



新编

Visual C++ 6.0

教程

牢固打造程序设计基础
轻松开启程序设计之门

北京科海电子出版社

<http://www.khp.com.cn>

新编电脑系列教程丛书

新编 Visual C++ 6.0 教程

北京科海培训中心 编著

北京科海电子出版社

<http://www.khp.com.cn>

内 容 提 要

读者对象:

- ☒ 想学编程但不知从何开始的用户
- ☒ 没有任何编程基础的用户
- ☒ 希望快速掌握
Visual C++ 6.0 编程的用户

达成目标:

- ☒ 领略什么是面向对象程序设计
- ☒ 直接学习面向对象程序设计的方法
- ☒ 掌握 Visual C++ 6.0 编程的核心知识

本书特点:

- ☒ 简单易懂, 快速入门
- ☒ 强调概念, 突出重点
- ☒ 深入浅出, 循序渐进

Visual C++ 是 Microsoft 公司综合性最强、最复杂的开发工具。本书从最基本的概念出发, 详细地讲解了如何使用 Visual C++ 6.0 进行编程的原理和方法, 并提供了丰富的、实用性很强的程序实例。

全书内容分为上下两篇, 上篇介绍面向对象的概念和 C++ 语言的程序设计方法; 下篇着重讲述了利用 Visual C++ 的集成开发环境设计应用程序的原理和过程, 包括集成环境的安装、应用程序框架、AppWizard 和 ClassWizard 的使用和结构原理、对话框和控制、绘图和打印以及 Visual C++ 对数据库的访问等内容。

本书结构清晰, 重点突出, 实例丰富, 是初学者很好的入门教程, 又可作为 Visual C++ 的培训教材使用。

品 名: 新编 Visual C++ 6.0 教程
作 者: 北京科海培训中心
责任编辑: 周 洁
排 版: 吴文娟
出 品: 北京科海电子出版社
印 刷 者: 北京市朝阳区科普印刷厂印刷
发 行: 新华书店总店北京科技发行所
开 本: 787×1092 1/16 印张: 24.25 字数: 388 千字
版 次: 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷
印 数: 0001~5000
盘 号: ISBN 7-900107-13-4
定 价: 28.00 元 (1CD)

丛书序

今天，计算机技术发生了日新月异的变化，它已全面渗透到全社会的各行各业。熟练地使用计算机已经成为现代人必备的一项技能，如何把握时代的脉搏，学会最新、最流行的电脑知识，使自己能够利用现代科技解决问题呢？

科海培训中心作为全方位电脑教育服务提供商，向广大初学者奉献上一套新编教材丛书。该丛书具有如下特点：

选材精良

该丛书精选了国内外著名软件公司最新的流行产品，包括操作系统、办公应用、程序设计、多媒体、互联网络等方方面面，对于任何一位想学习计算机应用的读者来说总有一本是适合的。每本新编教材所介绍的知识并不追求面面俱到，而是选择用户常用的功能进行介绍，以使广大初学者尽快上手。

内容设计精致

该丛书根据用户实际使用的需求，着重讲述各种电脑技能最基本、最常用的内容，语言简炼，操作步骤清晰。该丛书并没有简单地罗列知识点，在介绍完基本概念与理论后，举出相应的实例以帮助读者理解，具有通俗易懂、易于掌握、实用性强和容易上手等特点。大部分章节后都有习题，读者在阅读完每章后，通过习题可以总结回顾，掌握各章的知识要点。

版式新颖

该丛书新颖的版式使读者感觉轻松、活泼，不再有埋于书海中苦苦寻找的感觉，让读者耳目一新。

读者定位明确

本书的宗旨是以精炼的篇幅，让初级用户及广大电脑爱好者尽快掌握流行电脑技能的使用，不管您对电脑知识是一无所知，还是一知半解，只要您找到想学习使用的相关新编教程，就能帮助您简单、快捷、生动、方便地学习。

本丛书中的部分内容曾作为独立的培训教材受到读者的好评，同时许多读者也提出了中肯的意见，此次我们对大部分内容进行了修订，希望这套丛书能让您在学习计算机技术的道路上少走弯路，捷足先登，但不足之处在所难免，希望广大读者给予批评与建议，来信请寄：

