

程序设计基础

[美] 托尼·加迪斯 (Tony Gaddis) 著

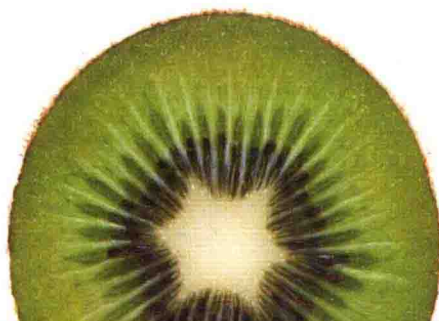
王立柱 刘俊飞 译

Starting Out with Programming Logic & Design

Third Edition

starting out with >>> PROGRAMMING
LOGIC AND DESIGN

3RD EDITION



TONY GADDIS



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

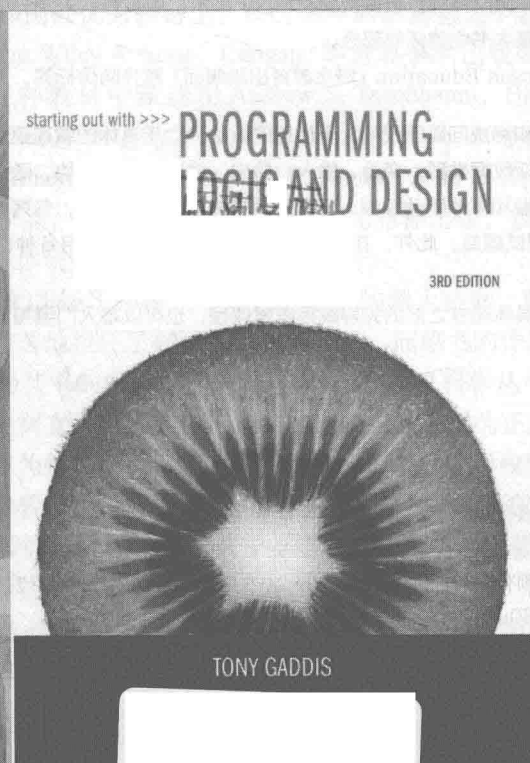
原书第3版

程序设计基础

[美] 托尼·加迪斯 (Tony Gaddis) 著

王立柱 刘俊飞 译

Starting Out with Programming Logic & Design
Third Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

程序设计基础: 原书第 3 版 / (美) 托尼·加迪斯 (Tony Gaddis) 著; 王立柱, 刘俊飞译.
—北京: 机械工业出版社, 2018.4

(计算机科学丛书)

书名原文: Starting Out with Programming Logic & Design, Third Edition

ISBN 978-7-111-59680-6

I. 程… II. ①托… ②王… ③刘… III. 程序设计 IV. TP311.1

中国版本图书馆 CIP 数据核字 (2018) 第 072774 号

本书版权登记号: 图字 01-2012-4852

Authorized translation from the English language edition, entitled *Starting Out with Programming Logic & Design*, 3E, 9780132805452 by Tony Gaddis, published by Pearson Education, Inc., Copyright © 2013.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press, Copyright © 2018.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

本书教授编程概念和解决问题的技巧, 所使用的方法独立于具体的编程语言, 且不需要读者有任何编程经验。内容不仅包括数据类型、变量、输入、输出、控制结构、模块、函数、数组和文件, 还有面向对象的概念、GUI 开发和事件驱动编程。配套的教学资源也十分丰富, 包括复习中的答案、编程练习中的解决方案、PPT 和试题库。此外, 在本书的相应网站上还提供了用各种语言实现的书中伪代码对应的程序。

本书适合作为学习具体语言之前的先导编程逻辑课程, 也可以是入门编程课程的第一部分。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 唐晓琳

责任校对: 殷 虹

印 刷: 北京瑞德印刷有限公司

版 次: 2018 年 5 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 27

书 号: ISBN 978-7-111-59680-6

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来,源远流长的科学精神和逐步形成的学术规范,使西方国家在自然科学的各个领域中取得了垄断性的优势;也正是这样的优势,使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中,美国的产业界与教育界越来越紧密地结合,计算机学科中的许多泰山北斗同时身处科研和教学的最前线,由此而产生的经典科学著作,不仅肇划了研究的范畴,还揭示了学术的源变,既遵循学术规范,又自有学者个性,其价值并不会因年月的流逝而减退。

近年,在全球信息化大潮的推动下,我国的计算机产业发展迅猛,对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇,也是挑战;而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下,美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此,引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用,也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始,我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力,我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系,从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品,以“计算机科学丛书”为总称出版,供读者学习、研究及珍藏。大理石纹理的封面,也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助,国内的专家不仅提供了中肯的选题指导,还不辞劳苦地担任了翻译和审校的工作;而原书的作者也相当关注其作品在中国的传播,有的还专门为其书的中译本作序。迄今,“计算机科学丛书”已经出版了近两百个品种,这些书籍在读者中树立了良好的口碑,并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化,教育界对国外计算机教材的需求和应用都将步入一个新的阶段,我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

