

彻底研究

C++程序设计

张耀仁 著



- ◎ 本书内容深入浅出，为学习C++程序设计的读者打下坚实的基础
- ◎ 叙述主线明确，读者按书中内容按部就班地学习，就能达到良好的学习效果
- ◎ 书中包含近200个经过测试的完整范例程序
- ◎ 内容不仅涵盖了最基本的语法，也深入探讨了面向对象的主要精髓
- ◎ 随书配盘内含书中所有的范例程序和Borland公司的C++编译器

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

C++ 程序设计彻底研究。

张耀仁 著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

北京市版权局著作合同登记 图字：01-2005-1868 号

版 权 声 明

本书为台湾基崙资讯股份有限公司独家授权的中文简体字版本。本书专有出版权属中国铁道出版社所有。在没有得到本书原版出版者和本书出版者书面许可时，任何单位和个人不得擅自摘抄、复制本书的一部分或全部并以任何方式（包括资料和出版物）进行传播。本书原版版权属基崙资讯股份有限公司。版权所有，侵权必究。

图书在版编目（CIP）数据

C++程序设计彻底研究/张耀仁著. —北京：中国铁道出版社，
2006.5

ISBN 7-113-07070-1

I. C… II. 张… III. C 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字（2006）第 048569 号

书 名：C++程序设计彻底研究

作 者：张耀仁

出版发行：中国铁道出版社（100054，北京市宣武区右安门西街8号）

策划编辑：严晓舟 郭毅鹏

责任编辑：苏 茜 郭毅鹏 黄园园

责任校对：李 旸

封面制作：白 雪

印 刷：北京鑫正大印刷有限公司

开 本：787×1092 1/16 印张：37 字数：879 千

版 本：2006年7月第1版 2006年7月第1次印刷

印 数：1~5 000 册

书 号：ISBN 7-113-07070-1/TP·1820

定 价：58.00 元（配盘）

版权所有 侵权必究

凡购买铁道版的图书，如有缺页、倒页、脱页者，请与本社计算机图书批销部调换。

出版说明

本书包含所有重要的有关 C++ 程序设计的知识，除了入门的基础知识之外，对较深入的内容也作了讲解，例如对 VPTR 和 VTABLE 都有精彩的说明。本书提供了极佳的学习步调和连贯的先后次序，叙述方式主线明显，使读者不会为枝节所扰而混淆了学习主线，以达到良好的学习效果。

随书配盘含有书内所有的范例程序源代码、已编译过的执行文件，以及 Borland 公司符合 ANSI/ISO 标准的最新 C++ 编译器“the Free Borland C++ Builder Compiler”，非常便于读者学习使用。

本书内容不仅涵盖了最基本的语法，也深入探讨了面向对象的主要思想，可为 C++ 程序设计语言的学习打下坚实基础，不仅适合程序设计语言的初学者，也适合使用 C++ 程序设计语言开发应用程序的工程师。

本书由台湾碁峯资讯股份有限公司提供版权，经中国铁道出版社计算机图书中心审选，由任颂华同志完成本书的整稿工作。

中国铁道出版社

2006 年 5 月

前言

C++是第一个被广泛使用的面向对象程序设计语言 (Object-Oriented Programming Language), 有许多研究机构和公司都采用 C++ 来开发各种用于科学计算和商用的大型应用程序。本书内容不仅涵盖了最基本的语法, 也深入探讨了面向对象的主要思想, 可为 C++ 程序设计语言的学习打下坚实基础。本书不仅适合程序设计语言的初学者, 也适合使用 C++ 程序设计语言开发应用软件的工程师。

本书的编写动机在于, 提供一本简单、易懂并真正对实际应用有所帮助的学习用书。虽然目前已有为数不少关于 C++ 的中文著作和翻译书籍, 但这些书籍在叙述的方式以及内容的取舍上, 并不完全符合实际的需要。让许多读者半途而废, 甚至在 C++ 开发环境下放弃 C++ 的强大功能, 继续延用较容易理解的 C 语言。也有读者虽然勉强读完, 却发现说明和举例过于简单、避重就轻, 除了书上的例题外, 仍然举步维艰, 无法用来解决自己实际面对的问题。

