

PEARSON



C++ 应用程序设计

[美] Richard Johnsonbaugh, Martin Kalin 著
方红琴 熊利荣 译



中国电力出版社
CHINA ELECTRIC POWER PRESS

C++ 应用程序设计

[美] Richard Johnsonbaugh, Martin Kalin 著

方红琴 熊利荣 译



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书是一本 C++ 入门书籍，是原著作者根据在 DePaul 大学最近几年的教学经验编写而成的。全书共 10 章，主要内容包括计算机系统及程序设计的简单介绍、数据类型和运算符、控制流、函数、数组等基础知识，以及类、继承、多态性，运算符重载、指定存储类别、模板和标准模板库。此外，本书还有 6 个附录，包括 ASCII 码表、C++ 预处理器、部分 C++ 函数和类方法、运行时类型识别、异常处理、奇数序号练习题的解答提示与答案。

本书案例和练习题覆盖的应用范围广，至少包含 350 个实例和 160 道程序设计练习题，案例全部来源于现实的应用。

本书适合从事 C++ 程序开发的初、中级程序人员使用和参考。

图书在版编目 (CIP) 数据

C++ 应用程序设计 / (美) 约翰巴夫 (Johnsonbaugh.R.), (美) 卡林 (Kalin, M.) 著; 方红琴, 熊利荣译. —北京: 中国电力出版社, 2010.12

ISBN 978-7-5123-0892-3

I. ①C… II. ①约…②卡…③方…④熊… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 181761 号

Authorized translation from the English language edition, entitled APPLICATIONS PROGRAMMING IN C++, 1st Edition, 0137489633 by JOHNSONBAUGH, RICHARD; KALIN, MARTIN, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 1999.

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 PEARSON EDUCATION ASIA LTD., and CHINA ELECTRIC POWER PRESS Copyright © 2011.

本书翻译版由 Pearson Education 授权中国电力出版社在中国境内 (香港、澳门特别行政区和台湾地区除外) 独家出版、发行。

未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 防伪标签，无标签者不得销售。

北京市版权局著作权合同登记号 图字：01-2010-5532

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

北京丰源印刷厂印刷

各地新华书店经售

*

2012 年 1 月第一版 2012 年 1 月北京第一次印刷

787 毫米 × 1092 毫米 16 开本 36.5 印张 878 千字

印数 0001—3000 册 定价 68.00 元

敬 告 读 者

本书封面贴有防伪标签，加热后中心图案消失

本书如有印装质量问题，我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

《C++应用程序设计》一书根据作者在 DePaul 大学最近几年的教学经验编写而成。本书的读者可以不具备 C++ 的基础知识，但应该有使用某种高级语言编程的经验。第 0 章是为编程经验较少的读者提供的背景知识。更多水平较高的读者可以从第 1 章开始。本书可以作为一个学期的 C++ 应用程序设计课程的自学教材。

本书及其补充材料——用户指南和网站——提供了帮助用户掌握 C++ 的丰富的支持材料。书中包含大量的实例、练习、应用程序示例、程序设计练习、程序设计中常见的错误及插图。

概述

在 20 世纪 80 年代和 20 世纪 90 年代早期，C 语言变成了许多应用程序和系统程序开发人员所选择的语言，许多个人计算机中使用的大多数主要软件多使用 C 语言编写，如字处理软件、数据库、通信软件包、统计软件、图像处理软件包等。所有 UNIX 环境下的软件同样用 C 语言编写，许多想要从一个平台移植到另一个平台的大型系统也用 C 语言进行编写。在 20 世纪 80 年代早期，AT&T 贝尔实验室的 Bjarne Stroustrup 对 C 进行了扩展，开发出了 C++ 语言，C++ 语言支持面向对象的程序设计。面向对象程序设计很好地适用于所有平台编写大型、复杂的软件系统，以及从便宜的个人计算机向昂贵的大型机中进行软件移植。C++ 还克服了 C 的一些缺点，它支持抽象数据类型，并通过模板支持泛型函数。C 语言是 C++ 的一个子集。

