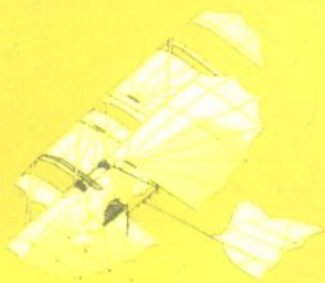


廖弘智

黄宗仁

著



Java

轻松上手



INTERNET 实务系列丛书

机械工业出版社

Internet 实务系列丛书

Java 轻松上手

廖弘智、黄宗仁 著

机械工业出版社

JS189/117

本书通过大量的实例对 Java 进行了全面的介绍。使读者能够轻松、快速地了解 Java，从而编写具有动态展示效果的 homepage。内容由浅入深，涉及 Java 的基础知识、建立 Java 程序的开发环境、图形用户界面（GUI）的建立、多线程程序设计、Java 的输入输出以及 Java 的网络传输功能。

本书为读者提供了学习 Java 的捷径，在内容设计上考虑了不同的层次的要求，是轻松学习 Java 的一本好的教程及参考书。

本书繁体字版名为《Java 轻松上手》，由第三波文化事业股份有限公司出版，版权归第三波文化事业股份有限公司所有。本书简体字中文版由第三波文化事业股份有限公司依出版授权合同约定，授权机械工业出版社依出版授权合同约定出版，未经出版者书面许可，本书的任何部分均不得以任何形式或手段复制或传播。

本书版权登记号：图字：01-96-1265

图书在版编目（CIP）数据

Java 轻松上手/廖弘智著.-北京：机械工业出版社，1997.1

（Internet 实务系列丛书）

ISBN 7-111-05473-3

I.J... I. 廖... II. Java 语言-程序设计 N.TP312

中国版本图书馆 CIP 数据核字（96）第 23786 号

出 版 人：马九荣（北京市百万庄南街 1 号 邮政编码：100037）

责任编辑：何伟新

三河永和印刷有限公司印刷·新华书店北京发行所发行

1997 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16 14.5 印张·362 千字

0 001-6 000 册

定价：26.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

序

近几年来，由于网络的快速发展，已经使得原本鲜为人知的网络世界在一瞬间就变成家喻户晓的名词。更使得原本只是用于学术用途上默默无闻的学术网络，一跃而成为全球通行的主流网络，再加上愈来愈多样化的应用纷纷出现，使得网际网络的用途也愈来愈多，几乎到了无所不包的地步。

其中目前在网际网络上最为流行的应用要算是 WWW (World Wide Web) 了，它除了可以让我们看看某家公司或某个人的 Homepage 外，还可以让我们在某公司的 Homepage 上留下信息，或者利用它来购买该公司所销售的物品。不过基本上你并不能在自己的 Homepage 上做太复杂的动作。

Java 的出现不但可以让你的 Homepage 更生动，由于它也是一种程序语言，所以可以拿它来设计一般的应用程序。除此之外，由于 Java 特殊的设计，可以将使用 Java 写好的程序编译成一种标准的机器码，这样不管所使用的机器是什么，只要在机器上有一个可以执行该机器码的解释器，便可以执行该程序。由于 Java 的这项特性，也就可能造成电脑软件流通方式的改变，并进而造成电脑使用方式的改变。

至于 Java 到底是否真会成功，就目前许多厂商投入的情形看来，Java 确实很有希望成功的。为了让大家能够更快的进入 Java 程序设计领域，所以笔者特地搜集许多 Java 相关资料，并实际测试 Java 所提供的各项功能，试图以最简单的例子来说明其用法，减少读者测试的时间。除此以外，笔者也将自己多年来程序设计的经验融入本书中，如在面向对象程序的基本概念介绍，图形使用界面的程序设计，文件存取，网络传输功能等章节中皆可见一斑。