为了符合读者的需要, 本书采取简明易懂的叙述方式, 并通过精心安排的大量例题, 使每学完一章都可以编写出相应的程序。例如, 如何避免语法和语义的错误, 如何使用预处理指令, 如何产生随机数, 如何估计程序运算所花费的时间, 如何从现有文件读取数据, 如何将执行结果存盘, 如何使用对象来仿真实际的互动关系等常遇到的编程问题, 都可在本书内找到答案。本书配盘内附超过 180 个完整的范例程序, 全部经过符合最新 ANSI/ISO 标准的 C++ 编译器测试, 并能正确执行。

除了完整的范例程序以外, 对于程序的运行机制, 比如函数间互相调用的详细过程、声明 (Declaration) 和定义 (Definition) 的区别、作为多态 (Polymorphism) 基础的后期绑定 (Late Binding)、虚拟函数、VPTR 以及 VTABLE 等主要的概念都有清楚的说明。

C++ 的特性经常使得同一个功能至少有 3 个以上的正确写法, 这全由个人的使用习惯和熟悉程度而决定。本书的特点之一, 即逐步介绍各种编写作风格。此外, 在开发一个庞大的应用程序时, 基于降低开发难度, 提高再利用率, 利于今后程序的修订与维护, 以及使结构合理化等因素, 实际上无法将全部程序编写成一个文件。本书将详细说明如何将程序分为许多小文件来共同解决问题的技巧。

中文科技图书中的英文名词翻译经常造成学习上的困难。由于 C++ 语言的主体就是由英文词汇或是其缩写所构成的, 强加翻译常造成阅读的困难, 也不利于程序的编写。因此, 对于 C++ 的关键字在本文中尽量中英文并列, 以便于学习。书末附录中列出 C++ 的关键字、C++ 运算符、标准链接库的常用头文件、函数的参数传递格式、重要名词中英对照表, 以及参考网址, 便于读者的学习。

本书共分 4 篇: 第 1 篇“C++ 程序语言基础”, 介绍计算机的基本结构、C++ 的基本语法、数据类型和使用环境、分支和循环处理两种程序流程控制, 以及函数和数组等内容。学习完这一篇, 就可以使用过程程序设计 (Procedural Programming) 来处理许多问题。

第 2 篇“高级 C++ 程序设计语言”, 介绍指针、字符串、函数的高级应用、预处理指令、数据流与文件的存取、输出格式、程序计时、struct 与数据结构、命名空间以及异常处理等内容。学习完这一篇, 就可以具备使用文件存取数据, 自由设置数据格式, 并将大型程序分为

许多小文件的能力。在第 9 章“字符串”中，有一个关于编码的程序，读者可以利用它把电子邮件转成只有接受方拥有破译码才能理解的文字，便于机密文件的传递。

第 3 篇“面向对象程序设计”，讨论类与对象、组合与继承、多态与虚拟函数、运算符重载、模板类和泛型程序设计等内容，以循序渐进的方式介绍封装（Encapsulation）、继承（Inheritance）和多态（Polymorphism）这 3 种面向对象语言的主要技术。在这一篇中，我们借助许多有意义的范例程序讲解了向上类型转换（upcast）、抽象化、派生类所定义的对象的结构和析构次序、混合组合和继承以建立新的类、重载虚拟函数、虚拟析构函数等被大部分介绍 C++ 的书籍所忽略的重要内容。在第 21 章中，通过复数演算（特别是交流电路阻抗的计算）的实例展示了“运算符重载”在简化程序上的强大功能。

