

Visual

C++实用大全

程文斌 金相凤 编

Visual Visual
Visual Visual
Visual Visual

北京航空航天大学出版社

Visual C++实用大全

程文斌 金相风等 编

北京航空航天大学出版社

(京)新登字 166 号

图书在版编目(CIP)数据

Visual C++实用大全/程文斌等编. -北京:北京航空航天大学出版社, 1995, 6
ISBN 7-81012-576-1

I. V... II. 程... III. C 语言, 程序设计 IV. TP312C

中国版本图书馆 CIP 数据核字 (95) 第 04671 号

内 容 简 介

本书是针对利用 Visual C++ 进行 C 和 C++ 程序设计的人编写的, 全面系统地介绍了 Visual C++。首先介绍 Visual C++ 集成环境(PWB), 说明集成编译器和调试器的用法; 然后介绍 C 和 C++ 语言的基础知识, 包括操作符和表达式、说明、控制语句、函数、指针和其它高级数据结构, 介绍文件输入/输出; 接着介绍了类型兼容、宏、条件编译指令、预编译处理, 讨论了面向对象的基本编程方法; 再就是说明了处理屏幕文本和图形的技术, 介绍了如何充分利用 Visual C++ 的库函数, 描述了与汇编语言接口技术; 此后, 介绍了用 Visual C++ 进行 Windows 程序设计的技术; 在附录部分给出了 Visual C++ 基础类库的快速参考。适合程序员使用。

● 书 名: Visual C++ 实用大全

Visual C++ SHIYONG DAQUAN

● 编 著 者: 程文斌 金相凤等 编

● 责任编辑: 张躬行 许传安

● 出 版 者: 北京航空航天大学出版社

● 地 址: 北京市海淀区学院路 37 号(100083), 发行科电话 2015720

● 印 刷 者: 北京朝阳科普印刷厂印刷

● 发 行: 新华书店总店科技发行所

● 经 售: 全国各地书店

● 开 本: 787×1092 1/16

● 印 张: 51.5

● 字 数: 1318 千字

● 印 数: 5000 册

● 版 次: 1995 年 6 月第 1 版

● 印 次: 1995 年 6 月第 1 次印刷

● 书 号: ISBN 7-81012-576-1/TP·164

● 定 价: 62.00 元

前 言

Visual C++的前身为 Microsoft C/C++,从 7.0 版开始,Microsoft C/C++演化为 Visual C++ 1.0,后来又推出了 Visual C++ 1.5 和 2.0。Visual C++非常庞大,功能很强,而 Microsoft C/C++相对较小,比较实用。

从 Visual C++ 1.0 开始,Visual C++的集成环境 Visual Workbench 运行于 Windows,是一个很好用的 Windows 应用程序开发环境,要特别指出的是,在新的 IDE 中可在 Windows 中直接调试程序。

在 Visual C++ 1.0~2.0,配备了 CodeView for Windows 用来调试用 Visual C++设计的 Windows 应用程序,增强了资源包括图标、光标、位图、对话框、菜单、热键和字体等的设计。特别是在 Visual C++ 2.0 中,资源设计已集成到 Visual C++的 Workbench 中,可用 Resource 菜单访问,用来设计 Windows 应用程序的界面。Visual C++在 MFC 中封装了 Windows API,包括函数、数据结构和消息等,2.0 版支持在 Windows 3.1,Windows NT 3.5 和 Windows 95 操作系统下开发应用程序,方便 C/C++读者开发 Windows 应用程序。

本书介绍 Visual C++提供的工具,包括 C++编辑器、编译器、程序员工作台(Visual Workbench)、Visual Workbench 集成调试器、CodeView 调试器、剖析工具和各种 Windows 开发工具,这部分对 Visual Workbench 和实用工具的介绍比 Microsoft 公司的手册还详尽、具体和实用。

使用每种设计语言时,程序员需要完整地去理解它。本书包含了 C、C++和 Windows 语言的所有重要的设计原理,包括新基础类库。对于初学者,前面的章节可以打好编写更复杂程序的坚实基础;对于高级设计者,这些章将作为一个手册,介绍 C++的原理。

