

THOMSON



TM

21世纪

高等院校计算机系列教材

Java

程序设计基础

Understanding Programming An Introduction Using Java

[美] Scott R · Cannon 著

Utah State University

李长青 谢君英 谢俊 译



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机系列教材

Java 程序设计基础

[美] Scott R.Cannon 著

Utah State University

李长青 谢君英 谢 俊 译

中国水利水电出版社

内 容 提 要

Java 语言是当前最为流行的编程语言之一。本书介绍了使用 Java 语言进行编程的各方面的知识。全书共分为三个部分,共 12 章。第一部分介绍了 Java 的基础知识,包括基本数据类型、简单的选择和重复、辅助方法以及问题分析和方法设计;第二部分介绍编写复杂应用程序所应具备的各项技能,如类的设计、简单数组、IO 流和文本处理、递归等;第三部分介绍了 Java 在 Internet 方面的应用,其中包括使用 Swing 和 Applet 进行编程。

本书由浅入深,样例丰富,可作为大学本科生学习 Java 语言的教材。对于 Java 编程感兴趣的初学者,本书也是一本很好的入门读物。

Scott R.Cannon, Understanding Programming An Introduction Using Java.

EISBN: 0-534-38933-3

Copyright © 2003 by Brooks/Cole, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All Rights reserved. 本书原版由汤姆森学习出版集团出版。版权所有,盗印必究。

China WaterPower Press is authorized by Thomson Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由汤姆森学习出版集团授权中国水利水电出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

北京市版权局著作权合同登记号:图字 01-2004-3013

图书在版编目(CIP)数据

Java 程序设计基础 / (美)坎农(Cannon, S. R.)著;李长青等译,一北京:中国水利水电出版社,2004.6

(21 世纪高等院校计算机系列教材)

ISBN 7-5084-2168-X

I. J… II. ①坎… ②李… III. 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(2004)第 053786 号

书 名	Java 程序设计基础
作 者	(美) Scott R.Cannon 著
译 者	李长青 谢君英 谢 俊 译
出版 发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 19.75 印张 447 千字
版 次	2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷
印 数	0001—5000 册
定 价	28.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

译者序

随着 Internet 的迅猛发展以及万维网 (WWW) 的日益普及, 与此相关的 Java 技术在短短的几年内以令人难以置信的速度向前发展。从最初应用在家用消费电子产品上的一个名为 Oak 的小型软件, Java 已经逐渐变成了 Internet 上倍受欢迎的首选编程语言。很多公司争相购买 Java 的使用权, 同时一些计算机公司纷纷推出 Java 的开发工具, 或在自己的产品中加入对 Java 技术的支持。随着 Java 技术的深入发展, 它在网络编程、便携式设备以及分布式系统开发等方面的潜力不断展现出来, 并吸引了全世界各类企业 & 个人的广泛关注。它的诞生给整个计算机行业带来了深远的影响, 已成为人们不可缺少的开发工具之一。

Java 是一种面向对象的编程语言, 它具有可移植性、健壮性、安全性以及并发性等诸多最先进的功能特性, 并在工业及其他各个行业中拥有广阔的使用前景。由于 Java 语言的很多优异特性, 它已发展成为首选的编程教学语言之一。首先, 它是一门实用性很强的语言, 既可以在您的本科阶段学习之用, 又支持在商业领域中的应用。其次, 它是一门功能强大的语言, 在现实世界中得到广泛的支持。另外, Java 支持模型和方法, 它们非常适用于越来越常见的大型复杂软件系统的开发。最近, Java 的重要性也日渐体现出来, Java 与 Internet、便携式设备以及分布式系统等紧密相关。

