java特性的剖析，参见www.cs.armstrong.edu/liang/JavaCharacteristics.pdf。

Java是功能完善的通用程序设计语言，可以用来开发可靠的、要求严格的应用程序。现在，它不仅用于Web程序设计，而且用于在服务器、台式机和移动设备上开发跨平台的独立应用程序。用它开发过与火星探测器通信并控制其在火星上行走的代码。许多曾经认为Java言过其实的公司现在使用Java开发分布式应用程序，便于客户和合作伙伴在Internet上访问。现在，一旦开发新的项目，公司都会考虑如何利用Java使工作变得更加容易。

万维网（World Wide Web，WWW）是从世界上任何地方的Internet都可以访问的电子信息宝库。Internet作为万维网的基础架构已经问世三十多年。丰富多彩的万维网和设计精良的Web浏览器是Internet流行的主要原因。

万维网上的主要创作语言是超文本标记语言 (Hypertext Markup Language, HTML)。HTML是一种简单的语言：能够部署Internet上的文档、链接互联网上的文档，能够在万维网上提供生动的图像、声音和视频。但是，除了简单的表单外，它不能和用户进行交互。用HTML制作的Web网页基本上都是静态的，也是很单调的。

Java一开始就富有吸引力，因为Java程序可以在Web浏览器中运行。这种能在Web浏览器中运行的Java程序称为Java小程序 (applet)。applet使用流行的图形用户界面与网络用户进行交互，处理用户的要求，这些界面中包括了按钮、文本字段、文本域、单选按钮等。applet使得网页更加灵活、生动和易于交互。图1-6显示了一个在Web浏览器上运行井字游戏的applet。

在Web浏览器中输入这个URL

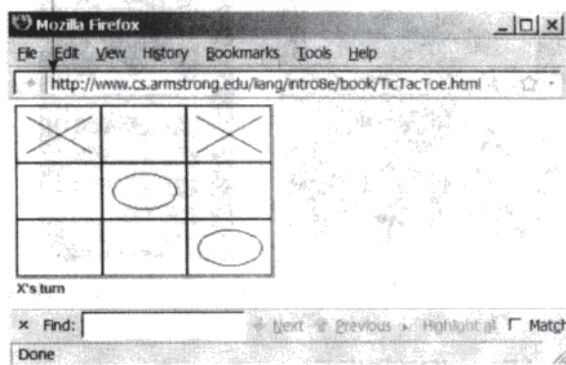


图1-6 井字游戏的Java applet内嵌在一个HTML页面中

提示 要查看Java applet的演示，请访问网站java.sun.com/applets。该网站提供了丰富的Java资源，还有很多其他很酷的演示applet的站点链接。java.sun.com是Sun公司的Java官方网站。

Java还可以用来开发服务器端的应用程序。这些应用程序可以在Web服务器上运行，生成动态网页。如图1-7所示，这是本书的自动打分系统，该系统就是用Java开发出来的。

