

BORLAND

C++

程式設計師手冊



Alon E. Cohen

瑩園電腦軟體研究開發部

書號：81420

開數：16 開

頁數：336 頁

適合讀者：稍具電腦基礎者

所需軟體：Borland C++ 3.0 軟體

本書特點：

Borland C++ 程式設計師手冊因內容相當多，為避免造成讀者在閱讀上的不方便及滿足不同的使用者，故將其分成二本書以節省使用者不必要的浪費，分別為『Borland C++ 程式設計入門手冊』、『Borland C++ 程式設計師手冊』，讀者可整套、或單獨購買。

Borland C++ 程式設計入門手冊

適合給初次接觸 Borland C++ 程式設計或初次學習 C 語言的人來使用。除指導使用者如何利用 Borland C++ 來設計程式外，更說明了 Borland C++ 與 Turbo C 之相容性，只不過 Borland C++ 多了一個支援 Windows 的程式，說明 Borland C++ 在語法及參數上的改變，可指導讀者撰寫程式、Compiler、Link ... 等多項功能。

Borland C++ 程式設計師手冊

屬於更進一步的程式設計應用，如 "執行時函數庫的交互參考" "螢幕函數" "Microsoft C 轉換至 Borland C++" "記憶體模式、浮點運算和疊覆" "組合語言界面" 等，為更深一層的程式管理，提供給 Borland C++ 的使用者或程式設計師修改程式時之參考。

內容概要：

第一章：提供了執行時程式庫的原始碼資訊以及首檔的內容。並且提供了一套以主題為序之執行時程式庫的交互參考索引。舉例來說，如果想找出哪些函數是和繪圖有關，可以利用 "繪圖" 或 "

圖形" 為題在本章中查閱。

第二章：說明 Turbo C++ 在文字和圖形上的處理。

第三章：介紹將 Microsoft C 的 C 程式轉換至 Borland C++ 時，所必須注意的相關事項。

第四章：包含記憶體模式，混合模式之程式設計，浮點運算的相關資料及覆疊。

第五章：告訴我們如何去撰寫可從 Borland C++ 程式中呼叫的組合語言程式。

附錄 A：列舉並解釋所有的執行或編譯時的錯誤或警告訊息，並指出造成錯誤的可能原因以及解決之道。

附錄 B：中英名辭對照。

相關書籍：

TURBO C++ 使用手冊

TURBO C 2.0 使用手冊

TURBO C++ 入門指引

C 語言武功祕笈

C 語言及各種語言間之聯結技巧

C 語言與 DOS/BIOS 間之探討實例

C 語言進階 C++

Borland C++ 程式設計入門手冊

目 錄

第一章 執行時函數庫的交互參考

1.1 為何要利用執行時函數庫.....	1-1
1.2 原始程式碼.....	1-2
1.3 Turbo C++ 的標頭檔.....	1-3
1.4 分類說明函數庫內的常式.....	1-8
1.4.1 分類常式.....	1-8
1.4.2 轉換常式.....	1-8
1.4.3 目錄管理常式.....	1-9
1.4.4 診斷常式.....	1-9
1.4.5 繪圖常式.....	1-9
1.4.6 輸入／輸出常式.....	1-11
1.4.7 介面常式 (DOS, 8086, BIOS).....	1-12
1.4.8 處理常式.....	1-13
1.4.9 數學常式.....	1-14
1.4.10 記憶體常式.....	1-15
1.4.11 多方面的常式.....	1-15
1.4.12 程序控制常式.....	1-16
1.4.13 標準常式.....	1-16
1.4.14 本文視窗顯示常式.....	1-16
1.4.15 時間與日期常式.....	1-17
1.4.16 可變的引數列常式.....	1-17

第二章 螢幕函數

2.1 關於螢幕模態.....	2-1
2.2 關於視窗及觀察窗.....	2-3
2.2.1 什麼是視窗?.....	2-3

2.2.2 什麼是觀察窗?.....	2-3
2.2.3 座標.....	2-4
2.3 文字模態的程式設計.....	2-4
2.3.1 螢幕輸入 /輸出函數.....	2-4
2.3.1.1 文字輸出及處理.....	2-5
2.3.1.2 視窗及模態控制.....	2-6
2.3.1.3 屬性控制.....	2-7
2.3.1.4 狀況查詢.....	2-8
2.3.1.5 游標形狀(Cursor shape).....	2-9
2.3.2 文字視窗.....	2-9
2.3.3 文字模態種類.....	2-11
2.3.4 文字顏色.....	2-12
2.3.5 高效率輸出.....	2-13
2.4 繪圖模態的程式設計.....	2-14
2.4.1 繪圖程式庫.....	2-15
2.4.2 繪圖系統控制.....	2-15
2.4.3 畫圖及填圖.....	2-19
2.4.4 螢幕及觀察窗的處理.....	2-20
2.4.5 繪圖模態下的文字輸出.....	2-22
2.4.6 顏色控制.....	2-25
2.4.6.1 圖元及調色板.....	2-25
2.4.6.2 背景色及畫圖顏色.....	2-26
2.4.6.3 CGA 下的顏色控制.....	2-27
2.4.6.4 在 EGA及 VGA下的顏色控制.....	2-29
2.4.7 繪圖模態下的錯誤處置.....	2-30
2.4.8 狀況查詢.....	2-32