本书在编排和写作上, 没有过分强调 Java 技术的细节, 也不想成为权威的参考手册。而是从最基本的 Java 概念讲起, 从创建一个简单的 Java 程序开始, 循序渐进地讲述 Java 编程涉及到的各方面的内容。全书共 12 个章, 分为三个部分。第一部分向您介绍了 Java 的基础知识, 其中包括基本数据类型、简单的选择和重复、辅助方法以及问题分析和方法设计。在学习完这一部分后, 您将可以编写最基本的 Java 应用程序, 并可以对 Java 编程有更深一步的了解。第二部分也是本书最重要的一部分, 将向您介绍编写复杂应用程序所应具备的各项技能。这一部分涉及较多的 Java 内容, 其中包括各种复杂数据类型和控制结构、类的设计、简单数组、IO 流和文本处理以及递归等。在掌握了这部分内容之后, 您将可以编写出相对复杂的应用程序, 并能解决各种各样的编程问题。此部分是继续深入学习 Java 编程的基础, 读者应牢固掌握。最后一部分介绍了 Java 在 Internet 方面的应用, 其中包括使用 Swing 和 Applet 进行编程。在这一部分您将学会如何使用 Java 来创建基本的图形用户界面, 并生成简单的网页以及与此相关的技术。在学完本书所讲全部内容之后, 您将对 Java 有一个全面的认识, 具备编写 Java 应用程序的基本能力, 并为以后的深入学习打下了坚实的基础。

本书既适合于作为大学本科生的 Java 语言教材, 也可供 Java 初学者自学使用。本书的最大特色就是按课程教学的方式来组织内容, 因此适合于教师授课, 也适合于学生入门阅读。

本书由李长青、谢君英、谢俊翻译，王延华老师审校了全书。此外，谢小花和盛海燕参与了本书的代码录入和排版工作。在本书的翻译过程中，我们力求忠实原文，将原文内容以简洁、流畅的语言展现给广大读者。但由于译者的知识水平有限，加之时间仓促，书中难免存在不当之处，恳请广大读者批评指正。

译者

2004年3月

前 言

一、写给教师的话

在每个班上都会有聪明的学生，他们觉得 Java 编程相当自然直观。但是老师们必须注意，你们的任务并不仅仅是教会这些学生，同时还需要通过一种好的教学手段，来让其余的学生需要并理解一本教材。

本书并不打算成为权威的 Java 手册，而是将重点放在将 Java 作为首选语言进行教学，这将会涉及到两个原则：第一，当学生们第一次见到就立即觉得需要时，他们才能学得最好；第二，通过亲身的实践和体验学生们才能学得最好。

本书努力让学生尽早地写出有意义的应用程序来。实际上，学生们直到看到循环和条件之后，才会真正开始认为 Java 编程是一种非常有用的解决问题的工具。本书在最开始的几个章节中，将以广度优先法开始介绍，首先教会学生一种实现 I/O 的方式、一种变量类型、一种实现条件转移的方式和一种循环语句。很快地，学生们就能够用 Java 的这一子集来编写有用且有趣的程序了。在介绍下一个概念之前，不要求学生全面了解一个概念的方方面面——如果真要这么做，就会推迟有意义的体验。稍后，当学生们有了一些经验，并理解了各种可以实现的有用的应用程序之后，本书再返回来讲述其他的循环变种、其他条件语句、复合布尔表达式和其他数据类型等。由于这种广度优先的方法，直到学生们有了一个坚实的开端之后，本书才对一些具体的问题进行介绍，比如 if 和 if-else 语句的嵌套组合、混合模式表达式、异常、截断等。相应地，本书力求避免讲述那些会让学生们不能理解的错误的机制和概念。

正如所看到的那样，广度优先方法实质上是螺旋型的。最初将以简化方式来介绍一个概念或方法。逐步地，随着对更多功能需求的升级，该原始简化方式将逐步进行扩展，直至呈现出一个完整的概念。这也意味着本书将同时以螺旋型的方式来介绍多个主题。例如，在返回来介绍条件的更多细节之前，将会讲述简化的条件、简化的循环和函数。这样做的目的很简单：就是使学生尽可能早地编写出有意义的程序。然后这种编程的经验可以在介绍更多细节时起到促进作用，并有助于对它们的理解。

各个章节实际上是教学单元，而非主题单元。读者将会发现在一章之中并没有涵盖学习某个特定主题的全部内容，因为那种方式适用于参考手册，但并不适用于教学课本。学生们也许并不准备现解（或者需要）整个主题。如果尝试着将本书与传统方法相比较，就会发现本书中章与章之间的关系并不特别有章可循。例如，在其他课本中可能会很容易确定介绍循环的是哪一章，但是循环的概念和语法在本书中却涉及到了好几章。本书有一章专门介绍编程设计和问题解析，但是有关这些概念的其他资料也融入到了整本书中。随着程序复杂性的增加和学生在理解大型软件项目中需求方面能力的提高，风格、文档说明和