华章网站: www.hzbook.com

电子邮件: hzsj@hzbook.com

联系电话: (010) 88379604

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037



华章科技图书出版中心

随着计算机与人类工作、生活的日益融合，人们的大部分工作要基于计算机来完成，越来越多的工作岗位要求从业者具备编程能力。因此，编程能力也逐渐成为人才的竞争力之一。

对于初学者而言，学习编程有很多难点，包括选择合适的编程语言、理解通过编程语言将人类意图转换为计算机程序的过程和方法等。同时，初学者往往容易拘泥于编程语言的语法细节，而忽视了对编程逻辑的理解，最终难以真正掌握通过编程解决实际问题的方法。

针对上述问题，本书采用了伪代码来讲授编程逻辑和方法，从而很好地解决了上述难题。本书的独特之处可总结为以下几点：

1. 本书使用的伪代码最接近具体的编程语言，而且简单实用，避免读者陷入繁杂的语法细节。语句表达洗练，语句功能明显。

2. 本书的算法示例经过精心设计，具体贴切，始终将正在学习的和已经学过的内容进行比较，前后关联，循序渐进，逻辑流畅，一气呵成，使读者具有不断增长的成就感和亲切感。

3. 本书将算法步骤、伪代码描述和程序流程图结合得天衣无缝，相得益彰，既照顾了读者的偏好，又促进了读者的全面发展。

4. 本书几乎每一章都有“重点聚焦”，将该章的知识点和代码都综合运用在实用的程序设计中，活学活用。

5. 每一章结尾都有知识点测试，随时随地进行精准辅导。特别是每一章结尾都有复习内容，包括多选题、单选题、简答题、算法工作、代码检查、编程练习，内容丰富详尽，堪比题库。学生可自学自查，教师可线上线下辅助教学。

6. 本书在相应的网站上提供了与本书伪代码程序对应的、用各种具体编程语言实现的程序。这不但弥补了因独立于编程语言而可能产生的理论与实验脱节的缺陷，而且大大扩展了本书的适用范围：凡是讲授程序语言设计的教师几乎都可以使用本书作为教材。

7. 本书扩大了课堂教学和实验教学的张力：使用伪代码讲授，可以突出算法和逻辑及其程序设计的核心内容；把伪代码转变为具体语言的实现可以充分调动学生理论付诸实践的主观能动性。

8. 教师资源丰富：包括复习的答案、编程练习的解决方案、试题库和授课的 PPT。

很高兴有机会翻译这本理念独特的程序设计教材，但由于翻译周期的要求，本书译稿中难免有翻译不当之处，请读者和同行不吝指正，我们将不胜感激。

译者

2018 年 4 月

欢迎学习本书第3版。本书教授编程概念以及解决问题的技巧，所使用的方法独立于具体的编程语言，不需要读者拥有任何编程经验。使用易于理解的伪代码、流程图和其他工具来学习程序逻辑的设计，规避了语法的困扰。

本书的基本主题不仅包括数据类型、变量、输入、输出、控制结构、模块、函数、数组和文件，还有面向对象的概念、GUI开发和事件驱动编程。本书文字清晰易懂，让学生感到友好和亲切。

本书各章都提供了大量的程序设计示例。短的示例突出编程主题，长的示例集中于问题求解。每章至少包括一个“重点聚焦”小节，对一个具体的问题逐步分析和求解。

本书是学习编程逻辑的理想选择，在用具体的语言学习编程基础之前，本书可以作为先导。

第3版的变化

本书的教学方法、内容组织和写作风格与上一版保持一致，但也做了很多改进，概括如下：

- 详细指导学生设计他们的第一个程序

第2章增加了2.8节。这一节将展示从分析一个问题到确定它的需求的全过程。在这个过程中将用一个示例使学生了解如何确定一个程序的输入、处理和输出，然后编写伪代码和绘制流程图。