在Web浏览器中输入这个URL

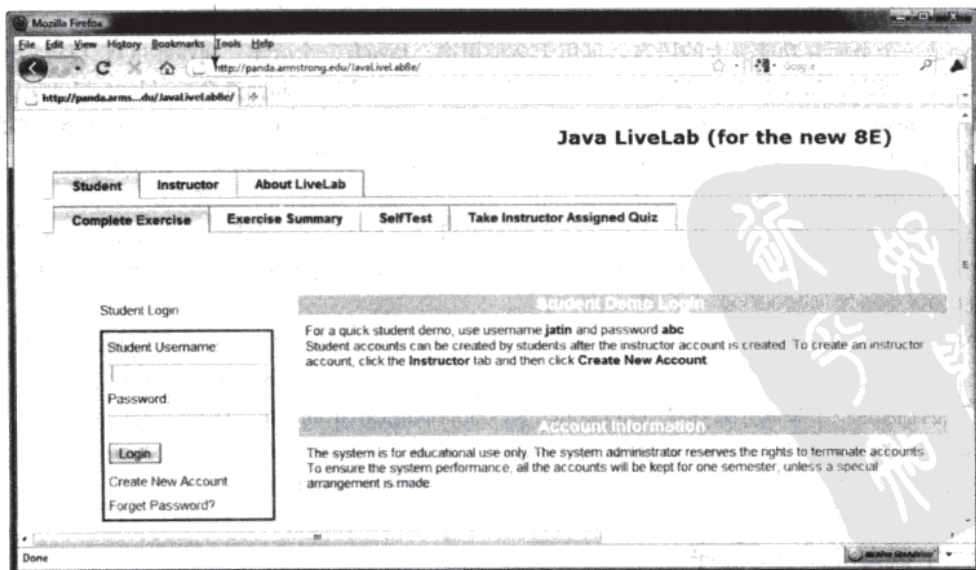


图1-7 用Java开发和本书配套的自动打分系统

8 • 第1章 计算机、程序和Java概述

Java是一个功能强大的程序设计语言，可以用它来开发台式机和服务器上的应用程序，也可以用它来开发小型手持设备上的应用程序。图1-8演示了用Java编写的在个人数字助理BlackBerry和手机上显示的日历。

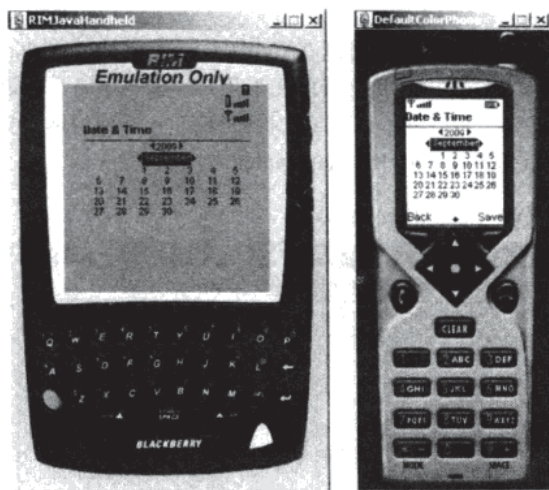


图1-8 可以使用Java来开发手持设备和无线设备上的应用程序，如BlackBerry（左）和手机（右）

1.6 Java语言规范、API、JDK和IDE

计算机语言有严格的使用规范。如果编写程序时没有遵循这些规则，计算机就不能理解程序。Java语言规范和Java API定义Java的标准。

Java语言规范（Java language specification）是对语言的技术定义，它包括Java程序设计语言的语法和语义。完整的Java语言规范可以在java.sun.com/docs/books/jls上找到。

应用程序接口（Application Program Interface, API）包括为开发Java程序而预定义的类和接口。Java语言的规范是稳定的，但是API一直在扩展。在Sun公司的Java网站上（java.sun.com），可以查看和下载最新版的Java API。

Java是一个全面且功能强大的语言，可用于多种用途。Java有三个版本：Java标准版（Java Standard Edition, Java SE）、Java企业版（Java Enterprise Edition, Java EE）、Java微型版（Java Micro Edition, Java ME）。Java SE可以用来开发客户端独立的应用程序或applet。Java EE可以用来开发服务器端的应用程序，例如，Java servlet和JavaServer Pages。Java ME可以用来开发移动设备的应用程序，例如，手机等。本书使用Java SE介绍Java程序设计。

Java SE也有很多版本，本书采用最新的版本Java SE 6。Sun公司发布Java的各个版本都带有Java开发工具包（Java Development Toolkit, JDK），Java SE 6对应的Java开发工具包称为JDK 1.6（也称为Java 6或者JDK 6）。

JDK是由一套独立程序构成的集合，每个程序都是从命令行调用的，用于开发和测试Java程序。除了JDK，还可以使用某种Java开发工具（例如，NetBeans、Eclipse和TextPad）——它们是为了快速开发Java程序而提供的一个集成开发环境（Integrated Development Environment, IDE）的软件。编辑、编译、链接、调试和在线帮助都集成在一个图形用户界面中，这样，只需在一个窗口中输入源代码或在窗口中打开已有的文件，然后点击按钮、菜单选项或者使用功能键就可以编译和运行源代码。

1.7 一个简单的Java程序

我们从一个简单的Java程序开始，该程序在控制台上显示消息“Welcome to Java!”。控制台

(console) 是指计算机的文本输入和显示设备。该程序如程序清单1-1所示。

程序清单1-1 Welcome.java

```
1 public class Welcome {
2     public static void main(String[] args) {
3         // Display message Welcome to Java! to the console
4         System.out.println("Welcome to Java!");
5     }
6 }
```

Welcome to Java!