第 4 篇“数值运算的应用”中，使用 C++ 以实际的范例来处理最佳化问题和常微分方程式的数值解。许多用户到目前为止都没有使用 C++ 面向对象的功能来处理数值问题的经验。对于需要耗费大量时间的数值问题，最通常的情况是使用 C、FORTRAN 或是 MATLAB。我们将在这两章中分别通过 simplex 最佳化计算以及常微分方程式的数值解，充分显示 C++ 不仅具有 MATLAB 的方便性，同时具有 C 和 FORTRAN 的高效率。

本书所附的光盘中含有书内所有范例程序的源代码，以章节为编号分别存放在易于查找的文件夹中。这些源代码可以自由修改编译，以符合自己的需要。除了原始程序代码外，光盘中还附有已编译过的执行文件，可以在个人计算机上即刻体验执行的情况。此外，光盘内还附有 Borland 公司符合 ANSI/ISO 标准的最新 C++ 编译器“the Free Borland C++ Builder Compiler”，只要有一台个人计算机，就可在不需添购任何设备的情况下，立即展开 C++ 的学习之旅。

由于 Microsoft Visual C++ .NET 的推出，本书作了相当程度的修改，务求程序在新环境下也能顺利运行。本书这一版的完成，得力于很多热心读者的支持，例如，施佩汝、唐平波、李小兵、黄园斌、周国旗等。特别感谢王能治教授的仔细校阅，同事孙明宗教授的鼓励和讨论，以及谢天扬和博士生蔡志仁在校稿和版面上的热心协助。此外，研究生邱俊杰、杨文裳、黄裕晖、陈信志和黄铭宏，以及许建丰也都先后在校稿和绘图上提供了很多协助。

本书在教学研究之余写成，疏陋之处在所难免，读者有任何指正，期望通过电子邮件与我联络，以便再版时更正。E-mail 地址：zen@mail.cgu.edu.tw。

张耀仁

目录

PART I C++程序语言基础

第 1 章 基本概念	3
1.1 计算机的发明与演进	4
1.2 计算机的基本结构	5
1.3 冯·诺依曼结构	10
1.4 计算机内部的数据表达方式	12
1.5 软件	21
1.6 程序语言 (Programming Languages)	22
1.7 C++程序语言	25
1.8 程序语言的演变	27
1.9 本章重点	29
1.10 本章练习	30
第 2 章 C++的基本语法和使用环境	31
2.1 基本程序开发步骤	32
2.2 第一个完整的 C++程序	34
2.3 延迟 DOS 窗口界面自动关闭的方法	38
2.4 Borland C++编译器的取得和安装使用	39
2.5 Visual C++.NET 程序开发步骤	43
2.6 第二个 C++程序	49
2.7 C++标识符的命名规则	52
2.8 本章重点	53
2.9 本章练习	54
第 3 章 基本数据类型	55
3.1 整数和浮点数	56
3.2 变量和常量	59
3.3 算术运算	62
3.4 标准数学函数的运算	68
3.5 逻辑值及其运算	70
3.6 字符与字符串	73
3.7 位处理运算	75
3.8 常犯的错误	79
3.9 本章练习	80
第 4 章 分支	83
4.1 算法的描述方式	84
4.2 变量的适用范围	86
4.3 if-else 语句	86

4.4	嵌套 if-else 语句	89
4.5	switch 语句	94
4.6	条件运算符	97
4.7	goto 无条件跳转语句	99
4.8	常犯的错误	101
4.9	本章重点	102
4.10	本章练习	103
第 5 章	循环	105
5.1	循环指令的种类	106
5.2	while 循环	106
5.3	continue 和 break	107
5.4	do-while 循环	108
5.5	for 循环	110
5.6	嵌套循环	115
5.7	常犯的错误	116
5.8	本章重点	117
5.9	本章练习	117
第 6 章	函数	119
6.1	函数的基本概念	120
6.2	以引用的方式调用	127
6.3	inline 函数	131
6.4	变量的适用范围和生存期间	132
6.5	常犯的错误	140
6.6	本章重点	140
6.7	本章练习	141
第 7 章	数组	143
7.1	一维数组	144
7.2	将数组当成函数的参数	146
7.3	二维数组	149
7.4	将二维数组当成函数的参数	150
7.5	常犯的错误	152
7.6	本章重点	153
7.7	本章练习	154