本书还介绍了如何调试代码,并避免语法和逻辑设计错误;帮助理解过程设计与面向对象设计的不同以及学会如何开发简单的 OOP 程序。

本书讨论了 Microsoft Windows 设计,有四章的篇幅为读者讲解 Windows 原理,并用示例解说如何编写简单的 Windows 程序。附录 C 列出了许多重要的 Windows 函数和基础类库函数和数据结构。在讲解时尽量选择了一些简单、完整的例子,以便可以研究、修改它,按自己的需要扩展在自己的程序中。

本书各章的主要内容是:

第一章至第六章介绍了包含于 Visual C++编译包中的 Visual Workbench 和实用程序。

第七章至第十六章讲述了 C/C++语言的基本程序设计原理。面向过程的一些章节讲述传统 C 和 C++程序设计原理。

第十七章至第二十章,给出了一个 C++面向对象设计的完整介绍。其中有定义和完整的例子,可以帮助读者完成面向对象程序的开发。

第二十一章至第二十三章讲述了如何为 DOS 建立图形应用程序,如何使用重要的库函数。第二十一章完整地介绍了 Microsoft 的 graphics 和 chart 库。第二十二章包含了 C、C++库函数的详细讨论。第二十三章讲述了 C、C++的汇编语言接口;描述了如何在 C、C++与汇编

语言之间进行参数传递;如何用汇编和低层函数与外部设备接口。

第二十四章至第二十五章介绍了 Microsoft Windows 原理,告诉读者如何使用 Visual C++ 编译器来开发 Windows 应用程序。这两章中的应用程序是传统的、基于消息用 C 编写的程序。

第二十六章和第二十七章致力于如何使用 Microsoft 基础类库去编程。通过使用强有力的 C++ 类、Microsoft 基础类库将大大缩短 Windows 应用程序的开发周期和源程序长度。

本书适用于 Microsoft C/C++、Visual C++ 以及各种演化版本。

目 录

第一部分 快速启动 C/C++ 编程

第一章 Visual C++ 编译器软件包	3
1.1 Windows 下的典型安装	3
1.2 建 议	4
1.2.1 最小的软硬件要求	4
1.2.2 推荐的软硬件配置	4
1.2.3 好的安装计划	4
1.2.4 AUTOEXEC.BAT 和 CONFIG.SYS	5
1.2.5 DOS 和 Windows 下的工作	5
1.3 实用程序包	5
1.3.1 程序员工作台 (Visual Workbench)	5
1.3.2 CL —— DOS 命令行编译器	6
1.3.3 Visual Workbench 集成编译器	6
1.3.4 LINK —— 连接器	6
1.3.5 EXEHDR —— EXE 文件头文件工具	6
1.3.6 NMAKE —— 工程维护	6
1.3.7 BUILD —— Visual Workbench 工程维护	6
1.3.8 LIB —— 库管理程序	7
1.3.9 HELPMAKE —— 帮助文件维护	7
1.3.10 BSCMAKE 和 SBRPACK —— 浏览器数据库维护	7
1.3.11 CVPACK —— 调试信息压缩器	7
1.3.12 IMPLIB —— 输入库管理程序	7
1.3.13 RM —— 文件转移	8
1.3.14 UNDEL —— 恢复删除	8
1.3.15 EXP —— 删除隐含子目录 DELETED	8
1.3.16 CodeView 调试器	8
1.3.17 Profiler	8
1.3.18 源浏览器 (Source Browser)	9
1.3.19 Windows 开发工具	9
1.3.20 资源编译器	10
1.3.21 Spy	10
1.3.22 WXServer	10
1.3.23 QH Advisor	10
1.3.24 386MAX	10