显示行号 (line number) 是为了引用方便, 它们并不是程序的一部分。所以, 不要在程序中敲入行号。

第1行定义了一个类。每个Java程序至少应该有一个类。每个类都有一个名字。按照惯例, 类名都是以大写字母开头的。本例中, 类名 (class name) 为Welcome。

第2行定义主方法 (main method)。为了运行某个类, 该类必须包含名为main的方法。程序是从main方法开始执行的。

方法是包含语句的结构体。本程序中的main方法包括了System.out.println语句。该语句在控制台上打印消息“Welcome to Java!” (第4行)。Java中的每条语句都以分号 (;) 结束, 也称为语句结束符 (statement terminator)。

保留字 (reserved word) 或关键字 (keyword) 对编译器而言都是有特定含义的, 所以不能在程序中用于其他目的。例如, 当编译器看到字class时, 它能知道class后面的字就是这个类的名字。这个程序中的其他保留字还有public、static和void。

第3行是注释 (comment), 它标注该程序是干什么的, 以及它是如何构建的。注释帮助程序员进行相互沟通以及理解程序。注释不是程序设计语句, 所以编译器编译程序时是忽略它们的。在Java中, 在单行上用两个斜杠 (//) 引导注释, 称为行注释 (line comment); 在一行或多行用/*和*/括住注释, 称为块注释 (block comment)。当编译器看到//时, 就会忽略本行//之后的所有文本。当看到/*时, 它会搜索接下来的*/, 并忽略掉/*与*/之间的文本。下面是这两种注释的例子:

```
// This application program prints Welcome to Java!
/* This application program prints Welcome to Java! */
/* This application program
   prints Welcome to Java! */
```

程序中的一对花括号将程序的一些组件组合起来, 形成一个块 (block)。在Java中, 每个块以左括号 ({) 开始, 以右括号 (}) 结束。每个类都有一个将该类的数据和方法放在一起的类块 (class block)。每个方法都有一个将该方法中的语句放在一起的方法块 (method block)。块是可以嵌套的, 即一个块可以放到另一个块内, 如下面代码所示。

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

类块

方法块

提示 一个左括号必须匹配一个右括号。任何时候, 当你输入一个左括号时, 应该立即输入一个右括号来防止出现遗漏括号的错误。大多数Java IDE都会自动地为每个左括号插入一个右括号。

注意 你可能想知道为什么main方法要以这样的方式声明, 为什么使用System.out.println(...) 就可以在控制台上显示信息。在现阶段, 你只需知道它们就是这么做的就可以。这一问题将在后续的章节中得到完整的回答。

警告 Java源程序是区分大小写的。例如, 在该程序中用Main代替main是错误的。

注意 像其他任何一种程序设计语言一样，Java也有自己的语法，而且你必须按照语法规则编写代码。如果你的程序违反了语法规则，例如，忘记了分号，忘记了花括号，忘记了引号，或者拼错了关键字String，Java编译器会报告语法错误。尝试去编译带有这些错误的程序，看看编译器会报告些什么。

程序清单1-1中的程序会显示一条消息。一旦你理解了这个程序，很容易将该程序扩展为显示更多的信息。例如，可以改写该程序来显示三条消息，如程序清单1-2所示。

程序清单1-2 Welcome1.java

```
1 public class Welcome1 {
2     public static void main(String[] args) {
3         System.out.println("Programming is fun!");
4         System.out.println("Fundamentals First");
5         System.out.println("Problem Driven");
6     }
7 }
```

```
Programming is fun!
Fundamentals First
Problem Driven
```



还可以进一步完成科学计算，并将结果显示到控制台上。程序清单1-3给出计算 $\frac{10.5 + 2 \times 3}{45 - 3.5}$ 的例子。

程序清单1-3 ComputeExpression.java

```
1 public class ComputeExpression {
2     public static void main(String[] args) {
3         System.out.println((10.5 + 2 * 3) / (45 - 3.5));
4     }
5 }
```

```
0.39759036144578314
```



