

• 精华 • 简明 • 实用系列丛书
Made Simple Books

北京科海培训中心

JAVA

简明教程



(英) Peter McBride 著
钟向群 译

机械工业出版社

MADE SIMPLE
BOOKS

北京科海培训中心

· 精华 · 简明 · 实用系列丛书
Made Simple Books

JAVA 简明教程

(英) Peter McBride 著

钟向群 译

机械工业出版社

内 容 提 要

JAVA 是面向对象程序设计语言中的极品,本书在不大的篇幅中,以大量的程序实例,精辟的讲解,将一个博大的语言简洁地一一展现出来。

本书共分 9 章,分别讲述了 Java 基础、变量与值、程序流、Applet 基础、用户接口、线程以及图形、图形与声音的处理,最后对各章习题进行了解答。

本书实用性强,是 Java 程序员一本较好的入门书,也是选修 Java 课程的大专院校学生学习的好教材。

图书在版编目(CIP)数据

JAVA 简明教程/(英) 麦克布赖德 (McBride, P.) 著;

钟向群译. —北京:机械工业出版社,1998. 12

(计算机精华·简明·实用系列丛书)

书名原文:JAVA Made Simple

ISBN 7-111-06787-8

I .J… I. ①麦… ②钟… III. JAVA 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(98)第 35140 号

出版人:马九荣 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:科培 责任校对:成昊

门头沟胶印厂印刷·新华书店北京发行所发行

1998 年 12 月第 1 版·1998 年 12 月第 1 次印刷

787mm×1092mm 1/16·12.875 印张·251 千字

0001—5000 册

定 价:19.00 元

丛书序

在计算机新技术迅猛发展、新知识应接不暇、新软件层出不穷的今天,对学会操作电脑的人需要拓宽使用面,让电脑发挥真正的作用。对有经验的用户,跟上潮流的发展而不落伍,也需要不断地更新自己的知识。

我们热情慎重地向广大读者推荐这套布局谋篇上独具匠心、内容精辟、讲叙简明而又实用的系列丛书;这套丛书取自英国非常畅销的“Made Simple”系列中的一部分,透过此系列让人感受到原作者的写作水平,对要介绍的软件、语言和系统的深刻理解,以及作者群体学术的严谨和扎实。“Made Simple”顾名思义使问题简单化,也就是一种将厚书写薄的丛书。每一本书抓住重点,将讲述的对象介绍得简明扼要,通俗易懂。无论是介绍操作系统,还是介绍编程语言或是开发环境,都坚定不移地遵循了这一原则。

- **书不在厚而在于精。**

此系列,每本小册子不足 200 页,每本书讲解一种软件或是一门编程语言,内容相对独立,使读者能迅速定位自己的需求。每个问题辅以三两实例、言简意赅、点到为止。最为可贵的是作者不搞“大而全”,而是直书精要之处,将基本概念、难点、常用方法及相关技巧一一展示给读者。

- **语言简朴,引导有方。**

本套丛书是很好的教材,特别是针对初学者,尤为难得。对关键概念,作者舍得花笔墨,用通俗的语言加以阐释;枝节之处,则当删则删,当漏则漏;而且全书都是用简明的图示来表达要点,让读者学得轻松、容易树立信心。

- **实用性(这是许多书称有而最不容易达到的)。**

此系列丛书的实用性称得上扎扎实实。全书以问题、任务为主线,辅以大量实例而构成。这些例子实用性强,并且这些例子并不单一,往往例子彼此相关,最后可能组成的是一个比较大的程序,或是一个复合技术。这与我们常见到的一些书,往往一个例子表达一个简单的功能,彼此无关,无助于读者构筑自己较复杂的应用程序。此系列对例子的解释也是用图示表达,其中读者可自行修改、替换。

此系列丛书大部分每章末均有习题,书后有习题答案,这些习题也很有特点,它不是简单地复述前面的概念,也不是前面例子的翻版,而比例子更具有创造性、思考和提高的余地和价值。这是很难得的,也是一本好教材的内涵所在。

我们在翻译此系列丛书时,尽可能地聘请有经验、高水平的译者,目的是为了保持原丛书鲜明的风格。翻译了其中 JAVA 和 UNIX 两本书的钟向群先生认为:“此套丛书很像一个隽永精品集,读者极易理解,却又回味无穷,科技书籍中有此效果者寥寥”,翻译 Visual Basic 和 Visual C++ 两书的熊桂喜教授认为:“这是一套非常难得的轻松型教材,值得推荐与学习”。