PART II 高级 C++ 程序设计语言

第 8 章	指针	159
8.1	内存地址与指针	160
8.2	指针与引用	164
8.3	数组与指针的代数计算	165
8.4	指针参数	169

8.5	函数指针	172
8.6	动态内存分配	174
8.7	常犯的错误	183
8.8	本章重点	185
8.9	本章练习	186
第 9 章	字符串	189
9.1	字符串的基本概念	190
9.2	字符串的输入与输出	192
9.3	字符串的处理	194
9.4	字符串的指针数组	200
9.5	字符串处理在编码上的应用	201
9.6	常犯的错误	203
9.7	本章重点	203
9.8	本章练习	204
第 10 章	函数的高级应用	207
10.1	函数的重载	208
10.2	参数的默认值	209
10.3	模板函数	210
10.4	随机数的取得	211
10.5	递归函数	216
10.6	排序与搜索	218
10.7	常犯的错误	225
10.8	本章重点	225
10.9	本章练习	226
第 11 章	预处理命令	227
11.1	预处理器	228
11.2	使用#define 进行文字取代	228
11.3	使用#define 设置宏命令	231
11.4	条件编译	234
11.5	其他与编译器有关的预处理命令	238
11.6	常犯的错误	239
11.7	本章重点	239
11.8	本章练习	240
第 12 章	数据流与文件的存取	241
12.1	数据流	242
12.2	文件的存取	244
12.3	文件的存取模式	247
12.4	数据的读取与写入	250
12.5	文件内容的位置标记	253
12.6	将文件的存取写成函数	256

12.7	常犯的错误	262
12.8	本章重点	263
12.9	本章练习	263
第 13 章	输出格式	265
13.1	使用格式操作符设置输出格式	266
13.2	输出格式设置间的交互作用	269
13.3	3 种格式设置语法的比较	271
13.4	文件存储格式的设置	272
13.5	矩阵和向量间的操作	273
13.6	常犯的错误	276
13.7	本章重点	276
13.8	本章练习	277
第 14 章	程序计时	279
14.1	程序的基本计时方法	280
14.2	更精确的程序计时方法	281
14.3	常犯的错误	283
14.4	本章重点	283
14.5	本章练习	284
第 15 章	struct 与数据结构	285
15.1	struct 的声明和使用	286
15.2	struct 构成的数组	288
15.3	struct 数据类型与函数参数的传递	290
15.4	struct 实例的动态声明	294
15.5	指针成员与数据结构	295
15.6	union 数据类型	299
15.7	enum 数据类型	300
15.8	常犯的错误	302
15.9	本章重点	302
15.10	本章练习	303
第 16 章	命名空间	305
16.1	因为名称相同而造成的问题	306
16.2	命名空间的基本语法	307
16.3	命名空间成员的存取	308
16.4	使用“using 指令”和“using 声明”以存取命名空间的成员	312
16.5	标准命名空间	316
16.6	未命名的命名空间	317
16.7	常犯的错误	318
16.8	本章重点	319
16.9	本章练习	319

第 17 章	异常处理	321
17.1	异常及其特性	322
17.2	异常处理的基本语法	323
17.3	异常的处理过程	325
17.4	抛出 <code>enum</code> 实例作为异常对象	327
17.5	抛出类所定义的对象	330
17.6	常犯的错误	333
17.7	本章重点	334
17.8	本章练习	334