C++ 是一种非常复杂的语言。幸运的是，大多数 C++ 程序员在掌握 C++ 的每一个特征之前就能从 C++ 强大的功能中受益。因为本书是一本入门书籍，所以主要介绍 C++ 中最有用的方面；在每章后面的 C++ 的补充材料包含更多 C++ 中更深奥和特殊的部分。我们的目的是让大家使用耳熟能详的技术编写实用的应用程序，而不仅仅惊叹于 C++ 强大的功能或仅掌握 C++ 中的小窍门。

本书具有以下特征：

- 实例和练习覆盖的应用范围广
- 来源于现实应用中的实例
- 丰富的程序设计练习，本书包含 160 多个程序设计练习
- 对程序设计中常见的错误进行了说明
- 介绍标准 C++ 的函数和方法
- 每个小节后都有相关的练习，读者可以自己检查对本节内容掌握的程度。书中包含近 600 个这样的练习。本书最后给出了每节中奇数序号题目的答案，偶数序号题目的答案在用户指南中给出
- 便于读者学习的插图

- C++语言最新的变化，包括新风格的函数首部，新风格的类型转换，名字空间和名字空间 `std`。
- 对算法的探讨（第 0.1 节）
- 各种计算机系统简介（第 0.2 节）
- 程序设计语言概述（第 0.4 节）
- 对程序开发过程的探讨，包括程序描述，算法设计，编写代码和测试（第 0.5 节）
- 解决问题的策略（第 0.5 节）
- 早期与文件相关的内容（第 2.5 节）
- 深入分析递归（第 3.8 节）
- STL（标准模板库）相关内容（第 10 章）
- 多个附录
- 依据各主题的作用及其关系对主题进行分组。这种组织方式可以让读者能够立刻编写简单有用的程序，而跳过或以后再学语言中很少用而且深奥的部分
- 易懂的代码。我们选择风格清晰的代码，而不是以 C++ 难以理解为借口编写一些晦涩的程序

C++：一种混合语言

C++实质上是将 C 扩展成支持面向对象程序的设计。C 支持传统的、面向过程的程序设计，所以 C++是一种混合性的语言，它既支持面向对象的程序设计，也支持面向过程的程序设计。在同一门课程中既讲解面向过程的程序设计，又讲解面向对象的程序设计并开发出适合两种程序设计的教学策略是一种挑战。本书基于 C++教学的实际经验，在实际教学过程中我们以往的许多观念发生了变化。在掌握 C++中面向对象的部分之前，部分面向过程的程序设计的知识也是需要的，因此，我们首先介绍基本控制结构（如循环、if 语句）、数据类型、运算符、数组和函数。介绍完 C++作为面向过程的语言的基础知识后，我们引入了类、继承和多态——这是面向对象程序设计中的关键概念。接着介绍运算符重载、存储分配、模板和其他 C++的特征。按这种方式安排的理由如下：

- 符合教育学的规律——从稍微抽象的主题（如基本控制结构）到更加抽象的主题（如类和继承）
- 除了一些简单的但价值不大的类外，所有的类都要用到基本的控制结构、数据类型、运算符、数组和函数，即使是完全“面向对象”的程序，类的实现也需要基本的控制结构、数据类型、运算符、数组和函数。

本书的组织

第 0 章简单地介绍了计算机系统和程序开发的相关知识（如果已经掌握了这部分内容可以跳过本章）。第 0.1 节介绍了算法。第 0.2 节描述了计算机系统的概况，包括硬件（如内存、CPU 等）以及软件（如文本编辑器、编译器等）。第 0.3 节介绍了整数、浮点数、字符和指令的内存表示。第 0.4 节介绍了程序设计语言（如汇编语言和高级语言）。第 0.5 节总结了程序开发的各个阶段：程序描述、算法设计、编码和测试。另外，第 0.5 节还给出了几个解决问题的策略，第 0.6 节对 C++语言进行讨论（第 0.6 节）之后结束了第 0 章。