在当今电脑书籍让人眼花缭乱,汗牛充栋的现状下,指导读者发现和正确选择一些好的读本是我们的义务,也是我们的责任,为读者编写诸如此类的教材是我们工作的方向。

希望此系列丛书帮助你开启电脑知识和程序设计的大门,相信读者是好书真正的评判者。欢迎来函来电联系与指正。

欢迎选购:

《C++ 简明教程》

《VISUAL BASIC 简明教程》

《VISUAL C++ 简明教程》

《JAVA 简明教程》

《PASCAL 简明教程》

《DELPHI 简明教程》

《UNIX 简明教程》

《Windows NT 简明教程》

《多媒体简明教程》

《硬盘管理简明教程》

科海丛书编译委员会

1998 年 10 月

前 言

Java 是一门新的程序设计语言,主要用于网络,特别是 Internet 环境中。同样的 Java 代码无需做任何改变,就可以下载到运行 UNIX 的 Sun 工作站上或是运行 DOS, Windows 的 PC 机以及运行 Maca 的苹果电脑上,并且能正常运行。虽然目前主要用于设计 Applet(一种小应用程序)来创建生动的 Web 页面,但也开始用于编写完整的应用程序了。

Java 语言以 C++ 为基础,在风格和结构上很相似。Java 是相当庞大而丰富的语言,熟练掌握其所有特性是需要花很多时间的——毕竟有那么多需要学习的东西。但是,其整个结构却是脉络清晰,模式简洁,因此,一旦掌握了其精髓,就很容易触类旁通、举一反三。

本书并不是为那些熟练的 C++ 程序员准备的。即使他也想转到 Java 上来——事实上为他们服务的书已足够多了。本书的对象是那些没有或是很少有编程经验的读者,因此它是从基本的东西讲起。在基本概念和语言运用的实践上,本书花了许多笔墨,以避免一开始就纠缠于技术细节。越简单,往往越实用,越单纯,往往越能抓住本质。——但愿您能接受并喜欢这种风格。

本书先讲述一些 Applet 的要点,以便很快能创建生动的 Web 页面,在此基础上,我们便有足够的信心去深入学习语言本身,从而开发大的应用程序。每种功能却通过一个 Applet 或是一个短程序展示出来,一个新概念在实际例子中往往更容易理解和学会。在每章的后面均有一些练习题——通过它们来检测自己是否掌握了所学知识,并从中领会一些新的思想。大程序的源程序文本文件以及所有练习的答案,可以在 Butterworth - Heinemann 站点的“MadeSimple Programming Webpages”中找到:

<http://www.bh.com>

愿您开卷有益。记住当您疲惫头胀时,不妨来点咖啡,因为,那里有 Java 的身影,Java 之名原来自于咖啡!

P. K. McBride. July

1997 年 7 月

目 录

第 1 章 Java 概述	(1)
1.1 什么是 Java	(1)
1.1.1 Java 能干什么	(1)
1.1.2 需要学些什么	(2)
1.2 程序设计语言	(2)
1.3 Java 虚拟机	(4)
1.3.1 JVM 与浏览器	(5)
1.4 面向对象程序设计	(5)
1.4.1 传统语言与 OO 语言	(5)
1.4.2 对象的性质	(6)
1.5 对象和类	(7)
1.5.1 继承性	(8)
1.5.2 多态性	(9)
1.6 Java 开发工具	(9)
1.6.1 工具安装	(9)
1.6.2 工具的组成	(10)
1.7 准备工作	(13)
1.8 第一个应用程序	(14)
1.8.1 用什么样的字处理器	(17)
1.9 第一个 Applet 小程序	(18)
1.9.1 源代码	(18)
1.9.2 HTML 文档	(19)
1.10 浏览 Applet	(20)
1.10.1 AppletViewer(Applet 浏览器)	(21)
1.10.2 使用普通浏览器	(22)
1.11 开发 Java	(22)
1.11.1 索引	(26)
1.12 练习	(27)