软件工程的方法也将整本书中随处可见。一般来说，概念将尽量建立在以前的例子和作业所体现出的需求之上。

Java 中一些更加高级的主题将以简介的形式进行介绍，为学生们以后更加详细地学习这些主题做好准备。甚至有一些高级主题根本就没有被提及，因为太多的语法和功能学习容易削弱良好的基础。

二、Java I/O

承前所述，在所有的编程语言中，Java 是惟一个“做任何事情都需要了解每一方面情况”的语言。这一点尤其体现在 I/O 上。传统方法教导学生直接使用标准的 Java 类库（比如 `PrintWriter` 和 `BufferedReader`）以及 AWT 或 Swing 包。但遗憾的是，这需要对某些概念（例如，异常和事件）有一个相当好的理解，而初学者尚未接触这些概念。因此，建立有意义的程序的时间要大大拖延了。在学生能够读入两个整数并输出它们的和之前，一些常见的 Java 教科书已经讲述了几百页的内容了。

本书提供了三个相当简单的类来支持磁盘和终端 I/O。这些类包装了刚才提到的 Java 类库，并且简化了所涉及到的语法和异常。IO 类（基于 Swing 中的 `JOptionPane` 类）提供给学生一种易于理解且易于使用的方法，以简化用于简单数值和字符串 I/O 的 GUI 窗口。`DiskInput` 和 `DiskOutput` 类为简单的数值和字符串磁盘文件访问提供支持。因此，学生们很快就能写出有意义的程序了。

这三个类很简单，也很直观。学生们在学完第 11 章后，就能够自己编写出这些类。同时，附录 A 中也给出了这些类的带注释的完整程序清单。

三、面向对象的编程

毫无疑问，Java 是一门面向对象的语言。要讲授 Java，教师就必须讲授对象、实例化、多态性和继承性。如果不能利用类、库、Bean、Applet、包以及其他方面的优点，那么 Java 就成了另外一门语言了。

但是遗憾的是，如果教材严格地从学术上的面向对象观点来讲授 Java，那么很快就会面临前面所讲述的“做任何事情都之前需要了解每一方面情况”的问题。在学生们编写第一个程序之前，教材很快就会陷入术语和概念的泥淖之中。在传统的 Java 教科书中，最初几章的教学曲线都十分陡峭。

本书将采用被认为是一种混合的方法。首先将静态类和“帮助”方法与以它们为基础编写的简单应用程序来进行讲授。当学生逐渐熟悉了方法和参数之后，本书再将动态类作为需要实例化的对象来进行介绍。最后，还增加了一个实例变量的主题。这种方法有助于促进对静态和动态之间差异的简单了解，而这一点对初学者来说往往是一个困难的主题。

四、Applet

有时候，一开始就使用 Applet 来讲授 Java。而在本书中，直到第 12 章才讲授 Applet。Applet 方法不仅使编写有意义的程序的时间延后，而且增加了学生们很难控制的语法和语

义。调试也会变得更加困难。学生们会开始过多地寄希望于“魔法词语”，而不是理解相关概念。

五、Java 2

本书基于 Java 1.2 版本，并使用相应的 Swing 库。Java 1.2（或更高版本）可以从 Sun Microsystems (www.java.sun.com) 下载。借助兼容 Java 1.2 的 Web 浏览器的支持，在讲授第 12 章 (Applet) 时即能达到理想的效果。如果学校没有配备上述 Web 浏览器，可以使用免费的 Opera（下载网址是 www.opera.com）。

六、一些教学特色

WRONG!

当演示一个错误方法时将会看到这个图标。

BUG!