PART III 面向对象程序设计

第 18 章	类与对象	337
18.1	程序设计方法的演进	338
18.2	抽象化和数据的隐藏	339
18.3	对象与类的关系	340
18.4	以对象为基础的银行账户操作程序范例	345
18.5	以对象为基础的电梯操作仿真范例	351
18.6	友元函数	356
18.7	常犯的错误	360
18.8	本章重点	360
18.9	本章练习	361
第 19 章	组合与继承	363
19.1	既有类的再利用	364
19.2	组合 (Composition)	364
19.3	组合对象的构造函数和析构函数	367
19.4	继承 (Inheritance)	369
19.5	protected 成员	374
19.6	派生类所定义的对象的结构和析构次序	374
19.7	混合组合和继承以建立新的类	376
19.8	常犯的错误	380
19.9	本章重点	381
19.10	本章练习	381
第 20 章	多态与虚拟函数	383
20.1	多态的基本概念	384
20.2	后期连接与虚拟函数	389
20.3	VPTR 和 VTABLE	393
20.4	纯虚拟函数与抽象类	395
20.5	重载虚拟函数	400
20.6	虚拟析构函数	402
20.7	常犯的错误	406

20.8	本章重点	406
20.9	本章练习	407
第 21 章	运算符重载	409
21.1	运算符使用的基本概念	410
21.2	补充几个类使用上的要点	412
21.3	使用成员函数重载二元运算符	415
21.4	使用 friend 函数重载二元运算符	419
21.5	重载一元运算符	422
21.6	含有指针数据成员的类	426
21.7	等效阻抗的计算	436
21.8	常犯的错误	439
21.9	本章重点	439
21.10	本章练习	440
第 22 章	面向对象的字符串处理	443
22.1	C 风格的字符串和面向对象的 string 类	444
22.2	String 对象的定义	444
22.3	字符串的更改、清除、剪接与部分复制	445
22.4	字符串之间的查找和比较	448
22.5	字符串对象与 C-style 字符串的互换	451
22.6	常犯的错误	455
22.7	本章重点	455
22.8	本章练习	455
第 23 章	模板类——向量和矩阵的定义	457
23.1	向量	458
23.2	Vector 模板类	458
23.3	矩阵	468
23.4	Matrix 模板类	470
23.5	对象数组的动态创造和删除	478
23.6	常犯的错误	481
23.7	本章重点	481
23.8	本章练习	482
第 24 章	泛型程序设计简介	483
24.1	C++标准模板连接库 (STL)	484
24.2	STL 的主要内容	484
24.3	使用 STL 的 vector 容器类	487
24.4	使用 STL 处理字符串数组	491
24.5	使用 complex 容器类处理复数数据	497
24.6	常犯的错误	500
24.7	本章重点	501
24.8	本章练习	501

PART IV 数值运算的应用

第 25 章 最优化问题的求解	505
25.1 最优化问题	506
25.2 Simplex 最优化求解法	507
25.3 最优化演算的 C++ 程序结构	510
25.4 没有约束条件的最优化问题实例	520
25.5 有约束条件的三维最优化问题	529
25.6 曲线拟合问题	532
25.7 常犯的错误	536
25.8 本章重点	536
25.9 本章练习	537
第 26 章 常微分方程式的数值解	539
26.1 常微分方程式	540
26.2 使用 C++ 解初始值问题的程序结构	543
26.3 ODE 初始值问题的数值解	544
26.4 程序计算结果的输出	549
26.5 van der Pol 微分方程式的数值解	555
26.6 三阶 ODE 动态系统的数值仿真	558
26.7 常犯的错误	562
26.8 本章重点	562
26.9 本章练习	562
附录	565
附录 A C++ 的 74 个关键字	566
附录 B C++ 的运算符	567
附录 C 标准链接库的常用头文件	568
附录 D 函数的参数传递格式	571
附录 E 重要名词中英对照表	572
附录 F 重要参考网址	575
参考文献	576

PART

C++程序设计语言基础

本篇介绍计算机的基本结构、C++的基本语法和使用环境、基本数据类型、分支和重复处理两种程序流程控制语法，以及函数和数组等内容。学习完本篇内容，就可以解决程序设计中的许多相应问题。