第 1~4 章主要介绍了 C++ 作为面向过程的语言的基础知识，第 1 章讨论数据类型、整型和浮点型，标识符、运算符和算术、关系和逻辑运算符。学完第 1 章后，读者马上就能写出简单但有用的程序了。

第 2 章介绍了 if-else 语句，while、do-while 和 for 循环，文件，赋值、自增和自减运算符，break、continue 和 switch 语句，类型提升和转换，基本的格式化。因为在 Windows 环境中编译的 C++ 程序最好使用磁盘输入和输出文件，本章介绍了文件。

函数和程序结构在第 3 章中介绍。第 3.1~3.3 节介绍了参数和通过值调用的相关知识。第 3.5 节介绍了通过引用调用，第 3.7 节介绍了函数重载，第 3.8 节介绍了递归，第 3.4 节和第 3.6 节是应用示例。

第 4 章主要介绍了数组。因为指针和字符串与数组联系紧密，因而引入了指针和数组相关的内容。

介绍完 C++ 作为面向过程的程序设计语言的相关知识后，我们介绍了其面向对象的特征：类、继承和多态。第 5 章介绍了类的相关知识：类和对象，数据成员和方法，构造器和析构器，类的定义及其在程序中的使用，以及如何使用已有的类库。

第 6 章介绍了继承，包括继承过程中的 protected 成员、构造器和析构器。

第 7 章介绍了多态性（运行时绑定）。另外，本章讨论了运行时绑定与其他具有类似特征的技术之间的区别。本章还说明了怎样将抽象类用作一个具体的接口，并强调了面向对象模型中接口的重要性。

第 8 章介绍对类的运算符的重载，包括使用方法或顶级函数实现运算符重载。本章列举了大量的例子说明如何重载运算符更有利。

第 9 章介绍了存储。本章详细介绍了编译和运行时存储，以及 C++ 主要的存储类：auto、static 和 extern。

第 10 章主要介绍了 C++ 如何通过模板支持泛型函数，包括 STL（标准模板库），并提供了简短的程序说明 STL 容器、迭代器、算法和函数对象的使用。

章的结构

各章基本按以下方式组织：

目录

概述

节

练习

...

C++ 补充材料

程序设计中常见的错误

程序设计练习

第 1~10 章中都有几个小节专门讲解应用程序示例。这些小节中包含问题描述，简单输入、输出示例，问题的解决方法，以及对文件形式的 C++ 程序中的每行进行了讨论。

应用程序示例包括：

- 统计（第 2.9 节）

- 模拟 (Monty Hall 问题) (第 3.4 节)
- 投票计数 (第 4.4 节)
- task 类 (第 5.6 节)
- 胶片跟踪 (第 6.3 节和第 7.2 节)
- 关联数组类 (第 8.7 节)
- 排序和搜索 (第 9.4 节)
- 模板堆栈类 (第 10.2 节)
- 使用 STL 实现股票业绩报告 (第 10.5 节)

C++ 补充资料部分介绍了 C++ 中不常用的部分, 还给出了语言中的技术细节。

程序设计中常见的错误部分主要介绍了 C++ 容易出现错误的情况。

本书包含 160 多个程序设计练习, 覆盖了范围广泛的应用领域。

案例

本书包含超过 350 个例子, 使读者更加清楚地掌握 C++ 某些具体的知识点, 并说明了 C++ 的各种特征, 项目符号 “■” 表示每个实例结束。

练习

本书包含近 600 个对小节知识进行复习的练习, 其中包括答案为正确或错误的判断题, 简答题, 写出代码段, 有很少的一部分要求写完整的程序, 这些练习适合作为家庭作业或作为自测题, 奇数序号练习的答案在书的最后, 偶数序号练习的答案可以参照网站上的用户指南。我们在课堂上使用本书后, 才确信这些练习是非常重要的。

每章最后的程序设计练习覆盖的应用很广, 例如:

- 投票欺诈 (程序设计练习 2.12)
- 举重比赛 (程序设计练习 2.14)
- 模拟 (程序设计练习 3.15 和 5.14)
- 生命的游戏 (程序设计练习 4.18)
- 数字图像 (程序设计练习 4.20)
- 数据结构 (程序设计练习 5.8, 5.2, 9.2~9.7)
- 数据库 (程序设计练习 5.15)
- 活动安排 (程序设计练习 5.18 和 10.14)
- Windows 环境 (程序设计练习 7.8)
- 日期服务程序 (程序设计练习 7.10)
- 软件工程 (程序设计练习 9.1)
- WWW (程序设计练习 10.12)

不是所有的读者都会对每个应用感兴趣, 但是, 我们想让读者知道 C++ 能够解决各种各样的问题。

附录

本书有 5 个附录供读者参考。附录 A 包含 ASCII 码表, 附录 B 详细解释了 C++ 预处理

器。附录 C 包含大多数有用的 C++ 函数和类方法。其中描述了每个方法函数的参数和返回值，包含它的头文件以及功能。附录 D 介绍运行时类型识别（RTTI），附录 E 介绍了异常处理。

语言的最新变化

C++ 语言的最新变化包括：

- 逻辑类型 `bool`
- 新风格的头文件
- 新风格的类型转换
- STL（标准模板库）
- 异常处理
- 运行时类型处理
- 名字空间及名字空间 `std`
- 运算符 `new[]`

全书都使用了这些特征，但对于使用旧编译器的读者，也解释了如何将相关技术转换成 C++ 原有风格的方法。

在介绍逻辑类型 `bool` 之前，用整型来表示错误（0）和正确（1）。

新风格的头文件删除了扩展名 `.h` 并在 `C` 的头文件中增加了前缀 `C`。例如，旧风格的 C++ 头文件 `iostream.h` 变成了 `iostream`，`C` 的头文件 `stdlib.h` 变成了 `cstdlib`。这些头文件包含的标准功能都组合在名字 `std`（标准，Standard）中了。将这些功能组合的技术术语称为名字空间，正像书中所示的一样，为了使用这些标准功能，在所有的 `#include` 指令后增加一行：

```
using namespace std;
```

例如，旧风格的 `#include` 指令

```
#include<iostream.h>
```

被替换成

```
#include <> iostream
using namespace std;
```

原有风格的类型转换

```
(int)x
```

和类型转换

```
int(x)
```

被下面四个类型转换替代

```
const_cast dynamic_cast reinterpret_cast static_cast
```

可以根据使用的情况分别选择其中的一个进行类型转换。

运算符 `new` 和 `new[]` 是有区别的，详细参见书中说明。

网站

本书相关的网站 <http://condor.depaul.edu/~mkalin> 中包含源代码、头文件和所有书中应用实例的数据文件，一些较长的例子的源代码，教学大纲示例，幻灯片及勘误表。

感谢

感谢 Ed Angel（新墨西哥大学），Karen Bernstein（DePaul 大学），Ralph Ewton（德克萨斯大学艾尔帕索分校）对本书的审核。

感谢本书的文字编辑 Patricia Johnsonbaugh，他使我们平淡无奇的文字变得更加流畅，而且删除了那些可有可无的词句。我们的学生 Wen Tao Liu 帮助检查了大部分主要的代码段。