Java中的乘法运算符是*。如你所看到的，将一个数学表达式翻译成Java表达式是一个非常直观的过程，我们将在第2章进一步讨论Java表达式。

1.8 创建、编译和执行Java程序

在执行程序之前，必须创建程序并进行编译。这个过程是反复执行的，如图1-9所示。如果程序有编译错误，必须修改程序来纠正错误，然后重新编译它。如果程序有运行错误或者不能产生正确的结果，必须修改这个程序，重新编译，然后重新执行。

可以使用任何一个文本编辑器（editor）或者集成开发环境来创建和编辑Java源代码文件。本节演示如何从命令窗口创建、编译和运行Java程序。如果希望使用像Eclipse、NetBeans或者TextPad这样的IDE，请参考补充材料II。从命令窗口，可以使用记事本（NotePad）来创建Java源代码文件，如图1-10所示。

注意 源文件的扩展名必须是java，而且文件名必须与公用类名完全相同。例如，程序清单1-1中源代码的文件必须命名为Welcome.java，因为公用类的类名就是Welcome。

Java编译器将Java源文件翻译成Java字节码文件。下面的命令就是用来编译Welcome.java的：

```
javac Welcome.java
```

注意 在编译和运行程序前必须先安装和配置JDK。补充材料I.B介绍如何安装JDK以及如何设置Java程序的编译和运行环境。如果你在编译和运行Java程序的过程中遇到问题，请参考补充材料I.C，这个补充材料还解释了如何使用基本的DOS命令，以及如何使用Windows记事本