1.4 C/C++和 Windows 编程特点	11
1.4.1 C++ 3.0 适用范围	11
1.4.2 编译器优化	11
1.4.3 编译器存储模式	12
1.4.4 P 码	12
1.4.5 MOVE —— Microsoft 虚拟环境覆盖	13
1.4.6 基本类型库(MFC)	13
1.4.7 函数在线	15
第二章 启动程序员工作台	16
2.1 进入 Visual Workbench 环境	16
2.1.1 主菜单分类	16
2.1.2 书写新的应用程序	27
2.1.3 编辑应用程序	29
2.1.4 拷贝、剪切和粘贴(Copying ,Cutting 和 Pasting)	30
2.2 编译和连接	32
2.2.1 在 Visual Workbench 内工作	32
2.2.2 在 DOS 命令行下工作	33
2.3 使用 Microsoft 帮助系统	34
2.3.1 程序语句帮助	34
2.3.2 其他帮助特性	34
2.3.3 打印帮助信息	36
2.4 设置重要的 Options 选项	36
2.4.1 Project... (工程选项)	36
2.4.2 调试选项	40
2.4.3 目录选项	40
2.4.4 编辑器选项	43
2.4.5 Workspace 选项	44
2.4.6 工具选项	45
2.4.7 颜色选项	46
2.4.8 字体选项	48
2.4.9 Tools. INI 文件	48
2.5 改变 build 过程	49
第三章 Visual Workbench 编辑器	50
3.1 编辑器命令	53
3.2 编辑器激活及文本键入	53
3.3 字符、字和行的删除	55
3.4 使用剪贴板	55
3.5 进一步说明如何移动光标	57
3.6 字符系列的搜索和替换	57

3.7	书签的设置和搜索	61
3.8	存储和装入文件	62
3.9	自动缩进	63
3.10	打印选择的文本和整个文件	64
3.11	对匹配	66
3.12	其它有关命令	66
3.13	命令综述	67
3.14	编辑器缺省的修改	*68
3.15	编辑器的帮助	68
第四章	启动 C/C++ 编译器	70
4.1	缺省编译器选项下的简单编译	70
4.1.1	在 Visual Workbench 下编译	71
4.1.2	在 DOS 命令行下编译	72
4.2	设置编译器和连接器开关	72
4.2.1	编译器开关	73
4.2.2	输出文件编译器选项	74
4.2.3	预处理器编译器选项	74
4.2.4	语言(Language)编译器选项	75
4.2.5	浮点编译器选项	75
4.2.6	源列表编译器选项	75
4.2.7	杂类编译器选项	76
4.2.8	宏汇编选项	77
4.2.9	连接编译器选项	77
4.2.10	优化编译器选项	77
4.2.11	存储模式编译器选项	78
4.2.12	连接器选项	78
4.3	用 Make 文件控制编译操作	80
4.3.1	Visual Workbench 工程文件	81
4.3.2	DOS NMAKE	85
4.4	特别编译选项	86
4.4.1	模块定义文件(.DEF)	86
4.4.2	资源编译器	88
4.5	排除错误	88
第五章	集成调试器	89
5.1	错误的类型	89
5.1.1	编译时间错误	89
5.1.2	运行时间错误	90
5.1.3	逻辑错误	90
5.2	生成调试信息	91

5.3	指明程序参数	92
5.4	控制程序执行	92
5.4.1	监视程序的输出	92
5.4.2	单步执行程序(Step over code)	93
5.4.3	跟踪执行代码(Tracing Into)	95
5.4.4	单步跳过一段程序代码	95
5.4.5	停止程序运行	97
5.4.6	重新开始	97
5.5	检查变量的值	97
5.5.1	什么是表达式	97
5.5.2	监视表达式	97
5.5.3	计算和修改表达式	100
5.5.4	检查寄存器的值	101
5.5.5	显示局部变量值	102
5.6	使用断点	103
5.6.1	设置断点	103
5.6.2	使用断点工作	105
5.7	解决一般保护错误	108
5.8	检查错误	108
5.9	调试动态链接库	109
5.10	Hard Mode	109
第六章	启动高级工具:CodeView,Browser 和 Profiler	110
6.1	CodeView	110
6.1.1	一个简单的 DOS 例程	111
6.1.2	C++ 程序使用 CodeView	117
6.1.3	Windows 程序使用 CodeView	119
6.2	源浏览器	123
6.2.1	通用浏览器信息	124
6.2.2	符号定义查找	124
6.2.3	符号引用查找	125
6.2.4	创建调用树	127
6.3	Profiler	128
6.3.1	Profiler 批处理文件	128
6.4	代码书写形式	129