在第2章的“重点聚焦”小节，还添加了一个新内容，以计算手机超时费用为例，演示了从确定手动计算的步骤到将这些步骤转换为计算机算法的过程。

- 新调试练习

大部分章节都添加了一组新的调试练习。让学生检查一组伪代码算法并识别其中的逻辑错误。

- 流程图和伪代码之间的一致性更高

在整本书中，许多流程图已经修改，使它们与伪代码之间的联系更紧密。

- 嵌套重复结构扩展

在5.6节扩展了一个示例。

- 附加重复结构的可视化说明

在第5章的Do-While和For循环部分添加了新的可视化说明。

- 文件规范文档和打印间隔图

文件规范文档和打印间隔图在第10章讨论。

- 新的编程语言伴侣

增加了新的语言伴侣Python 3和C++。本书的语言伴侣都可以在网站 www.pearsonhighered.com/gaddis 上找到。

各章简介

第 1 章, 首先简要介绍计算机的工作原理、数据的存储和操作系统, 以及为什么我们用高级语言编写程序。

第 2 章, 介绍程序开发周期、数据类型、变量和顺序结构。学习使用伪代码和流程图来设计简单程序, 包括读取输入、执行数学运算和生成屏幕输出。

第 3 章, 演示模块化程序和自顶向下设计方法的好处。学习定义和调用模块、给模块传递实参、使用局部变量。引入层次结构图作为设计工具。

第 4 章, 介绍关系运算符和布尔表达式, 以及用决策结构进行程序流程控制的方法。还介绍 If-Then、If-Then-Else 和 If-Then-Else If 语句、嵌套决策结构、逻辑运算符、Case 结构。

第 5 章, 学习用循环创建循环结构的方法。包括 While、Do-While、Do-Until 和 For 循环, 还有计数器、累加器、运行总和和哨兵。

第 6 章, 首先讨论通用库函数, 例如生成随机数的函数。然后, 在学习如何调用库函数以及如何使用函数返回值之后, 学习如何定义和调用自定义函数。

第 7 章, 讨论用户输入验证的重要性。学习编写用于错误陷阱的输入验证循环。讨论的内容还有: 防御性编程、对明显和不明显错误进行预测的重要性。

第 8 章, 学习一维数组和二维数组的创建和使用。包含许多数组处理的示例, 包括对一维数组元素求和、计算平均值、求数组最大值和最小值, 以及对二维数组的行、列和全部元素求和。还演示了使用并行数组进行编程的技术。

第 9 章, 学习数组排序和数组元素查找的基础算法。包括的内容有: 起泡排序、选择排序、插入排序和折半查找算法。

第 10 章, 介绍顺序文件的输入和输出。学习读取和写入大集合数据, 将数据另存为字段或记录, 设计可用于处理文件和数组的程序。该章最后讨论了中断处理控制。

第 11 章, 讲述如何设计程序, 该程序显示菜单, 并根据用户的菜单选项来执行。该章还讨论了模块化菜单驱动程序的重要性。

第 12 章, 详细讨论文本处理。包括对字符串逐个字符处理的算法, 用于字符和文本处理的若干常用库函数。

第 13 章, 讨论递归及其用途。提供递归调用过程的可视化跟踪, 讨论递归应用程序。给出许多递归算法, 例如计算阶乘、求最大公约数 (GCD)、数组求和、折半查找, 还有经典的汉诺塔算法。

第 14 章, 将过程化和面向对象的编程进行了比较。包含类和对象的基本概念, 讨论了域、方法、访问规范、构造函数、访问器和变异器。学习如何使用 UML 来对类建模、如何在特定问题中寻找类。

第 15 章, 讨论 GUI 应用程序的基本内容, 以及如何使用可视化设计工具 (如 Visual Studio 或 NetBeans) 构建图形用户界面。学生要学习事件是如何在 GUI 应用程序中工作的, 以及如何编写事件处理程序。

附录, 列出了 ASCII 字符集, 与前 127 个 Unicode 字符编码相同。

内容组织

本书以逐步推进的方式讲授编程逻辑和设计。每一章都包含一组主题, 学生只要按部就

班地学习，就可以掌握本书的知识。虽然这些章节按照现有的顺序可以很轻松地讲授，但还是有一些灵活性。图 P-1 显示了各章之间的依赖关系。每个框代表一章或几章。箭头指向（即箭头终止）的章必须在箭头起始的各章节之前讲完。虚线表示第 10 章只有一部分内容依赖第 8 章的知识。