最后虽然我们尽量追求本书内容及编排的完整性及正确性，不过错误仍在所难免，望请各位读者不吝指教。

廖弘智、黄宗仁

1996 年 7 月

目 录

第 1 章 Java 简介	1	5.3 如何在 PAGE 中加入 Applet	74
1.1 Java 是什么?	1	5.3.1 <Applet>	75
1.2 Java 的特色?	2	5.3.2 Applet 的参数传递	76
1.3 Java 的应用	3	5.4 Applet 的绘图	78
1.4 Java 概观	6	5.4.1 画线	78
1.5 如何使用本书	10	5.4.2 矩形	78
第 2 章 程序设计环境的建立	11	5.4.3 圆角矩形	79
2.1 如何安装 JDK	11	5.4.4 立体矩形	80
2.2 如何安装 API 说明文件	12	5.4.5 多边形	81
第 3 章 面向对象程序设计	13	5.4.6 椭圆形	83
3.1 面向对象的基本概念	13	5.4.7 画弧	84
3.1.1 对象 (object)	13	5.5 颜色和字体设置	85
3.1.2 类 (Class)	16	5.5.1 颜色的设置	85
3.1.3 对象的一生	18	5.5.2 字体设置	85
3.2 Java 的面向对象支持	19	第 6 章 图形用户界面 (GUI)	87
3.3 面向对象程序实现	23	6.1 Java 的图形用户界面概念	87
第 4 章 Java 的语法	35	6.2 基本用户界面对象	88
4.1 变量与常量	35	6.2.1 标签 (Label)	88
4.2 Java 的表达式及运算符	46	6.2.2 按钮 (Button)	90
4.2.1 布尔值的运算符	46	6.2.3 正文字段 (TextField)	91
4.2.2 整数的运算符	47	6.2.4 复选框 (Check Box)	95
4.2.3 浮点数的运算符	49	6.2.5 单选按钮 (Radio Button) ...	97
4.2.4 字符的运算符	49	6.2.6 选项菜单 (Choice Menu) ...	98
4.2.5 对象的运算符	49	6.2.7 列表 (List)	99
4.2.6 字符串的运算符	49	6.2.8 滚动条 (Scrollbar)	101
4.2.7 其它的运算符	51	6.3 事件的处理	103
4.3 Java 的基本语句	51	6.3.1 与鼠标相关的事件	104
4.4 流程控制语句	52	6.3.2 与键盘相关的事件	109
4.4.1 判断结构	52	6.3.3 事件处理函数 (handleEvent) ...	114
4.4.2 循环结构	54	6.4 GUI 组件的布局	123
4.4.3 其它流程控制语句	55	6.4.1 FlowLayout	123
4.5 命令行参数	58	6.4.2 BorderLayout	124
4.6 异常情态 (EXCEPTION)	60	6.4.3 CardLayout	127
4.7 一些常用的类	64	6.4.4 GridLayout	128
第 5 章 Applet 程序设计	69	6.4.5 GridBagLayout 和 GridBagCon-	
5.1 Applet 概论	69	straints	129
5.2 Applet 的建立	70	第 7 章 高级 GUI 设计	135

7.1 使用 Panel	135	9.2 Java 的标准输入与输出	176
7.2 建立视窗	141	9.3 Java 的目录及文件管理	180
7.3 对话框的建立	147	9.4 Java 的顺序文件存取	186
7.4 文件对话框 (FileDialog)	150	9.5 Java 的随机文件存取	189
7.5 建立菜单	155	9.6 Java 的管道功能	196
第 8 章 多线程程序设计	162	9.7 输入输出过滤器	200
8.1 线程的基本概念	162	9.8 词法分析	207
8.2 单线程	163	第 10 章 Java 的网络传输功能	211
8.3 多线程	169	10.1 使用 URL 取得网络资源	211
第 9 章 Java 的输入及输出	174	10.2 Java 网络通信 (TCP)	215
9.1 Java 的输入与输出	174	10.3 Java 网络通信 (UDP)	221

第 1 章 Java 简介

1.1 Java 是什么?

最近一两年,许多报章杂志争相报导,在电信局、经济决策部门及一些网络服务开发厂商(ISP)的推广下,已经有愈来愈多的人加入 Internet 的使用行列。这些新加入的使用者除了想要使用 Internet 上所拥有的庞大资源外,最大的原因应该是受到 WWW(World Wide Web)的便利性及其所提供的多姿多彩的世界所吸引。其实 Internet 之所以会发展的如此快速,其中最大的功臣非 WWW 莫属了。因为 WWW 所提供的超媒体文件表示方式,大大提高了用户界面的友好性,使得许多应用成为可能。正因为 WWW 具备了如此耀眼的表示能力、其与生俱来的高效的传播能力以及价格便宜等多项优点,所以那些具有生意头脑的商人自然不会视而不见,也因此加快了 WWW 的发展。其发展的速度远超过目前其他媒体的发展速度,其结果自然就是今天在 WWW 上可见的各式各样的应用了(很难想象这只是这一两年所发展出来的成果吧!)。正因为 WWW 是如此具有吸引力,所以许多脑筋动得快的人便抓住此机会推动 Java 的发展。虽然 WWW 目前已经有许多好处,但是对人们永无止境的需求而言仍嫌不足。因此,太阳电脑公司(Sun Microsystem)便推出了一个新的语言 Java,人们可利用它来开发更具威力及说服力的展示系统。

其实 Java 在被设计出来时,原本的名称并不是叫做 Java,而叫做 Oak,这个名字的由来是因为该设计小组的窗外恰有一棵橡树;但后来该小组发现 Oak 这个名字已被该公司的其他程序语言所使用,这使得该小组伤透脑筋,在开过数次的会议后,才决定以他们常去的那家咖啡厅的店名(Java)为该语言的名称,所以您常可以在 Java 相关的数据看到一杯冒着热气的咖啡。