第 2 章 变量和值	(28)
2.1 数据类型	(28)
2.1.1 基本数据类型	(28)
2.1.2 参考数据类型	(29)
2.2 创建变量	(30)
2.2.1 声明	(30)
2.2.2 变量名	(30)
2.2.3 赋值	(31)
2.2.4 常见错误:NO.1	(33)
2.3 输出	(33)
2.3.1 System.out.print	(34)
2.3.2 混合输出	(34)
2.4 输入	(36)
2.4.1 System.in.read()	(36)
2.4.2 类型转换	(36)
2.4.3 异常	(36)
2.5 运算	(37)
2.5.1 增量符与减量符	(40)
2.5.2 常见错误:NO.2	(42)
2.6 赋值运算符	(42)
2.7 运算符优先级	(43)
2.8 数组	(44)
2.8.1 数组初始化	(44)
2.9 String 对象	(46)
2.9.1 使用字符串	(46)
2.9.2 String[] args	(47)
2.10 StringBuffer 对象	(47)
2.10.1 从键盘输入字符串	(49)
2.11 变量的作用域	(50)
2.11.1 Java 程序的形态	(50)
2.11.2 作用域	(50)
2.12 练习	(52)

第 3 章 程序流	(53)
3.1 测试条件值	(53)
3.1.1 关系运算符	(53)
3.1.2 逻辑运算符	(53)
3.1.3 AND 和 OR(与和或)	(54)
3.2 for 循环语句	(54)
3.2.1 代码块	(56)
3.2.2 循环条件的变化	(56)
3.2.3 循环嵌套	(57)
3.2.4 补注:随机数	(58)
3.3 while 循环语句	(58)
3.3.1 do...while	(59)
3.4 if 分支语句	(60)
3.5 多分支结构	(63)
3.6 Switch 语句	(65)
3.7 方法的使用	(67)
3.7.1 参数	(69)
3.7.2 返回值	(70)
3.8 错误与异常	(71)
3.8.1 运行错	(71)
3.8.2 异常	(72)
3.9 练习	(73)
第 4 章 Applet 基础	(74)
4.1 Applet 方法	(74)
4.2 AppletViewer	(76)
4.3 在浏览器中浏览 Applet	(78)
4.4 文本显示	(79)
4.5 字体(Font)	(81)
4.6 颜色	(84)
4.6.1 Color 对象与 setColor()	(84)
4.7 鼠标控制	(85)
4.8 新方法	(88)

4.9 Applet 参数	(90)
4.9.1 计划	(90)
4.9.2 <PARAM...>	(91)
4.9.3 getParameter()	(91)
4.10 练习	(94)
第5章 用户界面	(95)
5.1 GUI 及其组成元素	(95)
5.1.1 Button	(95)
5.1.2 Canvas	(95)
5.1.3 Checkbox 和 CheckboxGroup	(96)
5.1.4 Choice	(96)
5.1.5 Label	(96)
5.1.6 List	(96)
5.1.7 TextField	(96)
5.1.8 TextArea	(96)
5.1.9 Layout 与 Panel	(96)
5.2 按钮和事件	(97)
5.2.1 事件	(97)
5.3 事件目标	(99)
5.4 Checkbox 和 CheckboxGroup	(101)
5.4.1 创建 Checkbox	(102)
5.4.2 Checkbox 的选取	(102)
5.5 文本元素	(105)
5.5.1 文本的操作	(106)
5.6 数字的处理	(109)
5.7 Choice 和 List	(111)
5.7.1 addItem()	(111)
5.7.2 索引	(111)
5.8 布局	(115)
5.8.1 BorderLayout()	(115)
5.8.2 GridLayout()	(117)
5.9 面板	(119)
5.10 练习	(122)

第 6 章 图形	(124)
6.1 线	(124)
6.1.1 drawLine()	(124)
6.2 矩形	(126)
6.3 椭圆	(127)
6.4 多边形	(128)
6.5 弧	(130)
6.6 饼图	(131)
6.6.1 数组	(131)
6.6.2 预定义颜色	(132)
6.7 画布上绘图	(133)
6.8 练习	(139)
第 7 章 图像、声音和文字	(140)
7.1 图像文件	(140)
7.2 图像的使用	(140)
7.2.1 URL	(141)
7.2.2 图像显示	(141)
7.2.3 简单的动画	(143)
7.3 图像过滤	(144)
7.3.1 接口	(144)
7.3.2 类与方法	(144)
7.3.3 正式开始之前... ..	(145)
7.4 声音	(147)
7.5 声音控制	(150)
7.5.1 检查文件	(150)
7.5.2 启动与关闭	(151)
7.6 文字布局	(152)
7.7 字串与数组	(154)
7.8 练习	(156)
第 8 章 线程	(158)
8.1 线程	(158)