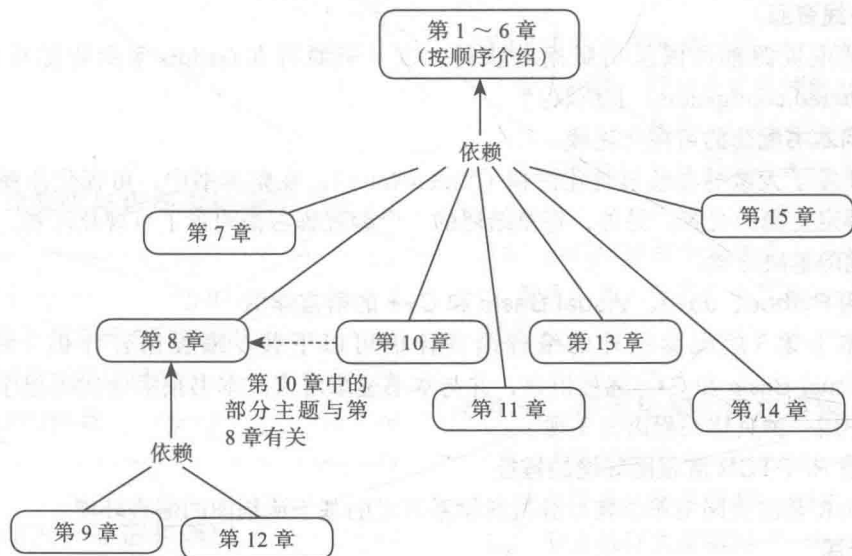


图 P-1 各章之间的依赖关系

本书的特点

概念陈述 本书的每一节都以概念陈述开头，简要概述了本节的要点。

示例程序 每章都有丰富的示例程序，有些是完整代码、有些只是部分代码。每个示例程序都旨在突出当前的主题。示例程序使用伪代码、流程图和其他设计工具。

重点聚焦 每章的“重点聚焦”都有一个或多个案例，详细地、逐步地分析问题和解决问题。

可视化注释 一系列专为本书开发的在线视频可在 www.pearsonhighered.com/gaddis 上查看。图标出现在文本中，针对特定的主题，提示学生关注相关的视频。

注意：“注意”在书中出现若干处，对与主题相关的有意义或经常误解的知识点给予简短的解释。

提示：“提示”建议学生了解可用于处理不同的编程或动画问题的最佳方法。

警告：“警告”告知学生有关可能导致程序故障或丢失数据的编程技术或做法。

编程语言伴侣 本书许多伪代码程序也用 Java、Python 和 Visual Basic 编写。这些程序出现在 www.pearsonhighered.com/gaddis 上的“编程语言伴侣”中。每个伪代码程序旁边的图标也出现在语言伴侣中。

知识点 在每一章的小节末尾大都会以提问形式列出知识点，旨在学完每一个新的主题之后，快速检查学生掌握的情况。

复习 每章都提供了一套全面而多样的复习题和编程练习，它们包括多项选择、判断正误、简答、算法工作台、调试练习和编程练习。

补充

学生在线资源

许多学生资源都可以从出版商处获得。以下资源可在 Gaddis 系列资源页面 (www.pearsonhighered.com/gaddis) 上获取：

- **访问本书配套的可视化注释**

已经开发了大量的在线可视化注释 (VideoNotes)。在整本书中，可视化注释图标提醒学生观看特定主题的视频。另外，每章结尾的一个编程练习都附带了可视化注释，介绍了如何开发问题的解决方案。

- **访问 Python、Java、Visual Basic 和 C++ 的语言伴侣**

专为本书第 3 版配套设计的编程语言伴侣可以下载。编程语言伴侣介绍了 Java、Python、Visual Basic 和 C++ 编程语言，并与本书逐章对应。本书很多伪代码程序都在编程语言伴侣中用一种具体编程语言实现了。

- **下载 RAPTOR 流程图环境的链接**

RAPTOR 是由美国空军学院计算机科学系开发的基于流程图的编程环境。

教师资源^①

以下补充资源仅供有资格的教师使用：

- 复习中所有问题的答案
- 编程练习的解决方案
- 每章的 PowerPoint 演示幻灯片
- 试题库

访问 Pearson Instructor Resource Center (<http://www.pearsonhighered.com/irc>) 或发送电子邮件至 computing@aw.com。

① 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师请联系机械工业出版社华章公司，电话：136 0115 6823，邮箱：wanguang@hzbook.com。——编辑注