这个图标通知你程序或代码段中包含错误。

- 语法框：语法以模型形式出现，这种形式更像是经验法则而不是规则。它可以支持螺旋式教学法——当为一个结构或主题讲授其他的语法时，无需替代以前给出的更为简易的形式，二者也不存在冲突。
- 例子：很少有学生会有足够的耐心去研究冗长而棘手的相关例子。然而，例子却是极为重要的，因为很少有学生具备足够的抽象思维能力，仅仅通过学习规则就能解决新出现的问题。在本书中尽量提供了许多简单易懂而又切题的例子程序和代码段。
- 通过实践学习：有一些练习需要实践和创新。每一章都包含一些实践框，其中包含一些简单的“如果这样会怎样”实践的建议，以讲授一个重要的主题。这些作业鼓励尝试和实践。
- 创造性挑战：每章都包括一些项目，而学生无法利用目前学过的知识来简单或完整地解决这些项目。这样做的目的就是表明对将要讲述的主题的需要，并鼓励自学。同时也鼓励创新和创造性的思考。

七、写给学生的话

本书的目标就是帮助学生尽早地编写出有意义的和有用的程序。因此，在某一章中讲述的简化模型和机制并不完全代表所描述的 Java 语言相关特性的所有功能。但是，它们将使學生有足够的语言能力来编写出更为复杂和更为有用的程序。随着你们的进步，这些简化模型将逐步扩展成为更多细节和更多功能。

在必要时程序例子是简单而完整的，而不必要时则只会出现代码段。重要的例子是与执行流程图在一起的，而执行流程图展示了执行过程中变量是如何变化的。常见编程错误的例子会用包含 bug 图形或“WRONG!”消息的路标来标出。

包含有用提示、重要概念或经验法则的段落均使用“概念”的旁注标出。段落中的重要词汇术语用斜体标出，并使用一个“关键术语”的旁注。同时，每章都有包含“如果这

样会怎样”实践的阴影框，要求你们对程序进行不同的测试并注意其相应结果。这也就是说，你们需要通过个人实践和研究来进行学习。

每章都可能有一组作业项目，同时这些项目需要用到本章所学习的工具。其中的某些作业还表明了学习下一章中所介绍工具的必要性，在每章结尾处都至少放置了一个有“创造性和挑战性”的项目。仅使用目前已经学过的相关知识来考虑这些问题，很可能得不出简单的解决方案。通常要通过创造性和革命性地应用已经学过的工具才能解决这些问题。对于有些项目，利用你们目前所掌握的技能也许还没有一个完整的解决方案，在这种情况下，需要尽可能地解决你们所能解决的部分，并指出或描述你们的解决方案的局限性。这些特殊问题的目的就是帮助学生理解并重视学习下一章中的工具和技术的必要性。

由于很多显著的原因，Java 经常被用作首选的编程教学语言。首先，它是一门实际的且常用的语言。它既可以在学位学习阶段学习，又支持在商业领域中的应用。同时它也是一门功能强大的语言，并在现实世界中得到广泛支持。另外，Java 语言支持模型和方法，而模型和方法对现实中越来越常见的大型复杂软件系统的开发工作来说非常重要。最后，但却有可能是最重要的，Java 与日渐重要的 Internet 便携式设备以及分布式系统等紧密相关。

祝你们好运！你们会发现计算机科学既充满挑战又富于趣味性。

八、Web 访问

下面这个 Web 站点是本书专用的，该站点提供所有可执行的示例程序、勘误表注释和其他有用资料的下载。

www.brookscole.com/compsci

如果发现了错误或印刷错误，请使用以下电子邮件地址来通知我：scott@cannon.cs.usu.edu。

九、致谢

我非常感谢以下各位编审：印第安纳州立大学的 Jim Ball、北维吉尼亚社区学院的 Edward Delean、伊利社区学院的 Catherine LaBerta、DeAnza 学院的 Mary Ann May-Pumphrey、南俄勒冈大学的 Kevin Sahr。