(NotePad) 和写字板 (WordPad) 来创建和编辑文件。所有补充材料都可以在本书配套的网站上访问到。

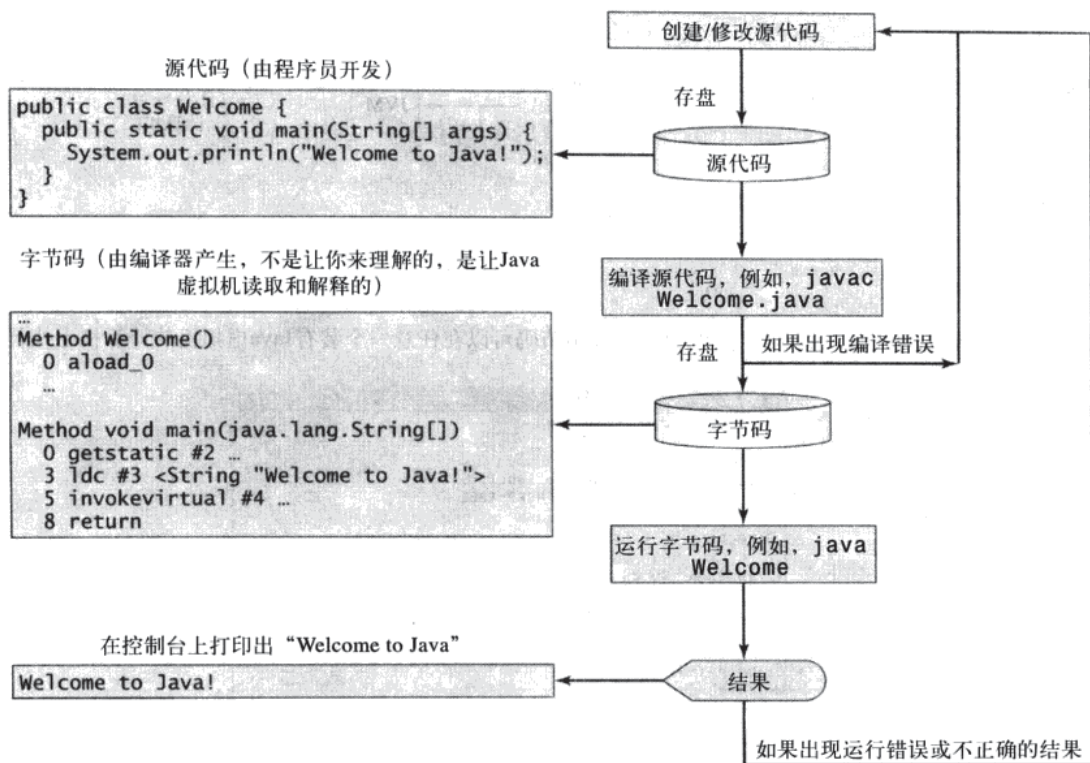


图1-9 Java程序开发过程就是重复地创建/修改源代码、编译和执行程序

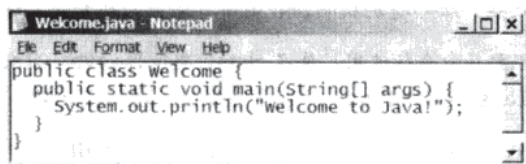


图1-10 可以使用Windows记事本创建Java源程序文件

如果没有语法错误, 编译器 (compiler) 就会生成一个扩展名为.class的字节码文件。所以, 前面的命令会生成一个名为Welcome.class的文件, 如图1-11a所示。Java语言是高级语言, 而Java字节码是低级语言。字节码类似于机器指令, 但它是体系结构中的, 是可以在任何带Java虚拟机 (JVM) 的平台上运行的, 如图1-11b所示。虚拟机不是物理机器, 而是一个解释Java字节码的程序。这正是Java的主要优点之一: Java字节码可以在不同的硬件平台和操作系统上运行。

执行Java程序就是运行程序的字节码, 可以在任何一个装有JVM的平台上运行字节码, 解释Java字节码。解释的过程就是一次将字节码中单独的一步翻译为目标机器语言代码, 而不是将整个程序翻译成单独的一块。翻译完一步之后就立即执行这一步。

下述命令用来运行字节码:

```
java Welcome
```

图1-12显示了用于编译Welcome.java的命令javac。编译器生成Welcome.class文件, 使用命令java

执行这个文件。

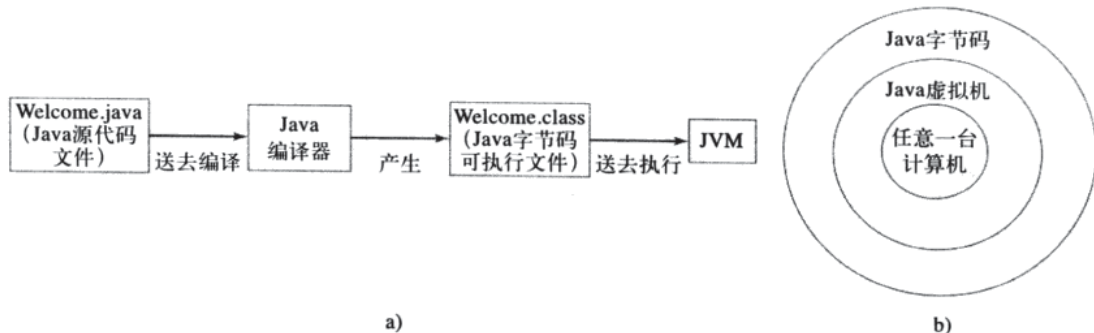


图1-11 a) Java源代码被翻译为字节码；b) Java字节码可以在任意一个装有Java虚拟机的计算机上执行

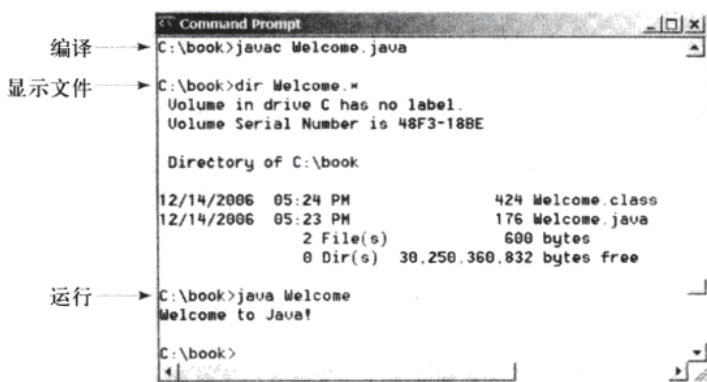


图1-12 程序清单1-1的输出显示消息“Welcome to Java!”