出版者的话

译者序

前言

第 1 章 计算机与程序设计导论 1

1.1 引言 1

1.2 硬件 2

1.3 计算机是如何存储数据的 5

1.4 程序是如何执行的 8

1.5 软件的种类 14

复习 15

第 2 章 输入、处理和输出 18

2.1 设计一个程序 18

2.2 输出、输入和变量 21

2.3 变量赋值和计算 27

重点聚焦：计算手机的超时话费 29

重点聚焦：计算百分比 31

重点聚焦：计算平均值 32

重点聚焦：将一个数学公式转换为编程语句 34

2.4 变量声明和数据类型 36

2.5 命名常量 40

2.6 手动跟踪程序 41

2.7 程序文档 42

重点聚焦：使用命名常量、风格约定和注释 43

2.8 设计你的第一个程序 44

复习 47

第 3 章 模块 52

3.1 模块简介 52

3.2 定义和调用模块 53

重点聚焦：模块的定义和调用 57

3.3 局部变量 61

3.4 将参数传递给模块 63

重点聚焦：将一个实参传给一个

模块 66

重点聚焦：通过引用传递一个

实参 71

3.5 全局变量和全局常量 73

重点聚焦：使用全局常量 74

复习 76

第 4 章 决策结构和布尔逻辑 82

4.1 决策结构简介 82

重点聚焦：使用 If-Then 语句 86

4.2 双重选择决策结构 88

重点聚焦：使用 If-Then-Else 语句 89

4.3 比较字符串 92

4.4 嵌套决策结构 95

重点聚焦：决策结构的多重嵌套 98

4.5 Case 结构 101

重点聚焦：使用 Case 结构 103

4.6 逻辑运算符 105

4.7 布尔变量 110

复习 111

第 5 章 循环结构 116

5.1 循环结构简介 116

5.2 条件控制循环：While、Do-While 和 Do-Until 117

重点聚焦：设计一个 While 循环 120

重点聚焦：设计一个 Do-While 循环 126

5.3 计数控制循环和 For 语句 130

重点聚焦：使用 For 语句设计一个计数控制循环 135

5.4 计算运行总和 143

5.5 哨兵 145

重点聚焦：如何使用哨兵	146	9.4 折半查找算法	257
5.6 嵌套循环	148	重点聚焦：使用折半查找算法	260
复习	151	复习	262
第 6 章 函数	155	第 10 章 文件	266
6.1 函数简介：生成随机数	155	10.1 文件的输入和输出	266
重点聚焦：使用随机数	157	10.2 采用循环处理文件	275
重点聚焦：用随机数表示其他值	158	重点聚焦：处理文件	278
6.2 写自己的函数	161	10.3 使用文件和数组	282
重点聚焦：基于函数的模块化	165	10.4 处理记录	283
6.3 更多的库函数	172	重点聚焦：添加和显示记录	286
复习	181	重点聚焦：搜索记录	289
第 7 章 输入验证	185	重点聚焦：修改记录	290
7.1 垃圾入，垃圾出	185	重点聚焦：删除记录	295
7.2 输入验证循环	186	10.5 控制中断逻辑	297
重点聚焦：设计一个输入验证 循环	187	重点聚焦：使用控制中断逻辑	298
7.3 防御性编程	191	复习	302
复习	192	第 11 章 菜单驱动程序	306
第 8 章 数组	195	11.1 菜单驱动程序简介	306
8.1 数组基础知识	195	11.2 模块化菜单驱动程序	314
重点聚焦：在数学表达式中使用 数组元素	199	11.3 使用循环重复菜单	318
8.2 数组的顺序搜索	206	重点聚焦：设计菜单驱动程序	320
8.3 数组的数据处理	210	11.4 多级菜单	332
重点聚焦：处理数组	216	复习	336
8.4 并行数组	221	第 12 章 文本处理	340
重点聚焦：并行数组的应用	221	12.1 引言	340
8.5 二维数组	224	12.2 逐字符文本处理	341
重点聚焦：二维数组的应用	227	重点聚焦：密码验证	343
8.6 三维或高维数组	231	重点聚焦：电话号码格式化和 去格式化	347
复习	232	复习	351
第 9 章 数组的排序和查找	237	第 13 章 递归	356
9.1 起泡排序算法	237	13.1 递归介绍	356
重点聚焦：使用起泡排序算法	242	13.2 递归求解	358
9.2 选择排序算法	248	13.3 递归算法举例	361
9.3 插入排序算法	253	复习	369

第 14 章 面向对象设计 372

14.1 过程化编程及面向对象编程 372

14.2 类 374

14.3 使用统一建模语言来设计类 383

14.4 寻找一个问题中的类及其功能 ... 384

 重点聚焦：寻找一个问题中的类 384

 重点聚焦：定义类的功能 387

14.5 类的继承 392

14.6 类的多态性 397

复习 401

第 15 章 GUI 应用程序和事件

 驱动编程 405

15.1 图形交互界面 405

15.2 设计 GUI 程序的用户接口 407

 重点聚焦：设计一个窗口 410

15.3 编写事件处理程序 412

 重点聚焦：设计一个事件

 处理程序 414

复习 415

附录 ASCII/Unicode 字符 418

计算机与程序设计导论

1.1 引言

想一想,人们使用计算机的一些不同方式。在学校,学生使用计算机写论文、搜索文章、发送电子邮件、学习网络课程。工作时,人们使用计算机分析数据、准备演讲、进行商业交易、与客户和同事沟通、完成设备制造中的机器控制,等等。在家里,人们使用电脑支付账单、网上购物、与朋友和家人聊天、玩计算机游戏。别忘了,还有手机、iPod、黑莓、汽车导航系统,以及许多其他的设备,它们也是计算机。在我们的日常生活中,计算机的使用几乎无处不在。

计算机能做各种各样的事情是因为人们能够为它编写程序。这意味着,计算机设计出来不只是一项工作,而是要做程序所指示它的任何一项工作。一个程序是一台计算机要完成一项工作而要执行的一组指令。例如,图 1-1 显示了两种常用程序的屏幕截图:微软 Word 和 PowerPoint。