当然 Java 并不是只能用于 WWW 上,就如其他语言一样它也可以独立于 WWW 之外来执行。其实就好象 Clipper 较适合用于设计一般数据库, Fortran 较适合用于设计科学运算, C 较适合于设计一些系统程序一样, Java 其实只是一个较适合于 WWW 上所使用的程序语言罢了。就如同可以拿 C 语言来设计数据库的软件一样,也可以拿 Java 来设计一些其他用途的程序,只是可能较不适合以及较麻烦而已。不过通过 Java 所提供的 API,我们仍然可以使用 Java 轻松的实现许多复杂的功能,只是速度不甚理想罢了。

Java 可以说是目前最热门的话题之一,主要是因为它可能对目前电脑界造成相当大的影响,且可能改变人类对电脑及电脑软件的看法。光从工程部门,经济决策部门以及经济部门急着跟 Sun 订立 Java 的合约便可看出个大概。由此看来 Java 确实可能造成电脑界另一次的革命。相信再过几年应该可能到处看到使用 Java 写好的软件放在网络上供使用者取用,且使用可以直接在解释器上执行 Java 机器码来执行 Java 的程序,如此以后,我们便可以不用去买软件回来安装,而只要在需要时去网络上抓取一个适当的软件执行即可。这有许多好处,首先可以省掉安装软件的手续,且在电脑中只需存放有用的数据和最基本的 OS,而不需存放一大堆可能很久才用一次的软件,因此对使用者而言,便可以节省许多不必要的花费,且厂商

也可以省掉将软件产品化的麻烦，更可以减少资源浪费及环保问题，可想而知，以后的电脑世界将会跟现在大不相同，且 Java 将扮演一个很重要的角色。我们希望本书可以帮助读者发展出另一个类似 Netscape 或 Yahoo! 的软件。

1.2 Java 的特色?

要了解一个语言最好先看看该语言到底有什么好处，能为我们做什么，且为什么非用该语言不可。所以本节要介绍一下 Java 所具有的特色，相信看完后将会加深读者学习 Java 的动力。

为什么要用 Java 呢？主要是因为 Java 具有以下的特色：

简单易学

Java 语言的基本架构其实是承袭 C++ 语言而来的，但它去除了 C++ 语言中一些较复杂及很少用到的功能。在 Java 中去除了原本 C++ 中之“运算符重载” (Operator Overloading)，“头文件”，“预处理”，“指针运算”，“结构”，“联合”，“多维数组”，“模板”，“数据类型转换”及“函数”等概念。既然有那么多的东西被去掉了，所以对于程序设计员而言自然是更容易使用了，他们只需要短时间就可熟悉其用法。对于没学过程序语言的人这也是一个好的开始，不过必须先了解面向对象程序的概念。

面向对象 (Object Oriented)

跟 C++ 一样，Java 也是一个面向对象程序语言，既然 Java 是面向对象的程序语言，那么 Java 自然也具有所有面向对象程序语言的好处了（容易维护，软件的可重用等）。至于面向对象的详细介绍则请参考第 3 章。

编译式语言

Java 的程序写好后必须利用 Java 的编译器将 Java 的源程序编译成中间码 (Java 的中间码称为 Bytecode)，然后才能执行，其实 Java 可以说是同时具备编译及解释两者特性于一身，因为其所编译出来的中间码在执行时还必须经过解释处理。所以写 Java 的程序时，只需要将程序编译完后再利用其解释器来执行程序。而使用传统的编译式语言编写程序时，则需要经过编译，连接及执行等过程。这个过程如果只执行一次那么便没有什么差别，原因是当我们在写程序时通常都需要经过该过程许多次执行、调试和修改程序等处理，所以使用 Java 来开发程序将可使这个过程变得简单。

由于目前 Java 的编译器及解释器的速度稍微慢了一点，所以倒也跟其他语言的开发过程所花的时间差不多，甚至更长，不过以后这个特色将会慢慢的呈现出来。

跨平台 (independent of platform)

因为 Java 程序经过编译后并不是产生可执行码而是产生一个统一规格的中间码 (Bytecode)。所以需要在每台机器上设置一个看得懂该中间码的虚拟机，使该机器能识别 Java 的中间码而可以执行该 Java 程序。所以使用 Java 编写的程序可以在不同的机器上被执行，这表示程序设计员再也不用去管所写的程序是要在什么机器上执行，因而可以减轻程序设计员的负担。

另外由于所产生出来的程序可以在不同的机器上被执行，这也意味着，会有更多的用户可以使用程序，所以针对同样功能的程序而言，使用 Java 所写出来的程序将会有更大的市场，对有生意眼光的人而言，这是一个非常有吸力的条件。

多线程 (Multithreaded)