感谢 DePaul 大学计算机科学、通信和信息系统学院的教职员工，特别是 Helmut Epp，他们给了我们很多帮助，并使我们有更多的时间完成本书。

我们一直得到 Prentice Hall 出版社热忱的帮助。特别感谢发行人 R.Apt 和策划编辑 Laura Steele。

R.J

M.K

目 录

前 言

第 0 章 计算机系统及程序设计	1
0.1 算法	2
0.2 计算机系统	4
0.3 数据的内部表示	7
0.4 程序设计语言	13
0.5 解决问题的方法与程序设计	16
0.6 为什么使用 C++	20
第 1 章 数据类型和运算符简介	23
1.1 第一个 C++ 程序	24
1.2 整型	26
1.3 标识符	29
1.4 应用实例：长度单位的转换	30
1.5 其他整数类型	31
1.6 浮点类型	37
1.7 算术运算	38
1.8 应用实例：一个数学问题	41
1.9 关系运算符和逻辑运算符	42
C++ 补充材料	47
程序设计中常见的错误	51
程序设计练习	53
第 2 章 控制流	55
2.1 if-else 语句	56
2.2 应用实例：计算个人所得税	66
2.3 while 语句	68
2.4 应用实例：税款计算（二）	70
2.5 文件	71
2.6 do-while 语句	76
2.7 for 语句	77
2.8 赋值、自增和自减运算符	80
2.9 应用实例：数据统计	85

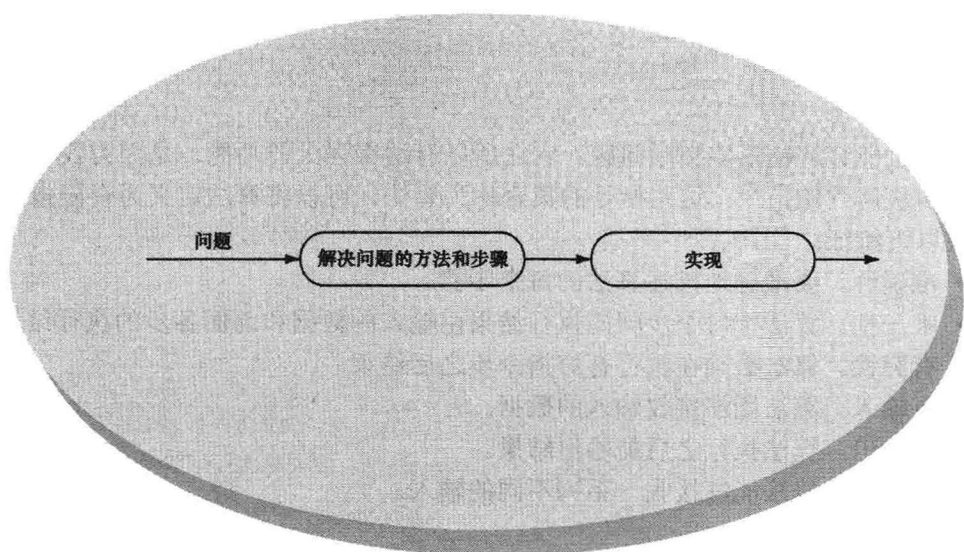
2.10	break、continue 和 switch 语句	87
2.11	应用实例：产生素数	93
2.12	类型提升和类型转换	95
2.13	输出格式化	100
	C++补充材料	104
	程序设计中常见的错误	105
	程序设计练习	108
第 3 章	函数	113
3.1	简介	114
3.2	变量的作用范围	123
3.3	值调用	125
3.4	应用实例：Monty Hall 问题	128
3.5	引用调用	138
3.6	应用实例：输出日历	142
3.7	函数重载	147
3.8	递归	149
	程序设计中常见的错误	157
	程序设计练习	162
第 4 章	数组	167
4.1	为什么使用数组	168
4.2	下标运算符	169
4.3	数组和指针	174
4.4	应用实例：投票统计	177
4.5	字符串	179
4.6	数组和函数	184
4.7	字符串处理函数	188
4.8	应用实例：文件合并	197
4.9	多维数组	204
4.10	应用实例：维护地址簿	210
	C++补充材料	218
	程序设计中常见的错误	219
	程序设计练习	223
第 5 章	类	229
5.1	面向对象程序设计和面向过程程序设计	230
5.2	类和对象	234
5.3	应用实例：Clock 类	243
5.4	应用实例：时间戳类	245