- 第 1 章 基本概述
- 第 2 章 C++的基本语法和使用环境
- 第 3 章 基本数据类型
- 第 4 章 分支
- 第 5 章 循环
- 第 6 章 函数
- 第 7 章 数组

1

CHAPTER

基本概述

在学习程序语言之前，必须对计算机的基本结构和运行方式有充分的认识，才能够发挥计算机应有的功能。我们将在本章中讨论 C++ 不同于其他程序语言的一些特性。

- 1.1 计算机的发明和演变
- 1.2 计算机的基本结构
- 1.3 冯·诺依曼结构
- 1.4 计算机内部的数据表达方式
- 1.5 软件
- 1.6 程序语言 (Programming Languages)
- 1.7 C++ 程序语言
- 1.8 程序语言的演变
- 1.9 本章重点
- 1.10 本章练习

1.1 计算机的发明与演进

计算机于 1946 年问世,正如汽车的发明是使人的双腿延伸一样,计算机的发明事实上是对人脑智力的继承和延伸。近十年来,计算机的应用日益深入到社会的各个领域,如管理、办公自动化等。由于计算机日益向智能化发展,于是人们干脆把微型计算机称之为“电脑”了。

计算机产生的动力是人们想发明一种能进行科学计算的机器,因此称之为计算机。它一诞生,就立即成了先进生产力的代表,掀开自工业革命后的又一场新的科学技术革命。

◆ 第一代电子管计算机 (1945~1956 年)

在第二次世界大战中,美国政府寻求计算机以开发潜在的战略价值。这促进了计算机的研究与发展。1944 年霍华德·艾肯 (1900~1973 年) 研制出全电子计算器,为美国海军绘制弹道图。这台简称 Mark I。

1946 年 2 月 14 日,标志现代计算机诞生的 ENIAC (The Electronic Numerical Integrator And Computer) 在费城公之于众。ENIAC 代表了计算机发展史上的里程碑,它通过不同部分之间的重新接线编程,还拥有并行计算能力。ENIAC 由美国政府和宾夕法尼亚大学合作开发,其运算速度比 Mark I 快 1 000 倍,ENIAC 是第一台普通用途计算机。

40 年代中期,冯·诺依曼 (1903~1957 年) 参加了宾夕法尼亚大学的小组,1945 年设计电子离散可变自动计算机 EDVAC (Electronic Discrete Variable Automatic Computer),将程序和数据以相同的格式一起存储在存储器中。这使得计算机可以在任意点暂停或继续工作,机器结构的关键部分是中央处理器,它使计算机所有功能通过单一的资源统一起来。

第一代计算机的特点是操作指令是为特定任务而编制的,每种机器有各自不同的机器语言,功能受到限制,速度也慢。另一个明显特征是使用真空电子管和磁鼓存储数据。

◆ 第二代晶体管计算机 (1956~1964 年)

1948 年,晶体管的发明大大促进了计算机的发展,晶体管代替了体积庞大的电子管,电子设备的体积不断减小。1956 年,晶体管在计算机中使用,晶体管和磁芯存储器导致了第二代计算机的产生。第二代计算机体积小、速度快、功耗低、性能更稳定。首先使用晶体管技术的是早期的超级计算机,主要用于原子科学的大量数据处理,这些机器价格昂贵,生产数量极少。

1960 年,出现了一些成功地用在商业领域、大学和政府部门的第二代计算机。第二代计算机用晶体管代替电子管,还有现代计算机的一些部件:打印机、磁带、磁盘、内存、操作系统等。计算机中存储的程序使得计算机有很好的适应性,可以更有效地用于商业用途。在这一时期出现了更高级的 COBOL (Common Business-Oriented Language) 和 FORTRAN (Formula Translator) 等语言,以单词、语句和数学公式代替了二进制机器码,使计算机编程更容易。新的职业,如程序员、分析员和计算机系统专家,与整个软件产业由此诞生。