cliff@khp.com.cn

丛书编委会

主编：北京科海培训中心

编委：何洪学 王晓玲 周明涛 王 健 王腾蛟
刘云峰 凌宇欣 赵华刚 肖国超 王 琪
陈堂安 吴怀宇 孙继青 于 丽 张丽英
刘 刚 越 辉 石 磊 郝春强

目 录

上篇 C++程序设计基础

第 1 章 C++入门.....	3	3.3.5 循环嵌套.....	36
1.1 C++的发展	3	第 4 章 数组.....	38
1.2 程序设计语言	4	4.1 一维数组.....	38
1.3 面向对象程序设计.....	5	4.1.1 一维数组的定义	38
1.4 简单的 C++程序实例.....	7	4.1.2 一维数组的初始化	39
1.4.1 简单程序的源代码	7	4.1.3 一维数组的引用	39
1.4.2 程序的编辑、编译和运行 ...	9	4.1.4 一维数组的程序举例	40
1.5 小结	13	4.2 多维数组.....	42
第 2 章 基本数据类型与表达式.....	14	4.2.1 多维数组的定义	42
2.1 标识符.....	14	4.2.2 多维数组的初始化	43
2.2 基本数据类型.....	15	4.2.3 多维数组的引用	44
2.3 常量	17	4.2.4 多维数组程序举例	45
2.4 变量	19	4.3 字符数组与字符串的处理	46
2.5 表达式.....	20	4.3.1 字符数组的定义和初始化 ..	46
2.6 小结	23	4.3.2 字符数组的输入输出	47
2.7 练习题.....	23	4.3.3 字符串	48
第 3 章 流程控制	25	4.3.4 字符数组程序举例	51
3.1 条件语句.....	25	4.4 小结	53
3.2 转向语句.....	28	4.5 练习题.....	53
3.3 循环语句.....	30	第 5 章 函数.....	55
3.3.1 while 语句.....	30	5.1 函数的定义.....	55
3.3.2 do-while 语句.....	31	5.1.1 无参函数的定义	55
3.3.3 for 语句	32	5.1.2 有参函数的定义	56
3.3.4 if 语句和 goto 语句构成的		5.2 函数的调用.....	57
循环语句.....	35	5.3 局部变量和全局变量	59
		5.3.1 局部变量	59

5.3.2 全局变量	60	7.1 类的定义和对象的生成	79
5.4 函数程序实例	60	7.2 成员函数和 this 指针	82
5.5 小结	62	7.3 构造函数	83
5.6 练习题	63	7.4 析构函数	84
第 6 章 指针与链表	65	7.5 小结	86
6.1 指针的基本概念	65	7.6 练习题	86
6.2 指针变量的定义	66	第 8 章 继承与多态	87
6.3 指针变量的引用	66	8.1 类的继承	87
6.4 结构体	69	8.2 单一继承	88
6.4.1 结构体的定义	69	8.3 多重继承	90
6.4.2 结构体对象	70	8.4 多态性和虚函数	92
6.4.3 结构体对象的引用	71	8.4.1 函数的重载	92
6.5 链表	71	8.4.2 运算符的重载	94
6.5.1 链表的建立	72	8.4.3 虚函数	94
6.5.2 链表的插入与排序	74	8.5 类的友元	98
6.5.3 链表的删除	75	8.5.1 类的友元函数	98
6.6 小结	78	8.5.2 类的友元类	99
6.7 练习题	78	8.6 小结	99
第 7 章 类与对象	79	8.7 练习题	100

下篇 Visual C++程序设计

第 9 章 Visual C++集成开发环境..	103	9.4 小结	109
9.1 Visual C++工程及其文件构成	103	9.5 练习题	110
9.2 启动 Visual C++	104	第 10 章 开发简单的 Visual C++程序	111
9.3 Visual C++集成开发环境概述	105	10.1 AppWizard 的使用	111
9.3.1 项目工作区	105	10.2 一个简单的应用程序	118
9.3.2 应用程序向导——AppWizard	107	10.3 程序结构剖析	123
9.3.3 类向导——ClassWizard	108	10.4 Win32 编程基础	125
9.3.4 向导工具栏——WizardBar	108	10.4.1 Win32 数据类型	125
		10.4.2 句柄	126
		10.4.3 标识符命名	126