第三章 由 Microsoft C 轉換至 Borland C++

3.1 環境與工具.....	3-1
3.1.1 .h與 .LIB 檔案的存取路徑.....	3-2
3.1.2 新版的 MAKE.....	3-5
3.1.3 命令列編譯程式.....	3-5
3.1.4 命令列選項與程式館.....	3-18
3.1.5 聯結程式.....	3-19

3.2 原始碼層次的相容性.....	3-22
3.2.1 __MSC 巨集.....	3-23
3.2.2 有些未列入標準的檔案可以用兩種名稱引入使用，如底下 所示：.....	3-23
3.2.3 記憶體模式.....	3-23
3.2.4 關鍵字.....	3-25
3.2.5 浮點傳回值.....	3-26
3.2.6 結構傳回值 (Structure returned by value).....	3-26
3.3 轉換指引.....	3-27

第四章 記憶體模式、浮點運算和疊覆

4.1 記憶體模式.....	4-2
4.1.1 8086 的暫存器.....	4-2
4.1.1.1 一般用途暫存器.....	4-3
4.1.1.2 分段暫存器.....	4-4
4.1.1.3 特殊用途暫存器.....	4-4
4.1.1.4 旗標暫存器(Flag Register).....	4-5
4.1.2 分段記憶體 (Memory Segmentation).....	4-6
4.1.2.1 位址計算法.....	4-7
4.1.3 近(Near)、遠(Far)、及巨大 (huge)指標.....	4-8
4.1.3.1 近指標(Near Pointer).....	4-9
4.1.3.2 遠指標(Far Pointer).....	4-9
4.1.3.3 巨大指標(Huge Pointers).....	4-11
4.1.4 Turbo C 所提供的六種記憶體模式.....	4-13
4.2 混合模式程式方式：位址修飾詞 (Addressing Modifiers).....	4-19
4.2.1 宣告近函數或遠函數.....	4-21
4.2.2 近、遠或巨大指標的宣告.....	4-23
4.2.3 指到一個指定的 "分段:偏移" 位址.....	4-25
4.2.4 使用程式館檔案.....	4-25
4.2.5 聯結不同模式的模組.....	4-27
4.3 浮點運算選擇項 (Floating-point options).....	4-29
4.3.1 模擬 80x87 晶片.....	4-30
4.3.2 使用 80x87 輔助處理器時.....	4-30
4.3.3 不使用浮點運算時.....	4-31

4.3.4 使用較快速的浮點運算.....	4-31
4.3.5 80x87 環境變數.....	4-31
4.3.6 暫存器和 80x87.....	4-33
4.3.7 消除浮點運算的異常.....	4-33
4.4 使用複數.....	4-34
4.4.1 使用 BCD.....	4-35
4.5 Turbo C++ 的 RAM使用方式.....	4-38
4.6 覆疊 (VROOMN).....	4-39
4.4.1 覆疊的工作原理.....	4-39
4.4.2 需求.....	4-42
4.4.3 使用覆疊.....	4-43
4.4.4 覆疊程式的設計.....	4-45
4.4.5 交換.....	4-48

第五章 組合語言介面

5.1 混合語言程式設計：和其它語言溝通.....	5-2
5.1.1 參數傳遞順序：C 和 Pascal.....	5-2
5.1.1.1 C 的參數傳遞順序.....	5-2
5.1.1.2 Pascal 的參數傳遞順序.....	5-4
5.2 從 Turbo C++ 呼叫組合語言常式的準備.....	5-7
5.2.1 簡化區段名稱.....	5-7
5.2.2 標準區段名稱.....	5-8
5.2.3 定義資料常數和變數.....	5-10
5.2.4 外部和總體性識別名的定義.....	5-10
5.3 從組合語言常式中呼叫 C函數的準備.....	5-12
5.3.1 引用函數.....	5-12
5.3.2 引用資料.....	5-13
5.4 定義組合語言常式.....	5-14
5.4.1 參數傳遞.....	5-15
5.4.2 處理傳回值.....	5-15
5.5 暫存器慣用法.....	5-20
5.6 從組合語言常式中呼叫 C函數.....	5-21
5.7 低階程式技巧：虛擬變數、夾在程式中的組合碼和插斷函數.....	5-23
5.7.1 虛擬變數.....	5-23

5.7.2 使用夾在程式中的組合碼.....	5-27
5.7.2.1 運算碼.....	5-31
5.7.2.2 程式中組合碼引用資料及函數.....	5-34
5.7.2.3 程式中的組合碼及暫存器變數.....	5-35
5.7.2.4 夾在程式中的程式碼、偏移值和大小重定 (size overrides).....	5-36
5.7.2.5 使用跳躍指令和址標.....	5-38
5.7.3 插斷函數.....	5-38
5.7.4 使用低階技巧的範例.....	5-41