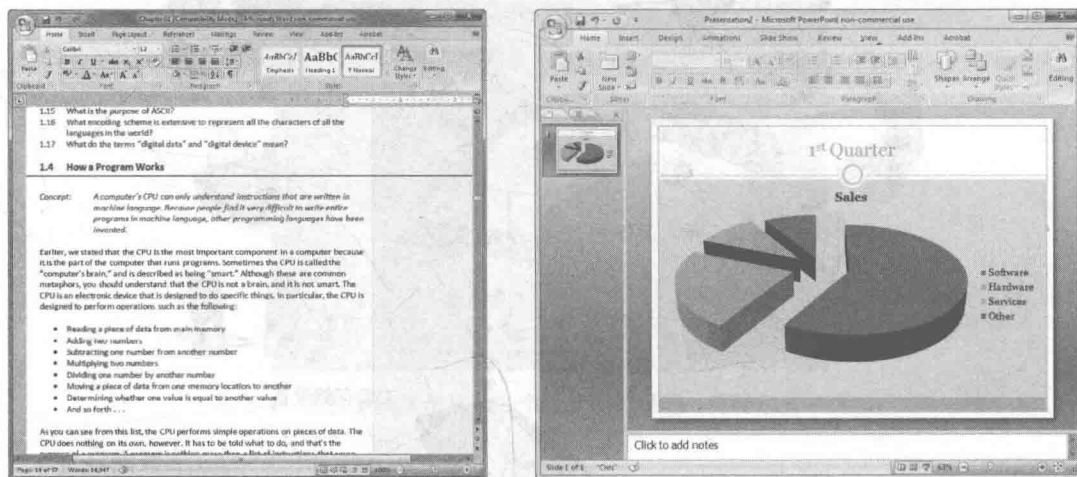


图 1-1 通常使用的程序

程序通常称为软件。软件是计算机必不可少的配置,没有软件,计算机一事无成。使计算机大有用场的所有软件都是程序员或软件开发人员编写的。设计、编写和测试计算机程序的人,都经过了必要的培训,具有相应的技能。程序员或软件开发人员就是这样的人。计算机编程是一个令人兴奋、报酬丰厚的职业。今天,你会发现商业、医药、政府、执法、农业、学者、娱乐,几乎每一个领域都需要程序员编程。

这本书向你介绍计算机编程的基本概念。在开始学习这些概念之前,你需要了解一些关于计算机的基本内容以及它们的工作方式。本章将建立一个坚实的基础知识,依靠这个基础知识,你才可以继续学习计算机科学。首先,我们将讨论计算机构成的物理组件。接下来,

我们将考察计算机是如何存储数据和执行程序的。最后，我们将讨论计算机软件的几种主要类型。

1.2 硬件

概念：构成计算机的物理设备称为计算机硬件。大多数计算机的硬件都类似。

术语“硬件”是指构成一台计算机的所有物理设备或组件。计算机不是一个单一的设备，而是一个一起工作的设备系统。就像一个交响乐团中的各种乐器，计算机的每一个组件都扮演着自己的角色。

如果你买过电脑，大概看过销售手册中列出的计算机组件，例如微处理器、内存、硬盘驱动器、显示器、显卡等。除非你（或者至少你有一个朋友）已经具备很多计算机的知识，否则要懂得这些组件是做什么的，是很头疼的事。如图 1-2 所示，一个典型的计算机系统主要包含以下组件：

- 中央处理单元 (CPU)
- 主存储器
- 二级存储设备
- 输入设备
- 输出设备

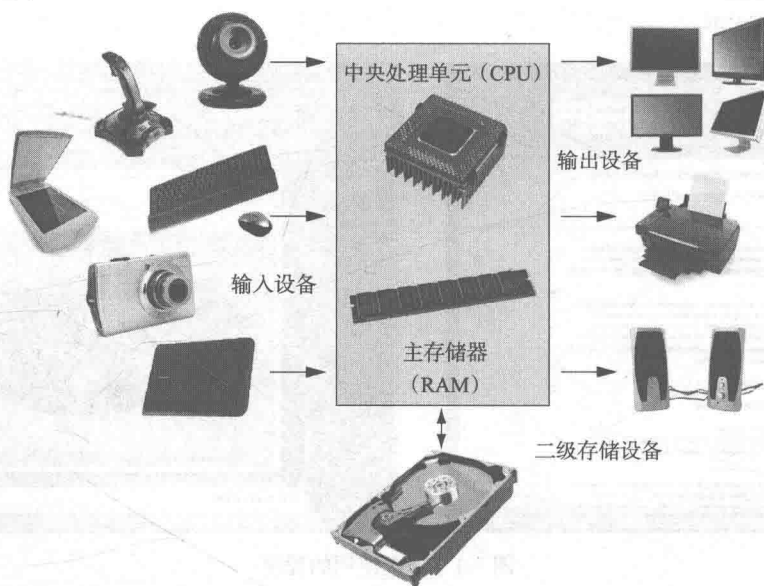


图 1-2 典型的计算机系统组件 (所有照片 © Shutterstock)