10.5 小结	127	13.6 练习题.....	189
10.6 练习题.....	128	第 14 章 对话框	193
第 11 章 应用程序框架与 文档/视图结构.....	129	14.1 消息映射.....	193
11.1 CPROGRAMApp 类.....	129	14.2 定义对话框.....	197
11.2 CMainFrame 类.....	134	14.3 增加对话框的功能.....	201
11.3 CPROGRAMDoc 类、CPROGRAMView 类与文档/视图结构	137	14.4 弹出对话框.....	206
11.4 CChildFrame 类	146	14.5 运行程序.....	208
11.5 应用程序框架中的文件列表	148	14.6 小结	209
11.6 小结.....	149	14.7 练习题.....	210
11.7 练习题.....	149	第 15 章 控件.....	212
第 12 章 Microsoft 类库基础	150	15.1 控件概述.....	212
12.1 Microsoft 类库概述	150	15.2 静态控件.....	215
12.2 根类: CObject	151	15.3 按钮控件.....	218
12.3 MFC 应用程序框架结构类....	151	15.4 编辑框控件.....	223
12.4 MFC 窗口类	155	15.5 列表框控件.....	226
12.5 MFC 异常类	162	15.6 组合框控件.....	233
12.6 MFC 文件类	163	15.7 滚动条控件.....	237
12.7 绘图和打印类.....	164	15.8 常用控件应用实例.....	239
12.8 ODBC 类	166	15.8.1 对话框和控件的 手工编辑	241
12.9 小结	167	15.8.2 生成对话框类	244
12.10 练习题.....	167	15.8.3 为控件建立关联的 成员变量.....	245
第 13 章 菜单、ClassWizard 与 消息映射.....	168	15.8.4 手工加入其他成员变量	249
13.1 编辑菜单资源.....	168	15.8.5 成员变量的初始化	252
13.1.1 系统生成的菜单	168	15.8.6 建立消息映射与 响应函数.....	253
13.1.2 菜单的编辑	170	15.8.7 扩充响应函数	257
13.2 ClassWizard 简介	171	15.8.8 程序的完成	259
13.3 建立消息映射	174	15.9 小结	261
13.4 增强菜单项.....	179	15.10 练习题.....	261
13.5 小结	189	第 16 章 实例分析.....	263
		16.1 应用程序功能设计	263

16.2	编辑对话框.....	264	18.1	绘图和打印类.....	326
16.3	建立成员变量.....	268	18.2	绘图操作的实现过程.....	330
16.4	建立菜单项.....	273	18.3	坐标与坐标模式.....	334
16.5	建立成员函数.....	277	18.4	绘图与打印应用实例.....	340
16.6	小结	289	18.5	小结	346
16.7	练习题.....	290	18.6	练习题.....	346
第 17 章	高级控件	291	第 19 章	访问数据库	349
17.1	高级控件简介	291	19.1	MFC 提供的数据库访问类....	349
17.2	标签控件.....	292	19.2	建立、连接数据源.....	352
17.3	树控件.....	311	19.3	建立访问数据库的应用程序	359
17.4	旋转按钮控件.....	316	19.4	实现数据访问.....	366
17.5	滑动条控件.....	319	19.5	增加和删除记录.....	369
17.6	进度条控件.....	322	19.6	程序分析.....	372
17.7	小结	324	19.7	小结	380
17.8	练习题.....	325	19.8	练习题.....	380
第 18 章	绘图和打印	326			

上篇

C++程序设计基础

- 📖 第1章 C++入门
- 📖 第2章 基本数据类型与表达式
- 📖 第3章 流程控制
- 📖 第4章 数组
- 📖 第5章 函数
- 📖 第6章 指针与链表
- 📖 第7章 类与对象
- 📖 第8章 继承与多态

原书空白页

第 1 章 C++入门

C++是一门优秀的面向对象程序设计语言。C++在 C 语言的基础上进行了改进和扩充，增加了面向对象程序设计的功能，更适合编制复杂的大型软件系统。C++比 C 语言更容易被人们理解和掌握，独特的语言机制使它在计算机研究和应用领域得到了广泛应用。本章我们将引入面向对象程序设计的概念，并通过简单的 C++程序来加深用户对面向对象程序设计的理解。

1.1 C++的发展

C 语言是美国贝尔实验室的 Dennis Ritchie 于 1972 年完成的，最早用来编写 UNIX 操作系统。随着 UNIX 操作系统的成功，C 语言广为人们所认识。C 语言的快速推广导致了 C 语言各种版本产生，但这些版本通常不兼容，这对希望开发出能够在多种平台上运行的代码的程序员来讲，是一个严重的问题。显然，人们希望出现标准的 C 语言版本。1989 年，美国国家标准协会指定了 C 语言的标准版本（ANSI C）后，C 语言发展更为迅速。

C 语言的特点是简洁灵活，运行效率高，可移植性好，但同时也具有它的局限性。比如它的检查机制较弱，本身几乎没有支持代码的重用结构，程序员很难控制较复杂的程序。1980 年，贝尔实验室的 Bjarne Stroustrup 对 C 进行了改进和扩充，1983 年正式把取得的成果称为 C++。在经过 3 次 C++的修订之后，1994 年制定了 ANSI C++标准草案，成为目前的 C++。C++继承了 C 语言的全部特征、属性和优点，增加了对面向对象程序设计的完全支持。目前，C++仍然在不断地进行发展和完善。