Scott Cannon

Logan, Utah

目 录

译者序

前言

第 1 章 第一个 Java 程序	1
1.1 用高级语言进行编程	1
1.2 一个简单的 Java 程序	4
1.3 字面输出	5
1.4 双精度型变量和声明	7
1.5 变量的输入和输出	8
1.6 赋值和表达式	9
1.7 类的调试	11
1.8 项目实例	15
1.9 小结	17
1.9.1 关键术语	17
1.9.2 概念	18
1.10 练习	19
1.10.1 简答题	19
1.10.2 项目	21
第 2 章 简单选择和重复	23
2.1 条件语句	23
2.2 语句块	26
2.3 while 语句	27
2.4 分析和设计简介	28
2.5 循环的使用	29
2.5.1 无限循环	29
2.5.2 确定循环	31
2.5.3 求和与计数	33
2.6 嵌套的条件和循环	36
2.7 项目实例	40
2.8 小结	42
2.8.1 关键术语	42
2.8.2 概念	43
2.9 练习	44

2.9.1 简答题	44
2.9.2 项目	46
第 3 章 辅助方法	48
3.1 模块化的程序设计思想	48
3.2 自顶向下的设计方法	49
3.3 具有返回值的方法	51
3.4 辅助方法的形参	53
3.5 变量的作用域	56
3.6 常量	57
3.7 项目实例	59
3.8 库类和辅助方法	62
3.8.1 普通的 Java 包	62
3.8.2 Math 类	63
3.8.3 私有库	64
3.8.4 专有库	65
3.9 小结	65
3.9.1 关键术语	65
3.9.2 概念	66
3.10 练习	67
3.10.1 简答题	67
3.10.2 项目	68
第 4 章 问题解析与方法设计	71
4.1 学习解决程序设计问题	71
4.2 有用的思考过程	71
4.3 基本组件	72
4.4 复合设计	73
4.4.1 循环设计	73
4.4.2 条件	75
4.5 自顶向下的设计	75
4.5.1 设计的开始阶段	76
4.5.2 范例	77
4.5.3 桌面测试	81
4.5.4 再访调试	82
4.6 软件的生命周期	83
4.7 小结	84
4.7.1 关键术语	84
4.7.2 概念	85
4.8 练习	85

第 5 章 更多的数据类型	88
5.1 浮点数	88
5.1.1 表示法	88
5.1.2 E 计数法	89
5.1.3 精度和范围	89
5.1.4 上溢和下溢	90
5.1.5 截取错误	90
5.2 整数	93
5.2.1 范围	93
5.2.2 混合型表达式	93
5.2.3 形参	95
5.3 更多的比较运算符	95
5.4 字符	97
5.4.1 表示法	99
5.4.2 字符类的 static 函数	100
5.5 字符串	102
5.6 布尔变量	105
5.7 项目实例	105
5.8 小结	107
5.8.1 关键术语	107
5.8.2 概念	108
5.9 练习	109
5.9.1 简答题	109
5.9.2 项目	110
第 6 章 更多的控制结构	112
6.1 单选 if	112
6.2 复合比较	114
6.3 for 循环	116
6.4 运算符的简写形式	122
6.5 多重选择及 switch 语句	125
6.6 do-while 循环	127
6.7 局部变量块	128
6.8 项目实例	129
6.9 小结	131
6.9.1 关键术语	131
6.9.2 概念	132
6.10 练习	133
6.10.1 简答题	133

6.10.2	项目	135
第 7 章	类设计	137
7.1	简单的磁盘 I/O	137
7.2	设计自己的动态类	140
7.3	对象操作	143
7.4	引用形参	146
7.5	构造函数	150
7.6	继承简介	152
7.7	静态和动态	152
7.8	项目实例	153
7.8.1	Fraction 类的用户手册	153
7.8.2	Fraction 类的定义	154
7.9	小结	156
7.9.1	关键术语	156
7.9.2	概念	157
7.10	练习	158
7.10.1	简答题	158
7.10.2	项目	159
第 8 章	简单数组	161
8.1	数组的声明和引用	161
8.2	数组的初始化	163
8.3	用循环处理数组	164
8.4	常见错误	168
8.5	把数组作为实参和形参	168
8.5.1	传递整个数组	168
8.5.2	传递单个单元	172
8.6	数组检索	173
8.7	对数组进行排序	175
8.8	回顾引用	178
8.9	对象数组	180
8.10	多维数组	181
8.11	项目实例	185
8.12	小结	187
8.12.1	关键术语	187
8.12.2	概念	187
8.13	练习	188
8.13.1	简答题	188
8.13.2	项目	189