下面我们仔细了解每一个组件。

中央处理单元

一台计算机，当按照一个程序来完成一个计算任务时，我们说它正在运行或执行程序。计算机中实际运行程序的部分称为中央处理单元，简称 CPU。CPU 是一台计算机最重要的组件，没有它，计算机就无法运行程序。

在最早的计算机中，CPU 是由真空管和开关等电气和机械部件构成的巨大设备。图 1-3 显示的便是这样一个设备。这是一张历史性的照片，两位女性与 ENIAC 计算机一起工作。

ENIAC 建于 1945, 用来计算美国陆军火炮弹道表, 许多人认为它是世界上第一台可编程的电子计算机。这台计算机, 高 8 英尺 (1 英尺 = 0.3048 米), 长 100 英尺, 重 30 吨, 主要是一个巨大的 CPU。

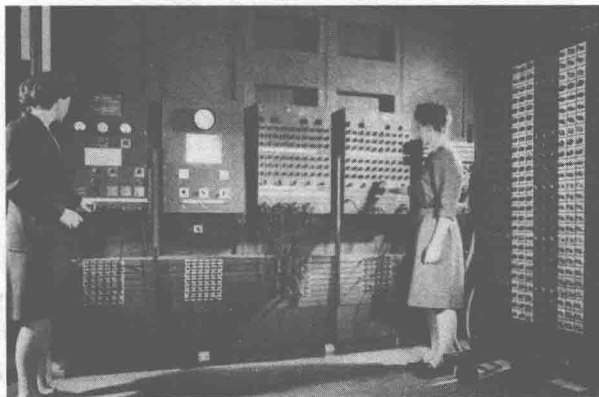


图 1-3 ENIAC 计算机 (由美国陆军提供的历史性计算机图片)

而今天的 CPU 不过是一个小芯片, 称为微处理器。在图 1-4 中, 一个实验室技术员显示给我们看的便是一个现代微处理器。它不仅比早期的旧电机时代的 CPU 要小得多, 而且功能也强大得多。



图 1-4 实验室技术员显示一个现代微处理器 (英特尔公司供图)

主存储器

你可以把主存储器看作计算机的工作间。当程序运行时, 程序和程序所处理的数据都存储在主存储器中。例如, 当你正在使用文字处理程序写一篇结课论文时, 文字处理程序和这篇论文都存储在主存储器中。

主存储器通常称为随机存储器, 即 RAM。之所以称作随机存储器, 是因为 CPU 能够快速访问随机存储在其中任何一个区域中的数据。RAM 通常是一种不稳定的存储器, 只是用来为正在运行的程序提供临时的存储。当计算机关机时, RAM 所存储的内容都将擦除。在你的计算机里, RAM 安装在内存条里, 如图 1-5 所示。

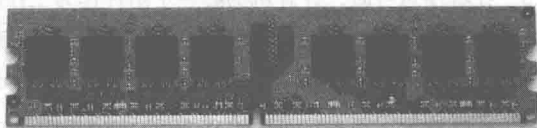


图 1-5 内存条

二级存储设备

二级存储设备能够长期存储数据,即使计算机断电也是如此。程序一般保存在二级存储设备,需要时加载到主存储器。文档、工资表、库存记录等这类重要的数据,也都应该保存在二级存储设备中。

最常用的二级存储设备是磁盘驱动器。磁盘驱动器是以磁盘为介质来记录数据的存储装置。目前大多数计算机都内置磁盘驱动器。也有外置磁盘驱动器,它们与计算机的一个通信端口连接。通常,外置磁盘驱动器可以用来备份重要的数据或将一台计算机上的资料转移到另一台计算机。

除了外置磁盘驱动器,还有很多种设备用来备份数据或转移数据。曾风靡多年的软盘驱动器便是其中的一种。软盘驱动器把数据记录在一个小的软盘上,这个软盘可以从驱动器中移除。然而软盘有很多缺点:它只能保存少量的数据,读取速度慢,有时还不稳定。近年来,软盘驱动器的用户数量急剧下降,取而代之的是一种性能优越的设备——USB 驱动器(简称U盘)。这是一种小型设备,与计算机通用串行总线 USB (Universal Serial Bus) 端口连接。对计算机而言,它看似一个磁盘驱动器,而实际上它没有磁盘,是一种特别的存储器,称为闪存,也称记忆棒、闪存驱动器。它廉价、可靠、体积小、易携带。

光学器件,例如 CD (Compact Disk, 高密度光盘) 和 DVD (Digital Versatile Disc, 数字通用盘),也都是非常流行的数据存储设备。CD 和 DVD 驱动器是用光学原理进行数据读取的设备,数据不是记录在磁道上,而是记录在激光烧录的一系列“坑”中。这种光学器件可以保存大量数据,而且数见不鲜,是数据备份和转移的很好介质。

