

计算机语言技术系列丛书

VISUAL C++

甘特 孙涛 编写
陈瑞江 梁革

C / C++

编程技巧与范例

希望

学苑出版社

计算机图形与图像丛书

Visual C++

C / C++编程技巧与范例

学苑出版社

(京)新登字 151 号

内 容 提 要

本书详细系统地介绍了 Visual C++ C/C++ 程序设计技巧并且给出了大量的实例。主要分成两部分,第一部分包括的内容有: C 语言程序设计基础知识、数据结构、DOS 系统时间、外部设备的控制、计算机绘图、基本 DOS 和 BIOS 系统应用、存储器映射绘图法、ANSI.SYS 与 C 语言。第二部分包括的内容有: C++ 语言的程序设计基础知识、OOP 的知识、数据结构、虚拟函数和多态、C++ 的 I/O 流与文件管理、Microsoft 类库(MFC)、QuickWin 应用程序设计。

本书叙述清晰,通俗易懂,使用方便,是使用 Visual C++ 的用户极为有用的工具书,也是计算机应用人员和大中专院校师生必备的参考书。

需要本书者,请与北京海淀 8721 信箱书刊部联系,邮政编码: 100080, 电话: 2562329。

计算机语言技术系列丛书

Visual C++ C/C++ 编程技巧与范例

编 写: 甘 特 孙 清 陈瑞江
梁 革 张 军 晓 鸥
审 校: 沈 晓 田 霞
责任编辑: 甄国宪
出版发行: 学苑出版社 邮政编码: 100036
社 址: 北京市海淀区万寿路西街 11 号
印 刷: 施园印刷厂
开 本: 787×1092 1/16
印 张: 41.25 字数: 959 千字
印 数: 1~5000 册
版 次: 1994 年 4 月北京第 1 版第 1 次
ISBN 7-5077-0776-8/TP·8
本册定价: 49.00 元

学苑版图书印、装错误可随时退换

目 录

第一部分 C 语言程序设计技巧

第一章 C 语言基本数据处理的概念	3
1.1 变量名称的使用	3
1.2 变量的声明	4
1.3 基本数据类型	5
1.4 常数的表达方式	7
1.5 基本的算术运算	9
1.6 类型的转换	11
1.7 C 语言的特殊表达式	13
第二章 基本的输入与输出	16
2.1 printf()	16
2.2 scanf()	29
2.3 字符的输入和输出函数	31
第三章 程序的流程控制	38
3.1 关系运算符与逻辑运算符	38
3.2 for 语句	39
3.3 while 语句	45
3.4 do...while 语句	47
3.5 if 语句	50
3.6 else-if 语句	52
3.7 e1 ? e2 : e3 特殊表达式	55
3.8 switch 语句	56
3.9 break 语句	58
3.10 continue 语句	59
3.11 goto 语句	61
第四章 数组和指针	63
4.1 一维数组	63
4.2 二维数组	68
4.3 指针的基本概念	70
4.4 指针和数组的混合应用	72
第五章 函数的应用	75
5.1 函数的定义	76
5.2 主程序函数的位置	77

5.3	函数的返回值	78
5.4	数组数据的传递	80
5.5	下标的传递	84
5.6	递归式函数的调用	85
5.7	新式 ANSI 函数的声明	86
5.8	简单数学函数的使用	87
第六章	变量的等级	93
6.1	auto	93
6.2	static	94
6.3	external	97
6.4	static external	99
6.5	register	100
第七章	C 语言的预处理程序	101
7.1	宏 #define 命令	101
7.2	#include 功能	106
7.3	条件式的编译	108
第八章	用户自定义数据类型	111
8.1	typedef	111
8.2	结构数据类型	112
8.3	结构数据与数组	115
8.4	嵌套结构	116
8.5	结构数据与函数	118
8.6	union	121
8.7	enum	123
第九章	字符串的彻底剖析	125
9.1	由字符所组成的一维数组	125
9.2	字符串的应用	127
9.3	字符串与指针	131
9.4	gets()和 puts()函数	133
9.5	字符串数组	136
9.6	数组的字符串指针	137
9.7	命令行参数	139
第十章	基本位运算	141
10.1	&运算符	141
10.2	运算符	143
10.3	^运算符	144
10.4	~运算符	146
10.5	<<运算符	147
10.6	>>运算符	148