第 9 章 IO 流和文本处理	191
9.1 字符串辅助方法	191
9.2 字符串解析	195
9.3 异常处理	198
9.4 Java 流	201
9.4.1 PrintWriter 文本文件流	201
9.4.2 BufferedReader 文本文件流	204
9.4.3 StringTokenizer 类	205
9.5 项目实例	208
9.6 小结	213
9.6.1 关键术语	213
9.6.2 概念	214
9.7 练习	214
9.7.1 简答题	214
9.7.2 项目	215
第 10 章 递归	217
10.1 递归的定义	217
10.2 递归的方法	220
10.3 递归的跟踪	222
10.4 二分查找法	228
10.5 项目实例	230
10.6 小结	234
10.6.1 关键术语	234
10.6.2 概念	234
10.7 练习	235
10.7.1 简答题	235
10.7.2 项目	236
第 11 章 Swing 编程	239
11.1 Swing	239
11.2 窗口组件	240
11.2.1 创建一个标准窗口	241
11.2.2 向框架中添加内容	243
11.2.3 子网格	250
11.3 样例项目	251
11.4 JOptionPane 类	255
11.4.1 showMessageDialog()	256
11.4.2 showInputDialog()	256
11.5 小结	257

11.5.1 关键术语	257
11.5.2 概念	257
11.6 练习	258
11.6.1 简答题	258
11.6.2 项目	260
第 12 章 使用 Applet 编程	261
12.1 Internet 综述	261
12.2 Applet	264
12.3 从 HTML 调用 Applet	266
12.4 将 Swing GUI 转换为 Applet	267
12.5 示例	268
12.6 appletviewer	270
12.7 其他 applet 功能	270
12.7.1 添加图标	270
12.7.2 Applet 参数	272
12.7.3 Timer 与 Date	274
12.8 老式浏览器	276
12.9 applet 的安全性	277
12.10 小结	278
12.10.1 关键术语	278
12.10.2 概念	278
12.11 练习	279
12.11.1 简答题	279
12.11.2 项目	280
附录 A IO、DiskInput 和 DiskOutput 类	281
A.1 IO 类	281
A.2 IO 辅助方法说明	283
A.3 DiskInput 类	284
A.4 DiskInput 辅助方法说明	285
A.5 DiskOutput 类	286
A.6 DiskOutput 类辅助方法说明	287
附录 B Java 保留字	289
附录 C ASCII 码	290
附录 D 奇数编号简答题的答案	292

第 1 章 第一个 Java 程序

本章介绍使用 Java 语言创建第一个简单程序所需要的基本知识。这包括对高级语言、类变量和声明的一个简单介绍。另外，还将学到如何实现简单程序的输入、计算和输出。在本章结束时，读者将能够编写并运行一些完整的 Java 程序，以完成诸如计算圆的面积或解决类似方程式这样的任务。

1.1 用高级语言进行编程

在编写第一个程序之前，让我们来快速地看一看计算机是什么以及它能做什么。尽管有许多种类和型号的计算机，但是我们以许多家庭和学校中都能发现的 PC（个人计算机）来作为模型（见图 1.1）。学校实验室或家庭中的 PC 由 5 或 6 个基本部分组成：中央处理单元（或 CPU）、内部存储器、辅助或第二存储器、一个由键盘和鼠标组成的输入设备、一个监视器输出或显示设备以及可能还有一台打印机。