8.1.1 带线程的 applet	(158)
8.2 活动图像	(161)
8.3 下载控制	(164)
8.3.1 MediaTracker	(164)
8.3.2 线程与 loaded 变量	(165)
8.4 练习	(168)
第 9 章 习题答案	(169)
9.1 第 1 章答案	(169)
9.2 第 2 章答案	(170)
9.3 第 3 章答案	(172)
9.4 第 4 章答案	(174)
9.5 第 5 章答案	(176)
9.6 第 6 章答案	(180)
9.7 第 7 章答案	(182)
9.8 第 8 章答案	(186)

第1章 Java 概述

1.1 什么是 Java

Java 是一种与平台无关、面向对象、多线程而且可以动态链接的程序设计语言——这里面有许多的行话。我会尽可能地少用术语,但有时也很难避免,不过我会努力回答好“什么是 Java?”这个问题的。

Java 是 Sun Microsystems 公司开发的一种新的程序设计语言,主要用于万维网(WWW; World Wide Web)。它和 C++ 有许多共同点,如果你能掌握 C++,那么理解 Java 便是轻车熟路了。本书假定你对 C++ 以及其他程序设计语言并不清楚,当然对 Windows 和 Web 浏览器最好还是熟悉的。

1.1.1 Java 能干什么

用 Java 主要可以编制三类程序——Applet(小程序)、Application(应用程序)以及 bean(功能模块)。

Applet 是一种嵌在 HTML 文档中的程序,用以形成 Web 页面。一般来讲,它们都比较小,因此用浏览器下载时不必等待太长的时间。但其用途一般比较局限,可能是导航 Web 站点的一种源亮方式,或是某个吸引人们注意力的动画,也可能只是提供娱乐以鼓励浏览者定期返回。不过,Applet 也可以做一些更严肃的工作。普通的 Web 页显示一般的静态页面是很好的,但缺少交互与反馈的机制。而 Applet 则可以将一些在线的交互机制加入到 Web 页中。比如,保险业就可以用 Applet 来计算瞬时报价。

Application 是一种独立的程序,你可以用 Java 编写一个字处理软件、一个电子表格处理软件,也可编写图形工具,事实上可以是任何应用。Java 的一个重要设计思想就是能用于创建网络计算机上的应用程序。与在网络的每个 PC 机上安装标准软件相反,Java 应用程序可以安装于中心服务器上,而在用户使用时才将它下载到网络计算机上运行。而且这种模式使得软件升级简单而便宜,因为每个应用程序对于整个办公室来讲只需安装一次。使用 Java 的网络计算机尚处于其早期阶段。这些应用程序是模块化的,这与以前的 30MB 的“全活”(各种功能全部集中起来)字处理软件不同,基于 Java 的字处理软件,由许多小的编辑模块组成,对拼写检查、布局、打印等都分别有相应的模块,因而其下载速度也相当快。这种方式还使得更小的、更廉价的电脑也可以胜任工作。由于各个模块在不

同的应用中可重用,这就大大简化了软件的生产。比如,打印模块既可以用于字处理软件又可以用于电子表格处理软件来打印文档。

Bean 是对 Java 软件包(package)的补充。它们使模块化做得更彻底。其基本概念就是 Bean 是一个可执行特定的、有限功能的单位,而且相互可以调用,也可以被 Applet 和 Application 调用。

1.1.2 需要学些什么

在学习 Java 程序设计之前,你需要理解:

- 程序设计的一些基本的、核心的概念以及特定的面向对象程序设计的概念(下面几节将讲述);
- 如何设置一个运作 Java 的系统(本章后面将讲到);
- 应用程序及 Applet 的基本结构(第 2 章);
- 如何在程序中存储数据(第 3 章);
- 如何让程序重复执行动作或是在特定条件下执行(第 4 章);
- 程序中使用 Java 例程和结构的技术——本书其余部分就是通过一个大系统中一部分程序的开发来展开的。

注意:

本书集中在 Applet 上,你很快就会有所收获,而且会觉得很有趣。

1.2 程序设计语言

计算机其实只能真正懂一种语言,这就是机器语言(机器代码)。这种语言由(二进制)数字组成,芯片将这些数字翻译成指令、数值及内存地址。指令很简单,比如:“将某个数值移到内存的某个地址中去”,“比较这些数字”,“某值增 1”等等。