一个线程 (thread) 可以被想象成为是一个小的副程序, 而多线程 (Multithread) 则是可以让程序同时执行多个不同的副程序。在 Java 中, 只要照着其所规定的方式来做, 就可以让程序轻易的使用到多线程的功能 (例如, 一边演示动画, 一边还可以播放声音)。

这不但可以轻易地实现一些使用其他语言很难做到的功能, 重要的是如果使用者的作业系统可以提供多线程功能或机器上支持多个 CPU 的话, 那么 Java 也可以用到那些先进的功能。

垃圾回收 (Garbage Collection)

在 Java 中并没有动态分配内存的功能 (如 malloc 及 free), 因此程序设计员并不需要去关心内存到底是如何被配置的。在 Java 中所有的内存是当该对象被用到时才会被配置所需要的内存, 而当该对象的工作完成后其使用的内存便会被自动收回。至于内存的管理则是由垃圾回收器 (Garbage Collector) 来负责, 其垃圾回收器则是被放在一个线程 (thread) 中执行。由此可见, 利用 Java 来编写程序, 程序员又少了一些工作了。

健壮性 (Robust)

由于在 Java 中, 许多程序的错误会在编译时被找出来, 而且因为没有指针的使用, 所以在 Java 中就不会发生因为指针指到不同类型数据或指错地方所造成的错误, 且大部分的错误皆会在编译时被找出来, 所以程序自然较少出现一些奇怪的错误。

而为了安全起见, 在程序执行时期, Java 也会做一些内存使用的检查 (如数组边界的检查) 以免存取到不该存取的内存, 而影响到其他程序的执行。而如果 Java 发现程序有错, 那么 Java 便会触发一个异常情态 (Exception), 好让程序可以控制那些特殊的情形。因此使用 Java 写出来的程序, 自然较少有错误也不容易崩溃。

安全性 (Security)

Java 基于安全性的考虑, 所以对 Applet 便有一些额外的限制, 以保护系统不被破坏。

图形用户界面

在 Java 中, 也提供许多基本类以方便程序设计员设计人机用户界面, 因此程序设计员可以利用所提供的功能快速的设计出图形用户界面。

网络传输功能

Java 也在其所提供的基本类中, 加入了一些网络传输的功能, 因此可以使用 Java 轻松的编写出具备一般网络传输功能的程序。

可扩充性 (Extensible)

使用 Java 编写程序除了可以使用其所提供的一些基本类外, 在网络上还可以找到许多针对特殊需求而设计出来的类以随时扩充程序的能力。因此使用 Java 来编写程序, 很多功能程序员可以不必从头进行编写。

动态 WWW 演示功能

当然, 最重要的功能就是 Java 可以让用户在 WWW 的 HomePage 上执行用 Java 所写出来的程序, 因此除了多了一些基于安全考虑所做的限制外, 几乎所有 Java 所提供的功能都可以用在 WWW 上。看了以上这么多的好处, 相信读者不难了解为什么要用 Java 吧!

1.3 Java 的应用

那么 Java 到底可以让我们做什么呢? 这里要介绍几个使用 Java 所做出来的应用程序, 它

们会帮助读者了解 Java 的能力。

例一：动画

可以使用 Java 写出播放动画的程序，下图即是 Sun 所提供的标准动画范例的部分图片。可以在 Java 的开发工具箱 (JDK) 所附的范例程序内找到它，其中存放许多张图片。当程序执行时，这些图片便一张张连续播放出来以达到动画的效果，在该范例中并搭配有声音。



图 1-1

由此例我们可以了解，Java 可以播放多媒体的数据，如简单的图片、声音、动画、影片等。

例二：时钟

也可以使用 Java 写出可动态改变且较有弹性的程序，下图所示的时钟即是一例，它可获取用户电脑的系统时间并即时显示出来，其中时钟的各指针也会跟着改变，在这个程序中它利用了 Java 所提供的函数来获取时间，并利用绘图的功能画出时钟的外框，时针，分针及秒针等。

由此例我们可以了解 Java 也可以让我们做一些简单的绘图功能。

例三：魔术方块

除了可以使用 Java 来写如上所示的较简单程序外，也可以使用 Java 来写一些较复杂的程序，如曾经非常流行的魔术方块也可以用 Java 写出来。在这个程序中，可以利用鼠标来旋转魔术方块的角度及转动魔术方块。