5.5	构造器和析构器	252
5.6	应用实例：Task 类	265
5.7	类的数据成员和方法	271
5.8	使用类库	275
5.9	指向对象的指针和间接访问运算符	277
	C++补充材料	279
	程序设计中常见的错误	280
	程序设计练习	286
第 6 章	继承	291
6.1	简介	292
6.2	基本概念和语法	293
6.3	应用实例：记录电影胶片的发布情况	297
6.4	protected 成员	302
6.5	继承中的构造器和析构器	306
6.6	应用实例：继承序列	312
	C++补充材料	322
	程序设计中常见的错误	325
	程序设计练习	327
第 7 章	多态性	331
7.1	C++中的运行时绑定和编译时绑定	332
7.2	应用实例：电影胶片修订版	342
7.3	名字重载、名字重写和名字隐藏	351
7.4	抽象基类	358
	C++补充材料	362
	程序设计中常见的错误	363
	程序设计练习	366
第 8 章	运算符重载	371
8.1	基本运算符重载	372
8.2	应用实例：复数类	376
8.3	使用顶级函数重载运算符	380
8.4	友元函数	385
8.5	输入/输出运算符的重载	387
8.6	重载特殊运算符	389
8.7	应用实例：数组	398
	C++补充材料	403
	程序设计中常见的错误	403
	程序设计练习	406

第 9 章 存储类别	409
9.1 编译时存储和运行时存储	410
9.2 变量的存储类别	415
9.3 指针和动态存储	428
9.4 应用实例：排序和搜索	437
9.5 带有指针数据成员类	443
C++ 补充材料	453
程序设计中常见的错误	454
程序设计练习	457
第 10 章 模板和标准模板库	463
10.1 模板的基础知识	464
10.2 应用实例：模板栈类	472
10.3 名字空间	480
10.4 标准模板库	483
10.5 应用实例：股票趋势报表	498
C++ 补充材料	506
程序设计中常见的错误	507
程序设计练习	510
附录 A ASCII 码表	514
附录 B 预处理器	516
附录 C 部分 C++ 函数和类方法	525
附录 D 运行时类型识别	541
附录 E 异常处理	545
奇数序号练习题的解答提示与答案	548

第0章

计算机系统及程序设计



- 0.1 算法
- 0.2 计算机系统
- 0.3 数据的内部表示
- 0.4 程序设计语言
- 0.5 解决问题的方法与程序设计
- 0.6 为什么使用C++

算法是指解决问题的方法与步骤。Adlai Stevenson 将烹制鲤鱼的算法描述如下：

1. 将 0.5~1kg 的鲤鱼在清水中放置两天。
2. 去除鱼鳞和鱼骨，切成鱼片。
3. 在鱼片上抹上奶油，并加入适量的盐和胡椒粉。
4. 将鱼片用中火烹制 20min。
5. 烹制完成，可以开始享用了。

从古至今出现过很多经典的算法，最早见于古老的巴比伦王国。实际上，“算法”一词源于 19 世纪波斯数学家 al-Khowārizmī。算法在科学和工程领域发挥着重要的作用。计算机能够自动地执行算法。将算法写成计算机能够识别的形式后，就可以让计算机自动地执行这些算法了。

本章首先介绍了算法的定义及特征、计算机系统以及数据在计算机中的表示。紧接着介绍了程序设计语言、解决实际问题的程序设计技巧，最后简要地介绍了 C++ 程序设计语言。

0.1 算法

计算机能够解决很多实际问题。它之所以有这么强大的功能，是因为我们向计算机中输入了解决问题的指令。这些指令的集合称为算法。可以将算法定义为有限指令的集合，它具有以下特征：

- 准确性。要准确地描述算法的每个步骤。
- 唯一性。算法中每个步骤的执行结果由输入的数据和前面各步的执行结果决定。
- 有穷性。算法必须在执行有穷指令步之后结束。
- 有输入。算法能够接收输入的数据。
- 有输出。算法执行之后能输出结果。
- 通用性。算法能够接收一系列不同的输入。