人们曾经的确用机器代码写过程序,但现在已没有哪个正常人会再这样做了——因为这太辛苦了,人们已有更多的方法去编写程序。

大多数程序设计语言采用了人们易于理解(或多或少)的文字和结构,程序员所写的指令序列就像一个文本文件,这就是所谓的源代码,该文件是必须最终转化为机器代码的。这种转化到目前为止,有以下两种方法:

- 一种叫“编译”。源代码传递给编译程序,该程序先检查语法错误,如果发现错,停下来并显示错误清单。程序员必须将错误挑出来并改正,最后成为无错误的代码,编译程序才能进行下一步,也就是将源代码转化为可执行程序,也就是一堆机器代码,这些机器代码只能在合适的机器上运行,也就是在其编译的同类机器上才能运行。

如果能在不同平台上运行同一个程序,比如 Windows 的 PC 机,OS/2 电脑或 Macintosh 机器,从理论上讲,你可以在不同平台上将源代码重新编译。而在实践中源代码往往需要一定程度的重写,因为每个机器处理事情的方式各不相同。

- 源代码也可以解释执行。也就是说源代码在执行过程中由解释程序来处理。每行代码依次取出、查错。如果没有错误的话,然后转化为机器代码并执行。与编译程序相似,对于不同平台上解释执行程序,也往往要重写一部分源代码。

JavaScript 就是 Java 的解释执行版本,也是用于 Web 页的一种新语言。其代码嵌于 Web 页中,当浏览器显示页面内容时执行。如果想了解更多有关 JavaScript 的知识,可查看有关 JavaScript 的资料。

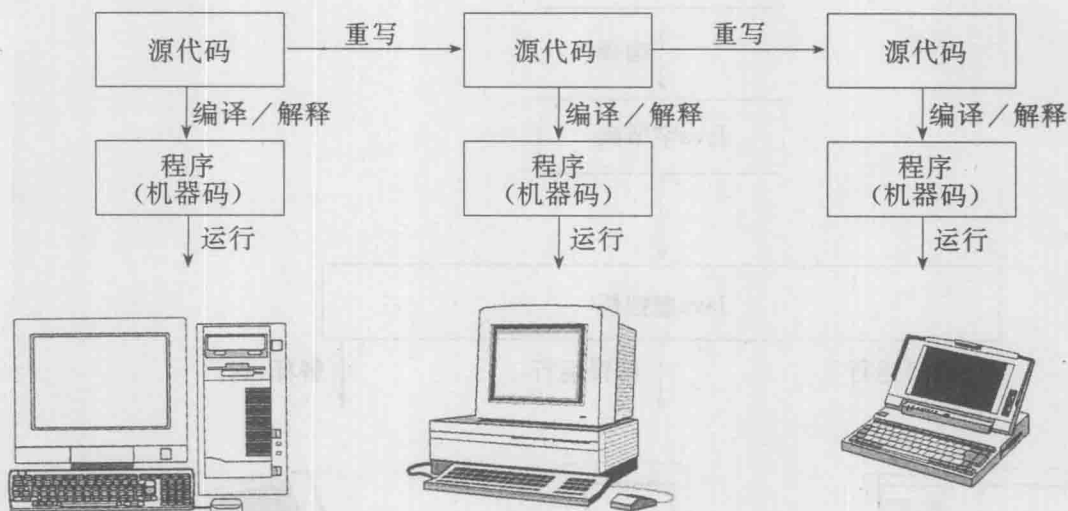


图 1-1

注意:

Java 程序可以同时有几个单独的例程在运行,这就叫多线程机制,能提供这种功能的程序设计语言不多。第 8 章将讨论有关问题。

1.3 Java 虚拟机

通过 Java, Sun 提供了解决在不同平台上运行同一程序的方案,这就是 Java 虚拟机(JVM: Java Virtual Machine)。将 Java 源代码传给 Java 编译器(编译程序)时,先转化为字节码(byte code),这不是机器码,而只是半成品。字节码传给任何安装了 JVM 的计算机时,字节码便被读取,并转化为机器码运行(见图 1-2)。