C++迅速得到了广大程序员的喜爱，很多著名的软件厂商都推出了 C++语言

的开发环境。在 Windows 操作系统下, Visual C++是 Microsoft 公司推出的 C++语言开发平台。Visual C++的特点在于它的语法符合 ANSI C++的标准, 并集成了 MFC(Microsoft Foundation Class,Microsoft 基本类库)。Visual C++提供的 MFC AppWizard 可以很容易地生成程序框架, 它提供的 MFC ClassWizard 可以帮助程序员轻松完成各种 MFC 类的使用和维护。如果人们从一开始就使用 Visual C++作为编程环境, 除能掌握 C++本身外, 更能为将来彻底掌握 Visual C++和 MFC 编程打下基础。

1.2 程序设计语言

程序

程序是以某种语言为工具编制出来的动作序列, 它表达了人们的思想。计算机程序是按照计算机程序设计语言的规范而编写的一系列动作, 它表达了程序员要求计算机执行的操作。

对计算机来说, 一组机器指令就是程序。人们常说的机器代码或机器指令, 是按照计算机硬件设计规范的要求, 编制出来的动作序列。

对计算机用户来说, 程序员用程序设计语言编写的语句序列也叫做程序。程序用文件的形式来保存, 源程序、源代码都是程序。

计算机要运行需要程序, 如果人们想利用计算机解决各种问题, 最重要的就是编写计算机程序。对程序的定义在学术界有严格的规定, 这里只作简单的表述。

程序设计语言

计算机程序设计语言有 3 种: 机器语言、汇编语言、高级语言。

机器语言是最原始的语言, 它所包括的计算机指令由二进制数表示。

汇编语言把计算机指令用一些助记符表示, 比如 MOV、ADD 等。用汇编语言编写的程序比机器语言编写的程序更容易让人读懂。机器语言和汇编语言都是面向具体硬件的编程语言, 被人们统称为低级语言。低级语言同硬件联系紧密,

用低级语言在一种 CPU 上编写的程序不能在另一种 CPU 上运行。低级语言运行速度快，但不易编写、不容易调试，可移植性也很差。

高级语言接近人类自然语言，采用了人类语言中的某些元素来控制过程和程序结构，因此高级语言具有易学易用的优点。高级语言编写的程序必须经过编译和解释才能被机器识别，用高级语言编写的程序被称为“源程序”或“源代码”。现代应用软件的开发大部份用的是高级语言。

在早期的高级语言中，有面向初学者的 Basic 语言，主要用于科学计算的 Fortran 语言，具有数学般严谨的 Pascal 语言，高效灵活的 C 语言等。其中，C 语言的影响面最为广泛。

C 和 C++

程序设计方法从结构化程序设计向面向对象程序设计发展。在学习面向对象程序设计之前，应该具有一些过程性语言的基础，即先学习 C++ 的过程性语言部分，然后再学习类部分。从根本上来说，C 能够很好地支持结构化程序设计，C++ 能够很好地支持面向对象程序设计。C++ 的泰斗 Bjarne Stroustrup 和大部分 C++ 程序员观点一致，没有学习过 C 的人可以直接学习 C++，即直接学习面向对象的程序设计。

但 C 程序设计经验锻炼了程序员的抽象思维能力，这为学习 C++ 更为抽象的概念和技术打下了基础。如果程序员使用 C 语言进行程序设计的经验丰富，对学习 C++ 也是很有利的。

1.3 面向对象程序设计

在面向对象的程序设计方法出现之前，占据主流的是结构化程序设计方法。结构化程序设计采用功能分解、自顶向下逐步求精的设计原则。当一些任务很复杂时，程序员将它分解为一系列较小的任务，直到这些任务小到易于理解的程度。结构化的程序往往清晰、易读，但随着数据量的增大，数据与处理数据的方法的

分离使程序越来越难以理解。另外，采用结构化程序设计方法的程序员发现，每一种相对于老问题的新方法都要带来额外的开销，对于这种重复投入，他们希望能够重用已知特性的部件。

面向对象程序设计（Object-Oriented Programming，简称 OOP）应运而生，它的本质是把数据和处理数据的过程当成一个整体——对象。C++充分支持面向对象设计。面向对象程序设计需要封装和数据隐藏，需要继承和多态技术。

封装和数据隐藏

比如一个技术员在组装一台计算机时，他想要一个显卡。他不必用集成电路芯片等材料去制作一个显卡，而是直接去购买一个他所需要的某种功能的显卡。技术员关心的是显卡的功能，并不关心显卡内部的工作原理。显卡自成一体的特性称为封装性。程序员无需知道封装单元内部如何工作就能够使用的思想，人们称之为数据隐藏。