例 0.1.1 分析下面的算法：

输入： n

输出： $1/1+1/2+\cdots+1/n$

1. $sum=0$
2. $i=0$
3. $i=i+1$
4. $sum=sum+1/i$
5. 重复执行第 3 步和第 4 步，直到 $i==n$ 为止

此算法计算 $\sum_{i=1}^n \frac{1}{i}$ 的前 n 项和。

算法执行时，输入 n ，输出数列前 n 项和。

第 1 行的语句 $sum=0$ 表示“把 0 赋值给变量 sum ”。一般地，语句
变量=表达式

表示“将表达式的值赋给变量”，也就是说，将变量当前的值替换成表达式的值。例如，执行语句

$$x=y$$

后, 将 y 的值赋给 x 、 y 的值不变。表达式中 “=” 的含义是赋值, 称为赋值运算符。

第 4 行的表达式为 $i==n$, 如果 i 的值和 n 的值相等, $i==n$ 的值为真, 否则, $i==n$ 的值为假。表达式中 “==” 的含义为 “相等”, 称为等号运算符。虽然 “=” 和 “==” 看起来非常相似, 但它们的含义却完全不同。

下面来看看这个算法的执行过程, 如果输入的 n 的值为 3。指定一个具体的输入值, 模拟算法的执行过程称为追踪算法执行。第 1 行和第 2 行分别将变量 sum 和 i 赋值为 0。第 3 行的表达式将 $i+1$ 的值赋给 i , 因为 i 原来的值为 0, $i+1$ 的值为 1, 因此执行此表达式后 i 的值为 1。第 4 行将 $1/i$ 的值加到 sum 中, 执行完这行语句后 sum 的值为 1。第 5 行先测试 i 的值是否等于 n , 因为 i 的值为 1, n 的值为 3, 算法重复执行第 3 行和第 4 行。

执行第 3 行后, i 的值变为 2, 接着将 $1/i$ (即 $1/2$) 加到 sum 中, 因此 sum 的值为 $1+1/2$ 。因为 i 的值为 2, 与 n 的值不相等, 算法重复执行第 3 行和第 4 行。

执行第 3 行后, i 的值变为 3, 接着将 $1/i$ (即 $1/3$) 加到 sum 中, 这时 sum 的值为 $1+1/2+1/3$ 。因为 i 等于 3, 与 n 的值相等, 因此算法结束, sum 中存放的是数列前 3 项的和。 ■

必须准确地描述算法中的每一步。例 0.1.1 中所示的对算法每个步骤的描述都非常准确。只有准确地描述算法中的每个步骤, 计算机才能自动地执行算法, 而且任何人在任何机器上执行算法的结果都相同。毫无疑问, 任何人在任何机器上执行例 0.1.1 中的算法时, 对于某个给定的值 n , 都会得到唯一的前 n 项和 sum 。

只要输入的值相同, 算法的每个中间步骤都会产生唯一的结果。例如, 当程序执行过程中 sum 和 i 的值分别如下:

$$sum=1+1/2, \quad i=3$$

执行例 0.1.1 中算法的第 4 行后, sum 的值会变成 $1+1/2+1/3$, 任何人在任何机器上执行这个算法时都会得到相同的结果。

算法必须在有穷的步骤之内停止, 并且能得到所给问题的一个解。例 0.1.1 中的算法在经过 n 次循环后运行结束, 算法结束时能正确地计算数列的前 n 项和。

算法能接收输入并能输出运行结果。例 0.1.1 中的算法执行时会要求输入 n , 执行结束时输出 sum 的值。

算法应该具有通用性。例 0.1.1 中的算法执行时, 对于任意给定的 n 的值都能输出数列的前 n 项和。

现代计算机让人们从手工操作中解脱出来。许多以前由人工完成的工作, 如由科学家、工程师、会计等手工完成的表格现在都由计算机自动完成。这样既可以节省大量的时间, 还能提高事务处理的准确度。此外, 计算机还能处理由其他方式无法完成的许多复杂的算法。

练习

1. 写出算法, 计算 $\sum_{i=1}^{\infty} \frac{1}{(1+i)^2}$ 的前 n 项和。

2. 写出算法, 找出 a 、 b 、 c 中最小的数。

3. 写出算法, 找出 n 个不同的数组成的数列 $s_1, s_2, \dots, s_n$ 中最小的数, 要求输入数列 s 及数字 n , 输出最小数在数列中的位置。

4. 如何能让例 0.1.1 中的算法具有更强的通用性?