第二部分 C 和 C++ 编程基础

第七章	C 和 C++ 基础	133
7.1	C 的发展史	133
7.1.1	与其他语言的关系	134

7.1.2	C 语言的优势	135
7.1.3	C 语言的弱点	137
7.1.4	编程的规程	137
7.2	ANSI C 标准	137
7.3	C++ 的进化和面向对象的编程	139
7.4	C++ 的历史	139
7.4.1	使用 C++ 对象来进行流线代码设计	140
7.4.2	对 C 的提高	140
7.4.3	对 C 的主要提高	142
7.5	C 程序的基本元素	143
7.5.1	程序的五个基本成分	143
7.5.2	第一个 C 程序	144
7.5.3	第一个 C++ 程序	145
7.5.4	第二个 C 程序	145
7.5.5	第二个 C++ 程序	148
7.5.6	文件	150
第八章	数 据	154
8.1	标识符	154
8.2	关键字	154
8.3	标准 C 和 C++ 数据类型	155
8.3.1	字符	155
8.3.2	三种整数	157
8.3.3	unsigned 修饰符	157
8.3.4	浮点数	158
8.3.5	枚举型	159
8.4	访问修饰符	161
8.4.1	const 修饰符	161
8.4.2	#define 常量	162
8.4.3	Volatile 修饰符	162
8.4.4	const 和 volatile 一起使用	162
8.5	pascal, cdecl, near, far 和 huge 修饰符	163
8.5.1	pascal	163
8.5.2	cdecl	164
8.5.3	near, far 和 huge	164
8.6	数据类型转换	165
8.6.1	显式类型转换	166
8.7	存储类型	166
8.7.1	外层的变量说明	166
8.7.2	内层的变量说明	168

8.7.3	变量作用域	169
8.7.4	外层函数说明	169
8.8	操作符	170
8.8.1	位操作符	170
8.8.2	左移和右移	171
8.8.3	增量与减量	172
8.8.4	算术操作符	172
8.8.5	赋值操作符	173
8.8.6	复合赋值操作符	173
8.8.7	关系和逻辑操作符	175
8.8.8	条件操作符	177
8.8.9	逗号操作符	177
8.9	理解操作符优先顺序	178
8.10	标准 C 和 C++ 库函数	178
第九章	控 制	181
9.1	条件语句	181
9.1.1	if 语句	181
9.1.2	if - else 语句	182
9.1.3	嵌套 if - else	184
9.1.4	if - else - if 语句	185
9.1.5	? 条件语句	186
9.1.6	switch 语句	187
9.1.7	混合使用 if - else - if 和 switch 语句	193
9.2	循环语句	194
9.2.1	for 循环	195
9.2.2	while 循环	199
9.2.3	do - while 循环	201
9.2.4	break 语句	203
9.2.5	使用 CodeView	203
9.2.6	continue 语句	203
9.2.7	使用 CodeView	204
9.2.8	一起使用 break 和 continue	205
9.2.9	exit() 语句	206
9.2.10	atexit() 语句	209
第十章	编写和使用函数	211
10.1	函数原型和风格	211
10.1.1	构造原型	211
10.1.2	通过值调用和通过引用调用	213
10.1.3	存储类型	215

10.1.4	作用域	215
10.1.5	递归	215
10.2	函数参数	216
10.2.1	形式参数和实际参数	216
10.2.2	void 类型的参数	217
10.2.3	字符参数	217
10.2.4	整数参数	218
10.2.5	浮点参数	219
10.2.6	双精度参数	220
10.2.7	数组参数	221
10.3	函数类型	223
10.3.1	void 函数类型	223
10.3.2	char 函数类型	224
10.3.3	int 函数类型	225
10.3.4	long 函数类型	226
10.3.5	float 函数类型	227
10.3.6	double 函数类型	228
10.4	main()的参数	228
10.4.1	字符串	229
10.4.2	整数	229
10.4.3	浮点数	231
10.5	重要的 C++ 特性	232
10.5.1	直接插入	232
10.5.2	重载	233
10.5.3	省略号(...)	234
10.6	作用域规则遇到的问题	235
10.6.1	C 程序中一个未定义的符号	235
10.6.2	使用具有文件作用域的变量	236
10.6.3	用局部变量屏蔽全局变量	237
10.6.4	C++ 的作用域问题	238
10.6.5	C++ 作用域限定操作符	239
第十一章	数 组	240
11.1	什么是数组	240
11.2	数组和 C	240
11.3	数组说明	240
11.4	数组初始化	241
11.4.1	缺省初始化	241
11.4.2	显式初始化	242
11.4.3	不定长初始化	242