同样的道理，C++通过建立用户定义类型支持封装性和数据隐藏。定义好的类建立后，用户不需要知道它是如何工作的，只要知道它能够做什么样的工作就可以了。

继承和重用

如果想制造新的 DVD，工程师可以从设计草图开始或直接对现有的型号加以改进。当只需要对现有的型号增加一个功能时，工程师是绝对不想从头开始，而是在原有的型号上增加新的集成电路来实现新的功能。新的 DVD 被生产出来，并采用一种新的型号。这就是继承和重用。

C++采用继承支持重用的思想，程序可以在现有类的基础上声明新类型。新类型从已经有的类型派生出来，被称为派生类。新的 DVD 继承了原有 DVD 的所有属性，它是原来 DVD 的派生，同时增加了新的功能。

多态性

C++采用继承的方法构造类，并用多态性为每个类制定表现行为。例如，学生类需要统计成绩的操作。大学生继承中学生，中学生的统计成绩是语文、数学、

自然等课程的统计, 对于后继的大学生, 统计的成绩则为高等数学、普通物理、计算机等。

程序员利用 C++ 的继承性和多态性, 可以轻而易举地生成一系列类似但又独一无二对象。这些对象具有许多相似的特征, 但不同的对象又具有不同的表现形式。

1.4 简单的C++程序实例

我们先给出一个简单的计算圆的面积和周长的 C++ 程序源代码, 然后使用 Microsoft Visual C++ 6.0 来编辑、编译和运行另一个程序。

1.4.1 简单程序的源代码

下面给出程序 1.1 的源代码, 然后仔细解释每行代码的含义, 用户可以通过这个实例初步了解 C++ 的编程思想。

程序清单:

程序清单 1.1 ex1-1.cpp

```
// 计算圆的面积和周长

#include <iostream.h>
const double pi=3.14;
double calculate_area(int radius)
{
    return pi*radius*radius;
}
double calculate_perimeter(int radius)
{
    return 2*pi*radius;
}

void main()
{
    int input_radius;
    cout<<"请输入圆半径: ";
    cin>>input_radius;
    cout<<"圆面积为: "<<calculate_area(input_radius)<<"\n";
    cout<<"圆周长为: "<<calculate_perimeter(input_radius)<<"\n";
}
```

程序运行时显示:

请输入圆半径:

如果从键盘输入 3 并按回车键, 程序显示运行结果:

圆面积为: 28.26

圆周长为: 18.84

Press any key to continue

程序中, 在 “//” 之后的语句为 C++ 的注释行。注释行分为序言注释和注解性注释, 上面的例子中为序言注释, 说明程序的作用; 注解性注释用于程序中难懂的语句。程序执行时将会忽略注释行。

第 2 行的 `#include <iostream.h>` 是预处理包含指令, 其中的 `iostream.h` 是 C++ 提供的标准类库, 它提供了输入输出流 `cin` 和 `cout` 以及输入输出运算符 `>>` 和 `<<` 的定义。

第 3 行说明了一个浮点型符号常量 `pi`, 该常量的固定值为 3.14。

第 4~7 行是函数 `calculate_area` 的定义, 该函数可以根据输入参数 `radius` 计算圆的面积。

第 8~11 行是函数 `calculate_perimeter` 的定义, 它可以根据输入参数 `radius` 计算圆的周长。

第 13~20 行定义了主函数 `main()`, 其中 `main` 是函数名, 它是程序执行的入口。`Main` 由小写字母构成, C++ 对名字的大小写时敏感的, 所以要多加注意。

第 16 行是一个输出语句, `cout<<` 表示从显示器输出后面的 “请输入圆半径:” 的内容。

第 17 行是一个输入语句, `cin>>` 表示从键盘输入一个整数给变量 `input_radius`。

第 18 行是一个输出语句, 先输出字符串 “圆面积为:”, 再输出由 `calculate_area` 计算出的半径为 `input_radius` 的圆的面积, 最后输出 “\n”, 表示换行。

第 19 行也是一个输出语句, 先输出字符串 “圆周长为:”, 再输出由 `calculate_perimeter` 计算出的半径为 `input_radius` 的圆的周长, 最后输出换行符。

`cin` 称作字符流输入, `cout` 称作字符流输出, 这两个函数都在文件 `iostream.h` 中定义。运算符 `<<` 和 `>>` 的原意是表示算术左移和右移, 在 `iostream.h` 文件中被重新定义 (在面向对象中称作运算符重载), 作为输入输出运算符使用。`cin` 中使