输入设备

输入是计算机从人和其他设备那里收集的任何数据。收集数据并将其发送到计算机的组件称为输入设备。常用的输入设备有键盘、鼠标、扫描仪、麦克风和数码相机。磁盘驱动器和光盘驱动器也可以认为是输入设备,因为程序和数据是经它们检索,并加载到计算机内存的。

输出设备

输出是计算机为人或其他设备产生的任何数据。它可能是一份销售报告、一份名单或一个图形图像。数据发送到输出设备,输出设备将数据格式化,然后呈现出来。常见的输出设备是视频显示器和打印机。磁盘驱动器和 CD 刻录机也可以看作输出设备,因为计算机把数据保存在它们那里。

知识点

- 1.1 什么是程序?
- 1.2 什么是硬件?
- 1.3 列出计算机系统的 5 个主要组件。
- 1.4 计算机中实际上运行程序的是哪一个组件?
- 1.5 在程序运行时,计算机的哪一个组件作为工作区来存储程序和数据?
- 1.6 计算机的哪一个组件可以长时间保存数据,即使在断电的情况下也是如此?
- 1.7 从人和其他设备收集数据的是计算机的哪一个组件?
- 1.8 为人和其他设备把数据格式化然后显示出来的是计算机的哪一个组件?

1.3 计算机是如何存储数据的

概念：存储在计算机中的所有数据都要转换为 0 和 1 的序列。

一个计算机存储器要划分成一组小的存储单位，这个单位称为字节。一个字节只能够存储一个字母或一个很小的数字。要做有意义的事，计算机必须拥有很多字节。今天的大多数计算机都拥有数百万甚至数十亿字节的内存。

每个字节都分为 8 个更小的存储单位，这个单位称为位。位代表二进制数字。计算机科学家通常把位看作小开关，可以打开或关闭。但位实际上不是“开关”，至少不是传统意义上的开关。在大多数计算机系统中，位是微小的电气元件，可以容纳一个正电荷或一个负电荷。计算机科学家把一个正电荷看作开关处在打开的状态 ON，把一个负电荷看作开关处在关闭的状态 OFF。图 1-6 是计算机科学家想象的一个字节的存储：一组开关，每个都处在打开或关闭的状态。

当一个数据存储在一个字节中，计算机就把 8 位分别置于打开或关闭的状态，以此表示这个数据。图 1-7a 是数字 77 在一个字节中的存储方式，图 1-7b 是字母 A 在一个字节中的存储方式。在某一时刻，你将懂得这些模式是如何确定的。

数字存储

一位可以表示的数字非常有限。根据打开状态或关闭状态，它可以表示两个值。在计算机系统中，关闭的位表示数字 0，打开的位表示数字 1。与此完美对应的是二进制编码系统，即通常所称的二进制。在二进制中，所有数值都是 0 和 1 组成的序列。例如：

10011101

二进制的每一数字位都配有一个值，称为位置值。从最右端的数字位开始往左移动，位置值依次为 2^0 , 2^1 , 2^2 , 2^3 等，如图 1-8 所示。图 1-9 显示的是同一个图，只是位置值用数值 1, 2, 4, 8 等来表示。

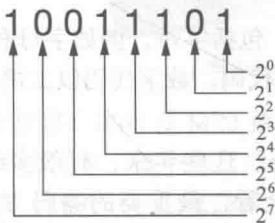


图 1-8 以 2 的幂表示的二进制数字的位置值

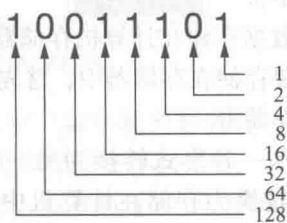


图 1-9 二进制数字的位置值

确定一个二进制数的值，只需将所有 1 的位置值简单相加即可。例如，二进制数

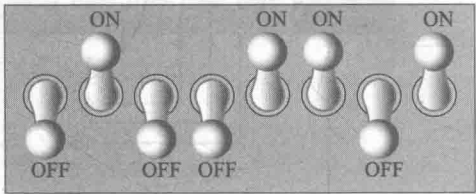
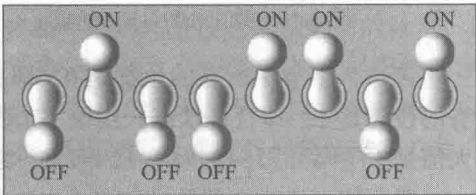
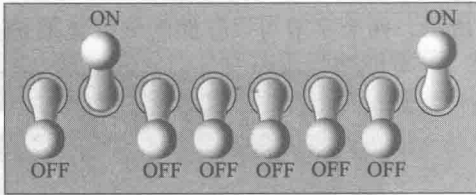


图 1-6 把一个字节想象为 8 个开关



a) 存储在一个字节中的数字 77



b) 存储在一个字节中的字母 A

图 1-7 数字 77 和字母 A 的位模式