11.5	访问数组元素.....	243
11.6	计算数组大小[sizeof ()]	245
11.7	数组下标越界.....	246
11.8	字符串输入输出.....	247
11.9	多维数组.....	249
11.10	数组作为函数参数	252
11.10.1	将数组传给 C 函数	252
11.10.2	数组传给 C++ 函数	254
11.11	字符串函数和字符数组	259
11.11.1	gets(), puts(), fgets(), fputs()和 sprintf()	259
11.11.2	strcpy, strcat, strcmp, strlen	261

第三部分 C++面向对象编程基础

第十二章	指 针.....	267
12.1	定义指针变量.....	267
12.1.1	指针变量说明.....	268
12.1.2	简单语句使用指针变量.....	269
12.1.3	指针变量初始化.....	271
12.1.4	地址操作符的错误使用.....	273
12.1.5	数组指针.....	273
12.1.6	指针的指针.....	274
12.1.7	串的指针.....	275
12.1.8	指针运算.....	276
12.1.9	指针运算和数组.....	278
12.1.10	操作符++和--的有关事项	280
12.1.11	比较指针	280
12.1.12	可移植性	281
12.1.13	指针的大小	281
12.2	函数的指针.....	282
12.3	动态存储.....	285
12.3.1	使用空指针.....	287
12.4	指针和数组.....	290
12.4.1	串(字符型的数组).....	290
12.4.2	指针数组.....	290
12.4.3	更多的有关指针的指针的内容.....	293
12.4.4	串指针的数组.....	297
12.5	C++引用类型	300
12.5.1	函数返回地址.....	300
12.5.2	使用 CodeView	301

12.5.3 何时使用引用类型.....	302
第十三章 C 语言中的 I/O	303
13.1 流函数.....	305
13.1.1 打开流.....	305
13.1.2 输入和输出的重定向.....	306
13.1.3 改变流缓冲区.....	307
13.1.4 关闭流.....	308
13.2 C 中的低级输入和输出	309
13.3 字符输入和输出.....	309
13.3.1 使用 getc(),putc(),fgetc()和 fputc()	309
13.3.2 使用 getchar(),putchar(),fgetchar()和 fputchar()	310
13.3.3 使用 getch()和 putch().....	310
13.4 串输入和输出.....	311
13.4.1 使用 gets(),puts(),fgetx()和 fputs()	311
13.5 整数输入和输出.....	312
13.5.1 使用 getw()和 putw()	312
13.6 格式化输出.....	315
13.6.1 使用 printf()和 fprintf()	315
13.7 使用 fseek(),ftell()和 rewind()	319
13.7.1 使用 codeview	321
13.8 格式化输入.....	323
13.8.1 使用 scanf(),fscanf()和 sscanf().....	324
第十四章 C++的输入、输出	325
14.1 使用 C++精简 I/O	325
14.1.1 cin,cout 和 cerr	325
14.1.2 抽取(>>)和插入(<<)操作.....	325
14.2 从 stream.h 到 iostream.h	332
14.2.1 操作符和成员函数.....	332
第十五章 结构、联合和其它数据结构	345
15.1 C 和 C++结构	345
15.1.1 C 和 C++结构:语法和规则	345
15.1.2 C++结构:语法和规则的扩展	347
15.1.3 访问结构成员.....	347
15.1.4 构造一个简单的结构.....	348
15.1.5 将结构传递给函数.....	349
15.1.6 构造一个结构数组.....	351
15.1.7 使用结构指针.....	353
15.1.8 传递结构数组给函数指针.....	355
15.1.9 C++中结构的使用	357