10.7	位字段	149
第十一章	测试符号与符号转换函数	152
11.1	isalnum()	152
11.2	isalpha()	153
11.3	isascii()	153
11.4	isctrl()	154
11.5	isdigit()	155
11.6	isgraph()	156
11.7	islower()	157
11.8	isprint()	158
11.9	ispunct()	159
11.10	isspace()	160
11.11	isupper()	160
11.12	isxdigit()	161
11.13	tolower()	162
11.14	toupper()	162
第十二章	文件的输入与输出	164
12.1	有缓冲区的输入与输出函数	164
12.2	C语言缺省的文件指针	182
12.3	无缓冲区的输入与输出	186
12.4	C语言缺省的文件句柄	191
第十三章	基本的序列结构	193
13.1	动态数据结构的声明	193
13.2	存储器的配置	193
13.3	链表序列的基本操作	196
13.4	序列的连接	199
13.5	序列中节点的删除	204
13.6	释放存储器空间	207
13.7	节点的插入	207
13.8	双向链表序列	208
第十四章	堆栈与队列	212
14.1	堆栈	212
14.2	队列	216
第十五章	二叉树	221
15.1	二叉树的节点结构	221
15.2	二叉树的建立	222
15.3	二叉树的打印	223
第十六章	DOS 系统时间	229
16.1	time()	229

16.2	ctime()	230
16.3	localtime()	230
16.4	asctime()	231
16.5	gmtime()	232
16.6	difftime()	232
16.7	clock()	233
16.8	随机函数	234
第十七章	认识键盘	236
17.1	__kbhit()	236
17.2	认识键盘码	237
第十八章	屏幕的控制	242
18.1	屏幕的清除	242
18.2	移动光标位置	243
18.3	设置光标外形	244
18.4	建立视区	245
18.5	卷动视区	246
18.6	动态画面的设计	248
第十九章	计算机绘图	250
19.1	了解屏幕模式	251
19.2	设置图形模式	252
19.3	__lineto()	253
19.4	__moveto()	254
19.5	__rectangle()	255
19.6	__ellipse()	256
19.7	__setpixel()	258
19.8	设计图样	259
19.9	线条样式的设置	261
19.10	动态画面的效果	262
19.11	文本与图形的结合	265
19.12	逻辑坐标	266
19.13	颜色的设置	267
19.14	视区的设定	268
第二十章	基本 DOS 和 BIOS 系统应用	271
20.1	IBM PC 内部结构	271
20.2	DOS 和 BIOS 系统调用的基本概念	275
20.3	DOS 系统调用	277
20.4	BIOS 系统调用	284
第二十一章	存储器映射绘图法	287
21.1	单色文本模式适配器	287

21.2 彩色文本模式适配器	291
第二十二章 ANSI.SYS 与 C 语言	293
22.1 光标与屏幕的控制	293
22.2 属性的控制	297
22.3 键盘的重新设置	300
第二十三章 鼠标器的控制	302
23.1 功能 0	302
23.2 功能 1	303
23.3 功能 2	304
23.4 功能 3	305
23.5 功能 4	311
23.6 功能 5	313
23.7 功能 6	317
23.8 功能 7	319
23.9 功能 8	321
23.10 功能 9	323
23.11 功能 10	325
23.12 功能 11	327
23.13 功能 15	327
第二十四章 QuickWin 应用程序设计	329
24.1 一个简单的程序实例	329
24.2 一个简单的绘图程序实例	330
24.3 About 对话方块	331
24.4 QuickWin 与 Windows 应用程序的差别	332

第二部分 C++语言程序设计技巧

第二十五章 OOP 简介	335
25.1 OOP 的特点	335
25.2 软件要求及适用对象	340
25.3 如何学习 OOP	340
25.4 结构化程序设计的缺点	340
25.5 OOP 概述	342
第二十六章 C++语言的基本概念	343
26.1 变量命名约定	343
26.2 变量的声明	344
26.3 基本数据类型	345
26.4 常数的表达	348
26.5 基本算术运算	350