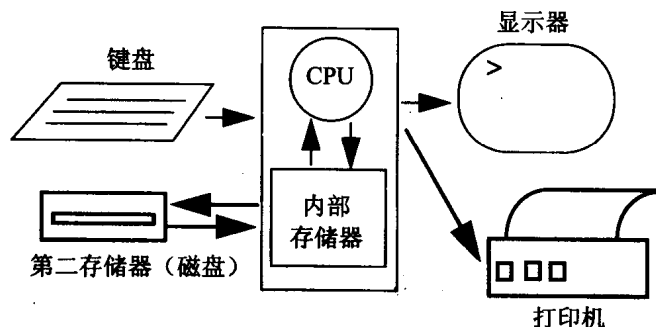


图 1.1 简单计算机系统的基本组成部分

存储器可以存储两类信息：程序和数据。程序就是一个指令序列。在处理器的控制下，键盘可以用来将这两类信息输入并存储到计算机的存储器中。稍后处理器就可以从这个存储器中读取程序和数据，并用它们来产生有用的结果。现在，读者也许会问什么是磁盘驱动器。磁盘只是另一种类型的存储器。存储器可以分为两个简单的种类：断电时丢失所存储内容的存储器（易失性存储器）和断电时仍然保留所存储内容的存储器（非易失性存储器）。内部存储器可以由易失性和非易失性存储器两部分组成。磁盘存储器是非易失性的存储器。

计算机的内部非易失性存储器通常是只读的。换句话说，它们由工厂进行设置，以用于保存一些数据。这种存储器通常叫做 ROM（只读存储器）。内部存储器中易失性的部分通常叫做 RAM（随机访问存储器）（也许读/写存储器的一个更为贴切的名字应该叫做 WRM，但是这种缩写比较难于记忆和辨认）。有几种更为新型的 RAM 存储器也是非易失

性的，但是它们一般只用于专用的计算机中。

由于断电时计算机内部的易失性存储器容易擦除或丢失数据，所以磁盘是特别有用的。在计算机再次启动时，存储在磁盘上的数据和程序后仍然存在并且可以使用。很显然，监视器可以用于显示来自内存的信息，就像打印机一样。因此，它们都是输出设备。CD-ROM 是另一种类型的存储设备，它一般只能写一次，但以后能读取多次。

PC 完成的所有处理过程都由 CPU 来处理和控制。遗憾的是，CPU 不能执行非常复杂的运算。甚至最为强大的 CPU 的功能也仅限于基本的数据处理能力，诸如加、减、乘、除、比较两个值、把数据或指令移入和移出内存以及类似的基本运算。这些命令就是通常所说的 CPU 基本汇编指令集或 CPU 机器语言。

通常我们说处理器的功能越强大，它可以执行基本指令的速度就越快——通常每秒执行好几百万次指令。如果处理器的运算能力很一般，那么计算机如何才能完成非常困难的任务（比如设计飞机或绘制气象图）呢？复杂性并不在于计算机的能力，而在于计算机程序的能力！

程序员的任务就是设计出基本指令集命令的一个逻辑序列，以得到（比较复杂的）所需结果。这一真正的能力在于程序员的智力和技术。例如，下面这组类似英语的 CPU 命令（每个命令代表一个机器指令）可能表示某个假设的计算机来计算一个圆的周长（使用公式 $\text{circ} = 2\pi r$ ）：

```
LOAD    radius
LOAD    pi
MULTIPLY
LOAD    two
MULTIPLY
STORE   circle
```

这里的 radius、pi、two 和 circle 只是一些简单的名字，可以用于表示这些值在存储器中存储的符号位置或地址（我们暂时忽略了所需的这些值是如何首次进入存储器的）。正如读者所认为的那样，如果程序员必须要用基本指令集运算（加、减、乘、除、比较和移动数据）来表示这些指令，那么复杂程序编写起来将会非常困难而且很容易出错。这就是最初的计算机的实际情况。但是，计算机科学家没有花太长的时间去判定必须还有一种更好的解决方法。

对于最初的计算机科学家而言另外一个问题也是很明显的。这就是每个不同的 CPU 制造商针对其特定的 CPU 系列通常会有一个稍微不同的基本指令集。Intel CPU 的机器语言可能和 Macintosh 计算机中 Motorola CPU 的机器语言有着相同的基本功能，但是这些功能的编码和实现却是不同的。因此，为 Intel CPU 编写的机器语言程序在 Macintosh 机器上可能无法工作，反之亦然。这样的程序是不能移植的。它不能从某一系列的计算机上直接放到其他系列的计算机上且仍然可以工作。这时程序需要彻底重新编写。

当开始编写程序的时候，也就是将公式进行文本化表示（代表许多机器语言运算）并将该表示转换成给定 CPU 适当的基本指令集运算的时候，编程就开始出现重要的变化。例如，圆周长的计算可以简单地表示为：