15.1.10	对结构的附加操作	360
15.2	联合	362
15.2.1	联合:语法和规则	362
15.2.2	构造一个简单的联合	363
15.3	其它数据类型	364
15.3.1	使用 typedef	364
15.3.2	使用枚举类型	365
第十六章	高级 C 和 C++ 编程专题	368
16.1	类型兼容	368
16.1.1	ANSI C 对于类型兼容的定义	368
16.1.2	什么是同一类型	368
16.1.3	枚举类型	369
16.1.4	数组类型	369
16.1.5	函数类型	370
16.1.6	结构和联合类型	370
16.1.7	指针类型	370
16.1.8	多源文件的相容性	370
16.2	宏	360
16.2.1	定义宏	371
16.2.2	宏和参数	371
16.2.3	宏扩展的问题	372
16.2.4	生成和使用自己的宏	373
16.2.5	编译器预置的宏	374
16.3	高级预处理语句	374
16.3.1	#ifdef 和 #endif 指令	374
16.3.2	#undef 指令	375
16.3.3	#ifndef 指令	375
16.3.4	#if 指令	375
16.3.5	#else 指令	376
16.3.6	#elif 指令	376
16.3.7	#line 指令	376
16.3.8	#error 指令	376
16.3.9	#pragma 指令	377
16.4	条件编译	377
16.5	高级预处理操作	378
16.5.1	串化操作(#stringize operator)	378
16.5.2	连接操作(##)	378
16.5.3	字符操作(#@)	379
16.6	头文件的正确使用	379

16.7	有效地使用头文件	380
16.8	预编译的头文件	380
16.8.1	产生预编译的头	381
16.8.2	在 Visual Workbench(程序员工作台)上使用预编译头文件	381
16.9	limits.h 和 float.h	381
16.10	错误处理: perror()	383
16.11	存储模式	383
16.11.1	Tiny(微)	384
16.11.2	Small(小)	384
16.11.3	Medium(中)	384
16.11.4	Compact(紧缩)	384
16.11.5	Large(大)	384
16.11.6	Huge(巨)	384
16.12	动态存储分配: 链表	385
16.12.1	使用链表时需要考虑的事项	385
16.12.2	一个简单的链表	386
第十七章	面向对象的编程介绍	390
17.1	并非是新内容	390
17.2	传统结构化设计	390
17.3	面向对象的程序设计	391
17.4	面向对象的术语	392
17.4.1	封装	392
17.4.2	类的层次性	393
17.5	认识 C++ 类	394
17.5.1	一个作为私有类的结构	394
17.5.2	C++ 类的语法规则	398
17.5.3	一个简单的 C++ 类	399
第十八章	C++ 的类	403
18.1	附加的类特性	403
18.1.1	一个简单的类	404
18.1.2	嵌套类	404
18.1.3	构造函数和析构函数	407
18.1.4	重载类的成员函数	413
18.1.5	使用友元函数来存取私有类变量	417
18.1.6	使用 this 指针	419
18.2	使用操作符重载	420
18.2.1	操作符重载和函数调用	420
18.2.2	重载的语法	420
18.3	派生类	423

18.3.1	派生类的语法	423
18.3.2	产生派生类	423
第十九章	C++的 I/O	428
19.1	枚举类型	428
19.2	引用变量	429
19.3	缺省参数	431
19.4	memset()	432
19.5	格式化的输出	432
19.6	C/C++ I/O 选择	436
19.7	iostream 类列表	437
19.7.1	输入流类	440
19.7.2	输出流类	441
19.7.3	缓冲流类	443
19.7.4	串流类	444
19.8	二进制文件	446
19.9	使用 extern "C"来结合 C 和 C++代码	448
19.10	编写自己的操纵符	450
19.10.1	无参数的操纵符	450
19.10.2	带有一个参数的操作符	451
19.10.3	多参数的操纵符	452
第二十章	基于面向对象环境的工作	455
20.1	C++中一个面向对象的栈	455
20.2	C++中面向对象的连接表	457
20.2.1	父类的产生	458
20.2.2	派生子类	458
20.2.3	使用友元类	460
20.2.4	测试一个完整的程序	462
20.2.5	连接表输出	470
20.3	面向对象 C++的其它内容	472
第四部分 DOS 图形、系统访问库及混合语言接口		
第二十一章	DOS 下的图形设计	475
21.1	文本和图形方式	475
21.2	使用简单的图形函数	479
21.2.1	画一个矩形	479
21.2.2	确定屏幕的分辨率	481
21.2.3	图形环境的信息	482
21.2.4	图形详解	484
21.2.5	使用基本图形函数	490