26.6 类型转换	351
26.7 C++语言的特殊运算	352
第二十七章 输入/输出与用户定义数据类型	355
27.1 printf()	355
27.2 为什么 OOP 程序不使用 printf()和 scanf()	355
27.3 OOP 程序特有的输出 cout	356
27.4 OOP 程序特有的输入	360
27.5 typedef	363
27.6 结构数据类型	364
27.7 嵌套结构	370
27.8 几个有关结构的实例	372
27.9 联合(union)	373
27.10 枚举(enum)	375
第二十八章 程序流程	380
28.1 关系与逻辑操作符	380
28.2 for 语句	381
28.3 while 语句	384
28.4 do ... while 语句	388
28.5 if 条件语句	390
28.6 else if 语句	393
28.7 e1? e2 : e3 条件表达式	394
28.8 switch 开关语句	395
28.9 break 语句	397
28.10 continue 语句	398
28.11 goto 语句	399
第二十九章 函数	401
29.1 函数定义	402
29.2 主程序和函数的次序	404
29.3 函数的返回值	406
29.4 结构数据与函数	408
29.5 地址的传送	409
29.6 函数参数值的初始化	413
29.7 函数重载	414
29.8 递归函数的调用	416
29.9 inline 操作符	418
第三十章 类和对象	420
30.1 类的定义方法	420
30.2 对象概念	421
30.3 构造函数和析构函数	424

30.4	如何在构造函数内传递参数	427
30.5	函数的重载	428
30.6	函数的重载与构造函数的区别	430
30.7	成员函数的参数是对象的情形	431
30.8	成员函数的返回值是对象的情形	433
30.9	类与结构	435
30.10	inline函数	436
30.11	类、对象的内存分配	438
30.12	静态类数据	439
30.13	综合应用实例	440
第三十一章	数组	446
31.1	一维数组	446
31.2	设置一维数组的初值	448
31.3	二维数组	449
31.4	设置二维数组的初值	451
31.5	数组数据在函数内的传递	452
31.6	结构与数组	454
31.7	在对象内声明数组	455
31.8	对象数组	460
31.9	字符串	462
31.10	gets()和puts()函数	465
31.11	用cin读取字符串	467
31.12	字符串处理函数	469
31.13	字符串数组	469
31.14	字符串与类	470
31.15	综合应用实例	472
第三十二章	重载操作符	475
32.1	重载单目操作符	475
32.2	含返回值的操作符重载	478
32.3	重载双目操作符	480
32.4	重载比较操作符	484
32.5	重载特殊操作符	486
32.6	不同数据类型之间的转换	488
32.7	不同类类型之间的转换	490
32.8	重载等号操作符	495
32.9	综合应用实例	498
第三十三章	继承	501
33.1	继承的目的	502
33.2	构造函数与派生类	506

33.3	重载函数、基类与派生类	509
33.4	多重派生	512
33.5	多层派生类的概念	515
33.6	某个类继承多个类的概念	518
33.7	类的嵌套	524
33.8	私有派生类	527
33.9	私有数据的派生	528
33.10	综合应用实例	531
第三十四章	指针	537
34.1	指针	537
34.2	指针与函数	542
34.3	指针与数组	543
34.4	指针与字符串	545
34.5	指向指针的指针	546
34.6	指针与结构	550
34.7	内存分配	551
34.8	指针与类	553
34.9	指针、构造函数和析构函数	554
34.10	指针与对象数组	556
34.11	堆栈、队列与二元树	557
第三十五章	虚拟函数和多态	572
35.1	指针与派生类	572
35.2	虚拟函数	574
35.3	虚拟函数的作用	578
35.4	纯虚拟函数	581
35.5	虚拟基类	583
35.6	综合应用实例	586
第三十六章	OOP 的其它概念	589
36.1	友元函数	589
36.2	友元函数对程序语义的作用	591
36.3	友元函数与重载操作符	593
36.4	友元类	596
36.5	static 类成员	597
36.6	this	599
第三十七章	C++的 I/O 流与文件管理	602
37.1	流类的结构	602
37.2	设备操作符	603
37.3	ostream 成员函数应用实例	605
37.4	istream 成员函数应用实例	605

37.5	C++的文件管理.....	608
37.6	命令行参数	618
37.7	DOS 设备文件	619
37.8	<<与>>的重载函数	621
37.9	综合应用实例	623
第三十八章	Microsoft 类库(MFC)	626
38.1	建立面向 DOS 的类库文件	626
38.2	设置参数	627
38.3	CString 类	628
38.4	CTime 类	630
38.5	CTimeSpan 类	630
38.6	CByteArray 类	632
38.7	CWordArray 类	633
38.8	CStringArray 类	634
38.9	CStringList类	635
38.10	CObList类	637
38.11	CMapStringToOb类	639
38.12	CMapStringToString类.....	641
第三十九章	QuickWin 应用程序设计	643
39.1	一个 QuickWin 程序.....	643
39.2	一个简单的绘图程序	644
39.3	__wabout().....	645
39.4	不同类型的窗口	645