5. 看看电话簿上拨打长途电话的操作步骤, 这个打长途电话的算法具有算法的 6 个特征(准确度、唯一性、有穷性、输入、输出、通用性)中的哪几个特性? 缺少哪几个特性?

0.2 计算机系统

计算机系统是指组成计算机的各个部件, 分为硬件系统和软件系统。硬件系统指能完成执行指令、存放数据、打印数据等操作的物理设备。软件系统指计算机程序, 这些程序能被硬件系统执行, 实现一定的算法。

根据计算机的规模可将计算机分为以下几类:

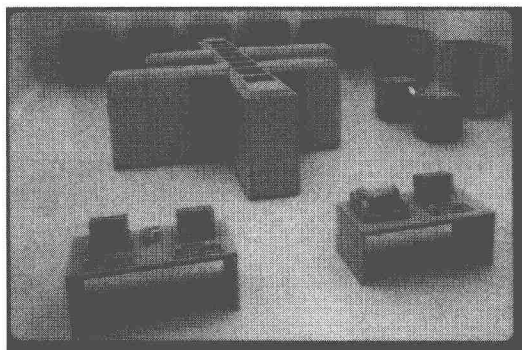


图 0.2.1 IBM 3090 大型机 (感谢 IBM 公司提供图片)

- 大型机和小型机: 大型机一般由几个人一起操作 (见图 0.2.1)。大型机通常比小型机大, 但近年来大型机与小型机的界限变得模糊了。著名的大型机制造商是 DEC 公司和 IBM 公司, 如 IBM 3090 和 DEC VAX 系统都是大型机系统。
- 工作站: 强大的单用户桌面系统。在众多的工作站制造商中, 表现最优秀的是 IBM 和 HP。

- 个人计算机: 与工作站类似的单用户系统, 它在性能方面不及工作站, 但价格低。

IBM PC 和 Apple Macintosh 是个人计算机的代表。

硬件系统中最重要部件是内存和中央处理器 (CPU), 如图 0.2.2 和图 0.2.3 所示。内存用于存放机器指令和数据, CPU 可使用内存中存放的指令和数据。CPU 主要由控制器、运算器和寄存器组 3 个部分组成。

寄存器组由本地寄存器单元组成。因为这些寄存器与 CPU 集成在一起, 控制器和运算器可以非常方便地利用它们存放计算得到的中间结果, 所以也将这些寄存器称为本地存储

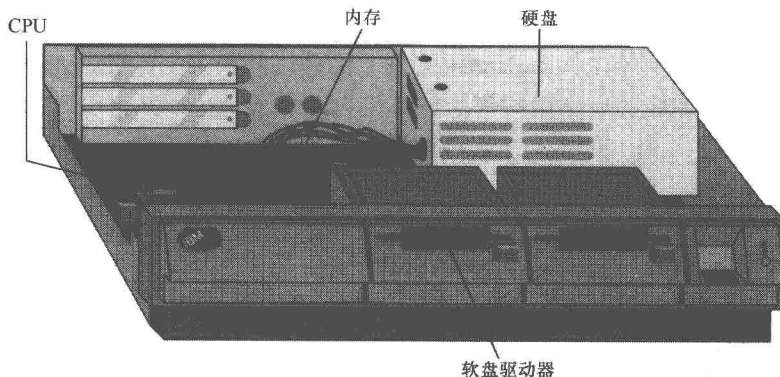


图 0.2.2 PC 机内部结构