注意 为了简单性和一致性，除非特别指明，否则所有的源代码和类文件都放在c:\book下。

警告 在执行程序时，不要在命令行使用扩展名.class。要使用java ClassName来运行程序。如果在命令行使用java ClassName.class，系统就会尝试去读取ClassName.class.class。

提示 如果要执行一个不存在的类，就会出现NoClassDefFoundError的错误。如果执行的类文件中没有main方法或敲错了main方法（例如，将main错敲成Main），则会出现提示NoSuchMethodError。

注意 在执行一个Java程序时，JVM首先会用一个称为类加载器（class loader）的程序将类的字节码加载到内存中。如果你的程序中使用其他类，类加载程序会在需要它们之前动态地加载它们。当加载该类后，JVM使用一个称为字节码验证器（bytecode verifier）的程序来检验字节码的合法性，确保字节码不会违反Java的安全规范。Java强制执行严格的安全规范，以确保来自网络的Java程序不会危害到你的计算机。

教学注意 教师可能需要学生使用包来组织程序。例如，你可能将本章的所有程序都放在一个名为chapter1的包里。要得到如何使用包的指南，请参考补充材料I.F。

1.9 （GUI）在消息对话框中显示文本

如图1-12所示，程序清单1-1中的程序是在控制台上显示文本。可以改写该程序，在消息对话框中显示

文本。要实现这个目标，需要使用JOptionPane类中的showMessageDialog方法。JOptionPane是Java系统中众多预定义的类之一，这些类可以反复使用，而不必每次“重新编写”。可以使用showMessageDialog方法在消息对话框中显示任意文本，如图1-13所示。程序清单1-4给出这个新的程序。

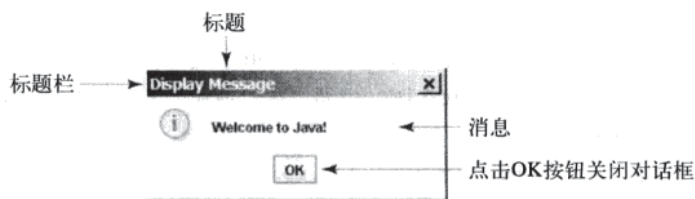


图1-13 在消息对话框中显示字符串“Welcome to Java!”

程序清单1-4 WelcomeInMessageDialogBox.java

```
1 /* This application program displays Welcome to Java!
2  * in a message dialog box.
3  */
4 import javax.swing.JOptionPane;
5
6 public class WelcomeInMessageDialogBox {
7     public static void main(String[] args) {
8         // Display Welcome to Java! in a message dialog box
9         JOptionPane.showMessageDialog(null, "Welcome to Java!");
10    }
11 }
```

这个程序使用一个Java类JOptionPane（第9行）。Java中预定义的分类组放在包中。JOptionPane放在包javax.swing中。使用第4行的import语句将JOptionPane导入，这样，编译器无须全称javax.swing.JOptionPane就可以找到类JOptionPane的位置。

注意 如果将第9行的JOptionPane用javax.swing.JOptionPane来替换，就无须在第4行导入它。javax.swing.JOptionPane是JOptionPane类的全称。

showMessageDialog方法是一个静态方法。这样的方法是要通过类名后面紧跟一个圆点运算符（.）以及带参数的方法名来调用的。方法将在第5章中介绍。调用showMessageDialog方法需要两个参数，如下所示。



第一个参数总是null。null是Java的关键字，将在第8章中详细介绍。第二个参数是要显示出来的文本字符串。

还有几种使用showMessageDialog方法的途径，目前你只需要知道两种方式就可以了。一种方式是使用如下面例子所示的语句：

```
JOptionPane.showMessageDialog(null, x);
```

这里的x就是要显示的文本字符串。

另一种方式是使用下面的语句：

```
JOptionPane.showMessageDialog(null, x,
    y, JOptionPane.INFORMATION_MESSAGE);
```

这里的x也是要显示的文本字符串，而y是表示消息对话框标题的字符串。第四个参数可以是JOptionPane.INFORMATION_MESSAGE，它是为了让消息框能够显示图标（i），如下面这个例子所示。



注意 `import`语句有两种类型：明确导入（specific import）和通配符导入（wildcard import）。明确导入是指将某个单独的类放在`import`语句中。例如，下面的语句就是从包`javax.swing`中导入`JOptionPane`。

```
import javax.swing.JOptionPane;
```