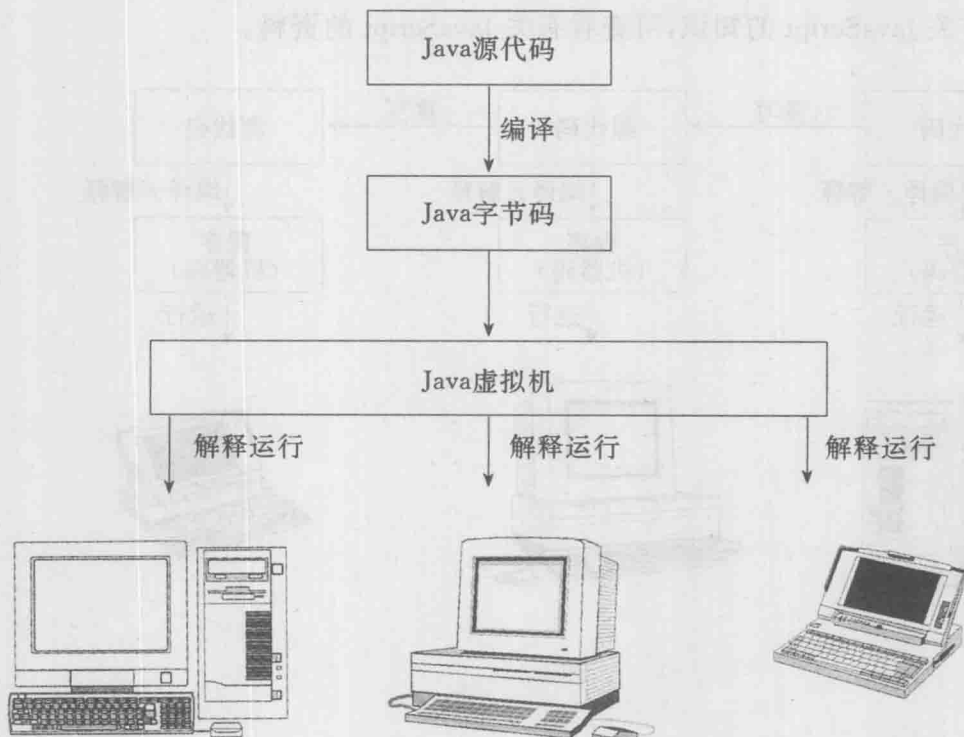


图 1-2

这个过程的主要优点是显而易见的。你可以将自己的 Java 程序公布在因特网(Internet)上,只要计算机有 JVM,便可以运行你的程序。如果 JVM 内嵌于 Netscape 和 Internet Explorer 中,则任何人都可以用这些浏览器来运行同样的 Java 程序了。

当然该过程的缺点也很明显,因为在执行前 JVM 需将字节码转化为机器码,因此程序的执行不会有编译程序那么快。

另一个缺点是,为特定类型的计算机写的程序,自然可以充分利用机器的性能和特性,优化其速度、增进其显示、利用其外设。而编写的在所有计算机上都能运行的程序当然只能使用那些共同能实现的功能特性。Sun 公司为突破此局限,专门为 Java 增添了一种功能,即处理“本地代码”(native code)的能力,这种代码便是专门为某类机器编写的。不过,此举似乎有点背道而驰了……

1.3.1 JVM 与浏览器

如果浏览器是 Netscape 2,3 或 4,或者是 Explorer 3,那么该浏览器就有了 Java 虚拟机功能,可以运行 Java 1.0 以下编译的 Applet 了。

如果浏览器是 Explorer 4,那么,你的机器也可以运行在 Java 1.1 下编译的 Applet——新的 Netscape 也能支持。

Java 虚拟机是浏览器软件的集成部分,无需安装。当然,要想真正能运行 Applet,应将浏览器中的“Enable Java”选项打开(见图 1-3~图 1-4)。

1.4 面向对象程序设计

Java 是一种面向对象(OO:Object Oriented)语言。但并不是第一个,它其实是植根于早期的面向对象语言中的,这种方法已成为程序设计的重要方式了。使用 OO 语言,一般来讲,你可以生成许多更好、更可靠的程序。

对于面向对象(OO)程序设计,人们从理论到实践已写了许多大部头的书。在此我只想花一两页略作介绍,但也足够窥其端倪、探其精髓了。当然后面在必要时还会经常提到这个重要概念。

1.4.1 传统语言与 OO 语言

我们先来看看“传统”程序设计语言与面向对象程序设计语言的区别。就以 Windows 屏幕上的按钮为例吧。在传统的程序设计语言中,为了创建一个按钮,先得编一段画按钮的程序,然后对鼠标的移动(鼠标光标)也要编程,而对单击动作的监控、单击效果也要编写程序,最后才是编写单击事件所要调用的程序。而