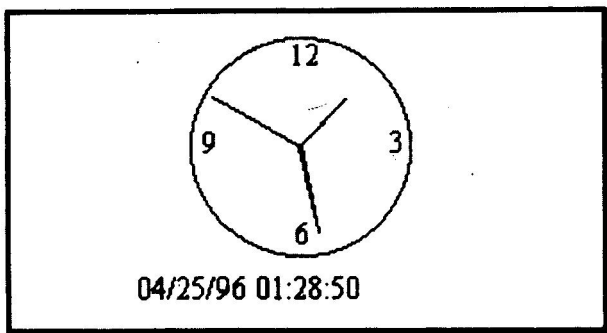


图 1-2

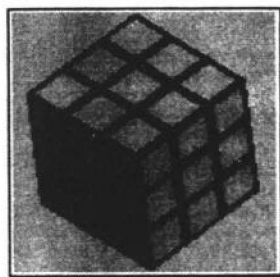


图 1-3

由此例可了解，Java 可以让我们用来编写游戏程序。其中应该也可以简单的看出，Java 跟一般程序一样也可以作一些复杂的数学运算。

例四：青蛙与苍蝇

这是一个很有趣的例子，其中有许多苍蝇会在青蛙前面飞来飞去，当苍蝇飞至青蛙的前面时，青蛙便会伸出其长长的舌头将可怜的苍蝇吞到肚子里去。

由此例我们可以了解使用 Java 可以设计出一个多媒体演示的程序。就如同一般多媒体演

示软件所能做的一样。

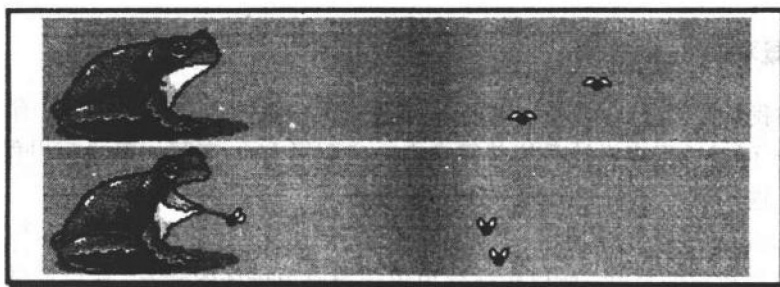


图 1-4

例五：统计图

也可以使用 Java 来画出统计图表，图 1-5 所示即是一例。在此例中，统计图的数据是可以动态改变的，其中每一秒都会有新的数据从 Server 端传过来，Client 便可根据所收到的数据随时更新统计图。

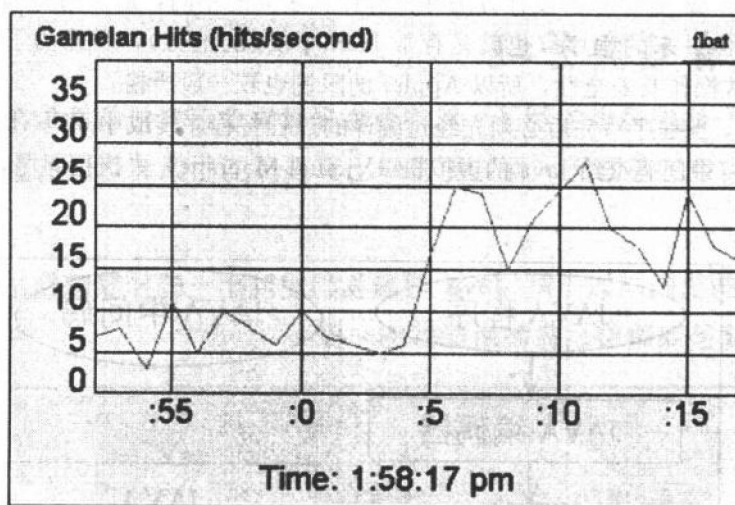


图 1-5

由此例我们可以了解 Java 可以让我们随时监看所想到的信息，如股票信息，或监看工厂的作业情形等，且可以支持网络传输的功能。

相信从上面的例子中读者应该可以大概了解 Java 可以做什么了吧！其实上面的例子只是 Java 所能做到的事情中的一小部分而已，如果读者对于 Java 的一些实例有兴趣的话，读者可以前往 <http://WWW.gamlan.com> 看看。在该站中有好几种不同类型的 Java 应用，共包含了几百个用 Java 写出来的例子及程序，并且它们还在不断地增加。因此可以上去看看目前 Java 到底有那些实际的应用，并可以从中找到一些觉得不错的例子，用它们装饰自己的主页 (Homepage) 或看看人家的主页是怎么写的。

也许会说“Java 的功能没什么特别的，它能做的我用其他语言也可以做到”。但请别忘了，

Java 有一项才能是其他语言所不具备的(至少目前是如此):就是它可以用在 WWW 上让主页变得更加生动、活泼、有趣。

1.4 Java 概观

在本节中我们将看看 Java 的动作方式,并给个简单的程序让读者对 Java 有更具体的了解。

首先 Java 可以让程序设计员依其需求来决定要用 Java 来写一般的应用程序或 WWW 上的应用程序。这两种不同用途的程序分别介绍如下:

- Application: 这类似于使用一般程序语言所产生出来的程序,由于其产生出来的程序具有可单独执行的特性,所以较适合用于一般应用程序的开发,所以叫做应用程序,Java 的应用程序可以用到 Java 所提供的全部功能。
- Applet: 这是一般传统程序所没有的,程序员可写出一个可在 WWW 上动态执行的程序,这种程序就叫做 Applet。跟 Application 不同的是 Applet 并不能单独被执行,要执行 Applet,必须先写一个包含该 Applet 的 HTML 文件,然后再通过可看 Java 的浏览器(Browser)来执行,如 Netscape 等。另外,由于安全性的考虑,所以使用 Java 的 Applet 来写程序有许多限制,如 IO 的限制,网络存取的限制等。

Application 或 Applet 的设计方法并没有什么多大的不同,最大的差别在于 Applet 通常是由 Server 端传过来的程序(也就是存放 Homepage 的地方),因此为了保护 Server 端及 Client 端的正常操作及安全性,所以 Applet 的限制也较一般严格。

在 Java 中,当程序写完后必须先经过编译的过程将程序转成中间码(Bytecode),当程序要执行时,再将中间码交给 Java 的虚拟机(Virtual Machine)去执行即可,下图便是一个简单的说明:

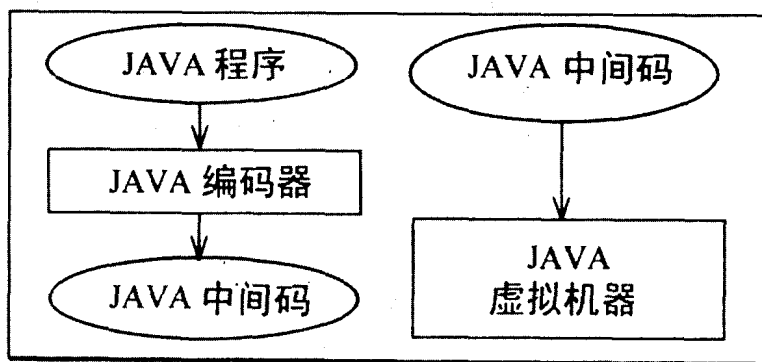


图 1-6

其中图中左边部分是当写完 Java 程序后,必须经由 Java 编译器将程序转成中间码,而要执行 Java 的程序时,则只要将所产生的中间码交给 Java 的虚拟机器即可。由上图我们也可以了解,程序只要转成中间码后,当执行程序时便不用再经过 Java 的编译器转换,而只要直接用 Java 的虚拟机器来执行即可。

因此只要我们产生一份 Java 的中间码后,不仅可以执行该程序许多次,更可以借助传送该程序的处理而达到多个机器执行同一个程序的目的,这也是 Applet 的工作原理。下图便是一个简单的说明,注意图中每个虚拟机器的颜色都不同这表示它们可以是不同的平台(如 PC,

MAC, Sun workstation 等)。

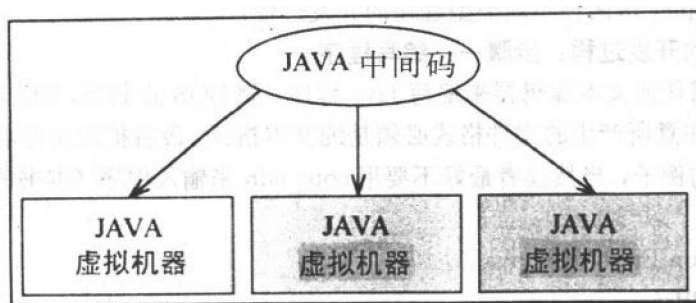


图 1-7

知道了 Java 的基本工作原理后, 接下来再来看看开发一个 Java 程序的流程, 也就是如何从头开始至产生一个可以执行的程序。

要开发 Java 的程序首先必须要有 Java 的开发工具箱 (Java Developers Kit), Java 的开发工具箱叫做 JDK。JDK 是太阳微电脑公司发展出来供程序设计员开发 Java 程序用的, 其中主要包括了 Java 的编译器, Java 的解释器, 程序调试工具及一些基本的类库 (就好象函数库) 等。关于 JDK 的详细介绍, 在本书的第 2 章中将会有进一步的讨论, 本节只针对一些在本节中使用的工具做简单的说明。

有了程序开发工具, 接下来应如何着手呢? 首先让我们来看一下 Java 的程序开发流程, 如下图所示:

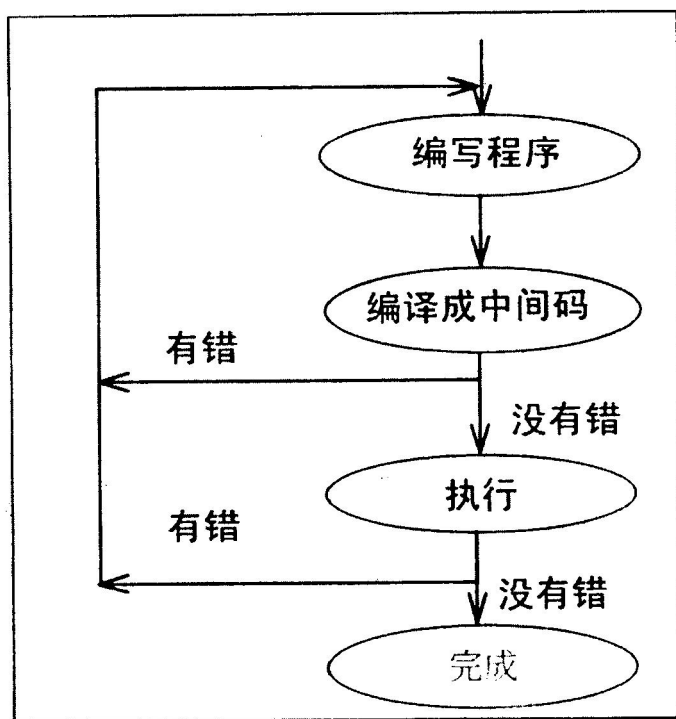


图 1-8

其中主要包含了编写程序，编译成中间码及执行三个步骤，下面就依此三个步骤来说明 Application 及 Applet 这两种不同类型程序的开发过程：

Application 的开发过程：步骤一：编写程序

首先可以使用任何文本编辑器来编写 Java 程序（如 DOS 的 PE2, EDIT, 或 windows 的记事本等），只要注意所产生的文件格式必须是纯文本格式，最后扩展文件名必须为 .java。下面是一个最简单的例子，当然读者最好不要用 copy con 来输入程序（本书的程序发展平台是 Windows 95）：

```
c: \>copy con FirstAPP.java
class FirstAPP {
    public static void main (String args [])
    {
        System.out.println ( "This is my First JAVA Application");
    }
}
^ Z
1 file (s) copied
```

```
c: \>
```

步骤二：编译成中间码

要将 Java 的程序编译成中间码只要使用 JDK 所提供的 Java 编译器（javac）即可，如下例：c: \>javac FirstApp.java

```
c: \>
```

如果编译的过程有错，则 Java 的编译器会打印出错误的信息，其中包括错误的行号及错误的种类，然后需要回到步骤一中看看程序那里有错并将错误改正过来，然后再重复步骤二。如此一直做到编译结果正确为止，在这里要注意的是，所给的程序名称必须是长文件名格式，否则 Java 的编译程序将会认不出来。

步骤三：执行程序

当编译的结果正确后，会产生一个扩展文件名为 .class 的文件，此时如要执行该程序只使用 JDK 所提供的解释器（java）即可，执行方式如下：

```
c: \>java FirstAPP
This is my First JAVA Application!
c: \>
```

由上例可看到程序的执行结果，如果执行结果与所要的有出入的话，则可回到步骤一中对程序做适当的修改后再重复步骤二及步骤三，直到正确为止。

Applet 的开发过程：步骤一：编写程序

首先也要使用文本编辑器来编写 Java Applet。与 Application 不同的是 Applet 不能单独存在而必须与 HTML 文件一起出现，所以也必须为 Applet 编写 HTML 文件，下例是一个简单的 Applet 的程序：

FirstApplet.html 的内容如下：

```
<center>
<h1>这是我的第一个 APPLLET</h1>
```

```
</center>
<APPLET CODE= "FirstAPPLET.class" WIDTH=200 HEIGHT=30>
</APPLET>
```

FirstApplet.java 的内容如下:

```
import java.awt Graphics;
public class FirstAPPLET extends java.applet.Applet {
    public void init () {
        resize (200, 30);
    }
    public void paint (Graphics g) {
        g.drawString ( "This is my First APPLET!", 40, 20);
    }
}
```

步骤二: 编译成中间码

要将 Java 的 Applet 编译成中间码一样是使用 JDK 所提供的 Java 编译器 (javac), 如下例:

```
c: \>javac FirstAPPLET.java
c: \>
```

当编译的结果正确后, 会产生一个扩展文件名为 .class 的文件。

步骤三: 执行程序

要执行该程序则可使用 JDK 所提供的 appletviewer 或任何可以观看 Java 程序的浏览器 (例如: Netscape 2.0) 调出第一步骤中所写的 HTML 文件即可, 如下即是使用 appletviewer 的例子:

```
c: \>appletviewer FirstAPPLET.html
thread applet-FirstAPPLET.class find class FirstAPPLET
Opening stream to: file: /C: /FirstAPPLET.class to get FirstAPPLET
其执行结果下图所示:
```