通配符导入是指导入一个包中所有的类。例如，下面的语句就是从包`javax.swing`中导入所有类。

```
import javax.swing.*;
```

除非要在程序中使用某个类，否则关于被导入包中的这些类的信息在编译时或运行时是不被读取的。导入语句只是告诉编译器在什么地方能找到这些类。声明明确导入和声明通配符导入在性能上是没有什么差别的。

注意 回顾程序清单1-1中的语句`System.out.println("Welcome to Java");`中用到的`System`类。这个`System`类并没有被导入，那是因为它在`java.lang`包内。`java.lang`包内的所有类在每个Java程序中都将被隐式导入。

关键术语

.class file（.class文件）

.java file（.java文件）

assembly language（汇编语言）

bit（比特）

block（块）

block comment（块注释）

bus（总线）

byte（字节）

bytecode（字节码）

bytecode verifier（字节码验证器）

cable modem（电缆调制解调器）

central processing unit（CPU，中央处理器）

class loader（类加载器）

comment（注释）

compiler（编译器）

console（控制台）

dot pitch（点距）

DSL（Digital Subscriber Line，数字用户线）

hardware（硬件）

high-level language（高级语言）

Integrated Development Environment

（IDE，集成开发环境）

java command（java命令）

javac command（javac命令）

Java Development Toolkit（JDK，Java开发工具包）

Java Virtual Machine（JVM，Java虚拟机）

keyword or reserved word（关键字或保留字）

line comment（行注释）

machine language（机器语言）

main method（main方法）

memory（内存）

modem（调制解调器）

Network Interface Card（NIC，网络接口卡）

Operation System（OS，操作系统）

pixel（像素）

program（程序）

programming（程序设计）

resolution（分辨率）

software（软件）

source code（源代码）

source file（源文件）

specific import（明确导入）

storage devices（存储设备）

statement（语句）

wildcard import（通配符导入）

注意 上面的术语都是本章所定义的。补充材料I.A按照章节顺序列出了本书所有的关键术语及其说明。

本章小结

- 计算机是存储和处理数据的电子设备。
- 计算机包括硬件和软件两部分。
- 硬件是计算机中可以看得见的物理部分。
- 计算机程序，也就是通常所说的软件，是一些看不见的指令，它们控制硬件完成任务。
- 计算机程序设计就是编写计算机执行的指令（即代码）。
- 中央处理器（CPU）是计算机的大脑。它从内存获取指令并且执行这些指令。
- 计算机使用0或1，因为数字设备有两个稳定的状态，习惯上就是指0和1。
- 一个比特是指二进制数0或1。
- 一个字节是指8比特的序列。
- 一个千字节大约是1000字节，一个兆字节大约是100万字节，一个千兆字节大约是10亿字节，而一个万亿字节大约是1000千兆字节。
- 内存存储CPU要执行的数据和程序指令。
- 内存单元是字节的有序序列。
- 内存是不能长久保存数据的，因为断电时信息就会丢失。
- 程序和数据永久地保存在存储设备里，当计算机确实需要使用它们时被移入内存。
- 机器语言是一套植入每台计算机的原始指令集。
- 汇编语言是一种低级程序设计语言，它用助记符表示每一条机器语言的指令。
- 高级语言很像英语，易于学习和编写程序。
- 用高级语言编写的程序称为源程序。
- 编译器是将源程序翻译成机器语言的软件程序。
- 操作系统（OS）是管理和控制计算机活动的程序。
- Java是平台无关的，这意味着只需编写一次程序，就可以在任何地方运行。
- Java程序可以内嵌在HTML网页内，通过Web浏览器下载，给Web客户带来生动的动画和灵活的交互性。
- Java源程序文件以扩展名.java结束。
- 每个类都被编译成一个独立的字节码文件，该文件名与类名相同，扩展名为.class。
- 使用javac命令可以从命令行编译Java源代码文件。
- 使用java命令可以从命令行运行Java类。
- 每个Java程序都是一套类定义的集合。关键字class引入类的定义，类的内容包含在块内。
- 一个块以左花括号（{）开始，以右花括号（}）结束。方法包含在类中。
- 每个Java程序必须有一个main方法。main方法是程序开始执行的入口。
- Java中的每条语句都是以分号（；）结束的，也称该符号为语句结束符。
- 保留字，或者称关键字，对编译器而言都有特殊含义，在程序中不能用于其他目的。
- 在Java中，在单行上用两个斜杠（//）引导注释，称为行注释；在一行或多行用/*和*/括住注释，称为块注释。
- Java源程序是区分大小写的。
- import语句有两种类型：明确导入和通配符导入。明确导入是指将某个单独类放在import语句中。通配符导入是指导入一个包中所有的类。