第一部分

C 语言程序设计技巧

第一章 C 语言基本数据处理的概念

对程序语言而言，最基本的数据处理对象就是变量和常数，在本章我们将对所有的变量和常数做一解释。另外，C 语言程序拥有许多不同于其它高级语言的表达式，本章也将一一说明。

1.1 变量名称的使用

C 语言对变量名称的使用是有一些限制的，它只能以下列三种字符做开头：

1. 大写字母。
2. 小写字母。
3. 下划线()。

至于变量名称的组成则是由下列四种字符所构成：

1. 大写字母
2. 小写字母
3. 底线()
4. 阿拉伯数字 0~9

范例：下列均是合法的变量名称：

SUM, hung, sum_1, _rg, x5, y61

范例：下列均是不合法的变量名称：

sum, 1 ← 变量名称不可有“,”符号
3y ← 变量名称不可由阿拉伯数字开头
x\$2 ← 变量名称不可含有“\$”符号

另外，有一点要注意的是，在 C 语言中大写字母和小写字母代表不同的变量。

范例：下列三个字符串，分别代表三个不同的变量。

sum
Sum
SUM

有关变量使用的另一点要注意的是，有些字为系统保留字(又称关键字)，因这些字在 C 编译程序中代表特别意义，所以不能使用这些字为变量名称。图 1.1 是 ANSI C 语言的保留字。

auto	break	case	char	const	continue
default	do	double	else	cenum	extern
float	for	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while				

图 1.1 ANSI C 语言的保留字

此外，在 Visual C++ 的 C 软件中，为了使 C 语言程序设计员能很方便存取 DOS 系统资源，又扩充了一些字为保留字，如图 1.2 所示。

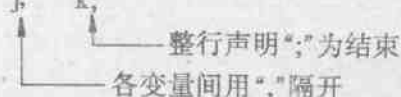
__asm	__based	__cdecl	__export	__far
__fastcall	__fortran	__huge	__inline	__interrupt
__loadds	__near	__pascal	__saveregs	__segment
__segname	__self			

图 1.2 Visual C++ 的 C 扩充的保留字

1.2 变量的声明

在前一章中我们已经说过，任何变量在使用前一定要先声明。

范例：若是想将 i, j, k 三个数声明为整数，则下列的声明方式均是合法的。

1. `int i, j, k;`

2. `int i, j, k;`

由于 i 及 j 之间仍是用“,”隔开，所以上述是合法的声明方式。

3. `int i;`
`int j;`
`int k;`

由于每一次声明完皆是由“;”做结束，所以这仍是合法的声明。

另外，你也可以在声明变量的同时设置变量的值。

范例：将 i 声明成整数，并将其设置成 7。

`int i = 7`

1.3 基本数据类型

C语言的基本数据类型有:

1. int: 整数
2. char: 字符
3. float: 单精度浮点数
4. double: 双精度浮点数

1.3.1 整数 int

有时候我们又可在整数前面加上一些限定字, 例如: short, long, unsigned 和 signed。所以事实上整数的声明有九种, 如图 1.3 所示。

整数声明类型	长 度	值的范围
int	16	-32768 ~ 32767
unsigned int	16	0 ~ 65535
signed int	16	-32768 ~ 32767
short int	16	-32768 ~ 32767
unsigned short int	16	0 ~ 65535
signed short int	16	-32768 ~ 32767
long int	32	-2147483648 ~ 2147483647
unsigned long int	32	0 ~ 4294967295
signed long int	32	-2147483648 ~ 2147483647

图 1.3 整数有关概念表 (长度单位是位)

1.3.2 字符 char

一般每一个 char 声明的变量所占据的存储器空间是 8 位。因为:

$$2^8 = 256$$

所以每个字符可代表 256 个不同的值。C 语言系统中, 这 256 个不同的值是根据 ASCII 码的值排列的, 而这些码的值包含小写字母、大写字母、数字、标点符号及其它一些特殊符号, 如附录 A 所示。

范例: 请声明一字符变量 single_char, 其声明方式如下:

```
char single_char;
```

范例: 声明一字符变量 single_char, 并将其值设置成 'a'。

```
char single_char = 'a';
```

另外在 C 里面, 有一些无法打印字符, 例如: '\0', 虽然在单引号中有 \ 和 0, 但是它们合起来只能算是一个字符, 我们又称这些字符为换码 (escape) 字符。图 2.4 是这