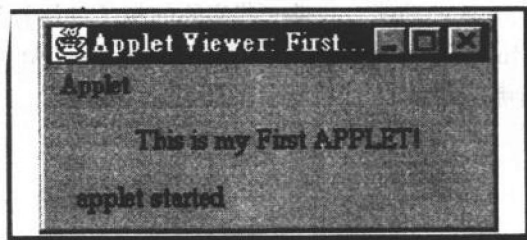


图 1-9

下面则是使用 netscape 观看 FirstApplet.html 的结果:

由图 1-9 及图 1-10 中, 可以看到使用 JDK 所提供的 appletviewer 只能看 HTML 文件中的 Applet, 且对每个 Applet 都会产生一个独立的视窗来执行, 而使用 netscape, 是可以在一

个画面中看到 HTML 文件中的所有内容，且可以同时看到多个 Applet。



图 1-10

1.5 如何使用本书

经过本章的介绍后，读者应该已经对 Java 有了基本的认识。那么要如何使用这本书呢？读者可以依照自己的需要依序看完本书的每一章节或直接跳至有兴趣的章节，下面就大概介绍一下本书的编排方式。如果读者是初学者，那么我们建议从第 2 章的 Java 开发环境的建立开始看起，并建立一个可以使用的程序开发环境。建立好程序开发环境后，便可以开始写程序了。要写程序当然得先了解如何写程序，所以在第 3 章中我们便介绍 Java 语言的基本结构也是 Java 的精髓部分，即面向对象。在该章中我们介绍了面向对象的一些基本概念及如何设计出一个面向对象的程序。然而这样仍是不够的，要真正将程序写出来，就必须要了解 Java 的语法，所以我们在第 4 章中介绍 Java 的语法。当将第 4 章以前的内容看完后，其实便可以开始写程序了，不过由于 Java 还有提供许多好用的 API，所以在本书后半部分的主要目标便是介绍 API 中较常用到及较具特色的功能，并举一些实际的例子供读者参考，尽量让读者能清楚其用法。这些章节包括 Multithread, Graphics User Interface, Applet, IO 及网络。读者可以视自己的需要来决定看哪一章。

第2章 程序设计环境的建立

2.1 如何安装 JDK

要设计 Java 的程序，必须先获得 Sun 公司所提供的 Java Developer's Kit (JDK)，其中包括了 Java 的编译器及解释器等开发程序时所需要的开发工具。除此以外，在 JDK 中还包括了可供程序设计员使用的一些标准 package 及范例程序。

截至本书截稿为止，最新的 JDK 版本为 1996 年 5 月所推出的 JDK 1.0.2 版本，基本上这个版本的功能与 JDK 1.0 及 JDK 1.0.1 差不多，只是更正了一些错误而已，目前 JDK 有 SPARC/Solaris, x86/Solaris, MS Windows 95/NT, 及 Mac OS 等版本，不过在本书中，笔者是以最普遍的 Windows 95 平台为开发平台，所以下面的介绍便以如安装 Windows 95 的程序开发环境为主，其他的工作平台的设置方式都差不多。

要获取 JDK，读者可以直接访问 Sun 公司 (java.sun.com)。或者也可以到国内的一些 ftp 站下载，下面就是存放 java.JDK 的一些站址：

ftp: //ftp.cis.nctu.edu.tw/pub/WWW/Java/JDK-1_0_2-win32-x86.exe

ftp: //gate.sinica.edu.tw/language/java/pub/JDK-1_0_2-win32-x86.exe

ftp: //ftp.vit.edu.tw/pc/win95/JDK-1_0_2-win32-x86.exe

这个程序约有 4MBytes，所以可能要下载一段时间，下载以后，读者只要直接执行此程序，此程序便会将整个 JDK 的内容解开，并在执行该程序时所在的目录下建立一个 java 的子目录，在该子目录中，有 4 个文件及 4 个子目录，说明如下：

- COPYRIGHT: JDK 的版权说明。
- index.html: JDK 的 html 的说明文件，可以直接通过此文件找到版权说明、README 文件、及 JDK 提供的所有范例程序。
- README: 里面有 JDK 内容的介绍，并列出了编写 Applet 时可用的 HTML 语法。
- src.zip: 这个文件内包含 Java 所提供之基本 package 的源程序代码，读者可以利用它了解其内部的实现方式。
- bin: Java 所提供的编译程序，直译程序及所有工具程序都放在这个目录中，所以必须在 path 中加上这个路径名称，以使用里面所提供的工具。里面所提供的工具及其功能如下所述：
 - ◆ appletviewer: 一个可以执行 Java applet 程序的工具。
 - ◆ Java: 执行 java Application 程序的工具。
 - ◆ javac: Java 的编译程序。
 - ◆ jdb: Java 的调试工具
 - ◆ javah: 一个可以分析 Java 的 class 文件并产生出可与 C 连接的头部文件及 C 程序的工具。
 - ◆ javap: Java class 文件的反编译程序。