复习题

注意 复习题的答案在本书配套网站上。

16 • 第1章 计算机、程序和Java概述

1.2~1.4节

- 1.1 给出硬件和软件的定义。
- 1.2 列举计算机的主要组件。
- 1.3 给出机器语言、汇编语言和高级程序设计语言的定义。
- 1.4 什么是源程序？什么是编译器？
- 1.5 什么是JVM？
- 1.6 什么是操作系统？

1.5~1.6节

- 1.7 描述Java的历史。Java可以在任何机器上运行吗？在计算机上运行Java时需要什么？
- 1.8 Java编译器的输入和输出是什么？
- 1.9 列举一些Java的开发工具。像NetBeans、Eclipse之类的工具是与Java不同的语言，还是Java的替代品或是对Java的扩展？
- 1.10 Java与HTML之间的关系是什么？

1.7~1.9节

- 1.11 解释Java关键字的含义，罗列你在本章中学到的一些关键字。
- 1.12 Java区分大小写吗？Java关键字是大写还是小写？
- 1.13 Java源文件的扩展名是什么？Java字节码文件的扩展名是什么？
- 1.14 什么是注释？注释能被编译器忽略吗？该如何表示一个注释行和一个注释段？
- 1.15 在控制台上显示字符串的语句是什么？在消息对话框中显示消息“Hello world”的语句是什么？
- 1.16 下面的程序是错的。重新排行使程序在morning之后显示afternoon。

```
public static void main(String[] args) {
}
```

```
public class Welcome {
    System.out.println("afternoon");
    System.out.println("morning");
}
```

- 1.17 找出并修改下面代码的错误：

```
1 public class Welcome {
2     public void Main(String[] args) {
3         System.out.println('Welcome to Java!');
4     }
5 }
```

- 1.18 编译Java程序的命令是什么？运行Java程序的命令是什么？
- 1.19 如果在运行程序时出现NoClassDefFoundError，产生这个错误的原因是什么？
- 1.20 如果在运行程序时出现NoSuchMethodError，产生这个错误的原因是什么？
- 1.21 为什么System类无须被导入？
- 1.22 以下两个import语句在性能上有没有不同之处？

```
import javax.swing.JOptionPane;
```

```
import javax.swing.*;
```

- 1.23 显示以下代码的输出结果：

```
public class Test {
    public static void main(String[] args) {
        System.out.println("3.5 * 4 / 2 - 2.5 is ");
        System.out.println(3.5 * 4 / 2 - 2.5);
    }
}
```



编程练习题

注意 偶数号练习的答案在本书配套的网站上。所有练习的答案都放在教师资源网站上。难度等级分为容易（没有星号）、适度（*）、难（**）和非常难（***）。

- 1.1（显示三条消息）编写程序，显示Welcome to Java、Welcome to Computer Science和Programming is fun。
- 1.2（显示五条消息）编写程序，显示Welcome to Java五次。
- *1.3（显示图案）编写一个程序，显示下面的图案：

```

      J      A      V      V      A
      J      A A      V      V      A A
J      J      AAAAA      V V      AAAAA
      J J      A      A      V      A      A

```

- 1.4（打印表格）编写程序，显示以下表格：

a	a^2	a^3
1	1	1
2	4	8
3	9	27
4	16	64

- 1.5（计算表达式）编写程序，显示 $\frac{9.5 \times 4.5 - 2.5 \times 3}{45.5 - 3.5}$ 的结果。
- 1.6（数列求和）编写程序，显示 $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$ 的结果。
- 1.7（近似求 π ）可以使用以下公式计算 π ：

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} + \cdots \right)$$

编写程序，显示 $4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} + \cdots \right)$ 的结果。在程序中用1.0代替1。

资源
程序
学习
PDG