附錄 A 錯誤訊息

A.1 執行時的錯誤訊息 (Run-time error messages).....	A-3
A.2 編譯錯誤訊息.....	A-6
A.2.1 致命錯誤 (fatal errors).....	A-8
A.2.2 錯誤 (errors).....	A-8
A.3 警告 (warnings).....	A-45

附錄 B 中英名辭對照表

第一章

執行時函數庫的 交互參考

1.1 爲何要利用執行時函數庫

BONANZA 堂園 (R)

本章將介紹 Turbo C++ 的函數庫和含入檔案。內容包括了以下三個部分：

- 解釋使用函數庫的原因。
- 列舉所有的標題檔並加以簡略的說明。
- 以函數的功能加以分門別類。

Turbo C++ 擁有超過 450 個不同的函數和巨集，處理經常遇上的問題而不必使用者自行撰寫程式。例如低階輸入輸出、高階輸入輸出、字串和檔案的處理、資料的轉換、數學上的計算、記憶體的配置、程序的控制以及其他許許多多不

同功能。讀者可以在函數庫手冊中找到所有函數和巨集的詳細資料。

Turbo C++ 的函數皆放在函數庫檔案之中 (Cx.LIB, CPx.LIB, MATH.LIB 和 GRAPHICS.LIB)。在 Turbo C++ 六種不同的記憶體模式中，除了微小模式 (tiny) 之外，其他五個模式都各自擁有不同的程式庫檔案 (微小模式 (tiny) 和小模式 (small) 使用相同的函數庫檔案。)

所有的函數庫都已經在標題檔案中以函數原型的方式加以宣告了。

1.2 原始程式碼

BONANZA 堂園 (R)

Turbo C++ 擁有超過 450 個功能強大的函數：IBM PC 的低階控制，和 DOS 之間的介面、輸入輸出流程式管理、字串和記憶體的處理、數學運算、排序、搜尋等等。以下列出為什麼要使用原始程式碼的原因：

- 修改函數庫裡的函數以符合需要。
- 在除錯的時候，假如想佑道函數內部運作的情形，擁有原始程式碼對工作將有很大的幫助。
- 檢視函數庫的原始程式碼會讓你充分瞭解某一函數究竟在做些什麼。
- 透過原始程式碼可以去掉 C 符號中你不希望有的前導底線。

■ 透過原始程式碼可以學到許多專業的程式寫作技巧。

Borland 公司基於 "開放架構" 的觀念，只要填好訂單，向 Borland 公司購買執行時函數庫原始程式碼，就可以任意的修改或使用原始程式碼。

1.3 Turbo C++ 的標頭檔

BONANZA 盛園 (R)

標頭檔(header file) 也叫做含入檔案(include file)。在標頭檔中包括有函數庫的函數原型宣告，Turbo C++ 和函數庫所定義的全域變數，同時標題內也定義了函數庫所用到的資料型態和符號常數。Turbo C++ 對標題檔的命名及標題檔的內容皆符合了 ANSI C 的標準，以下列出 Turbo C++ 所有的標頭檔。在檔名前標記 ANSI C 的標題檔是由 ANSI C 所定義的，而標記 C++ 則是 C++ 程式所使用的標題檔。

	alloc.h	宣告記憶體管理函數的原型 (配置以及釋放記憶體等等)。
ANSI C	assert.h	定義 assert 除錯巨集。
C++	bcd.h	宣告了 C++ 的 bcd 類型、bcd 的超載運算子(overloaded operators) 以及 bcd 的數學運算函數。

	bios.h	宣告用來呼叫 IBM PC ROM BIOS常式的函數。
C++	complex.h	宣告 C++的複數運算函數。
	conio.h	宣告用呼叫 DOS主控台 I/O常式的函數。
ANSI C	ctype.h	包括了字元分類巨集和字元轉換巨集 (例如: isalpha 和 toascii) 所需要的資訊。
	dir.h	包括了處理目錄和路徑名稱所使用的結構、巨集以及函數。
	dos.h	宣告 DOS和 8086-specific呼叫函數並定義所需的常數。
ANSI C	errno.h	定義代表錯誤碼的助憶碼。
	fcntl.h	定義有關函數庫常式 open 的符號常數。
ANSI C	float.h	包括浮點運算常式所需的變數。
C++	fstream.h	宣告檔案輸入輸出所使用的 C++資料流類別。

C++	generic.h	宣告 generic類別的巨集。
	graphics.h	繪圖函數的原型宣告。
	io.h	低階輸入／輸出常式的宣告及所使用的結構。
C++	icmanip.h	宣告 C++資料流 I/O處理器 (manipulators)以及用來建立參數化處理器的巨集。
C++	iostream.h	宣告 C++(version 2.0) 資料流 I/O 常式。
ANSI C	limits.h	包含環境參數，有關編譯時的限制以及整數值大小的限制。
ANSI C	locale.h	宣告提供國家及語言特性資訊的函數。
ANSI C	math.h	數學函數原型的宣告。 定義巨集 HUGE_VAL。 宣告 matherr常式所用到的結構。
	mem.h	宣告記憶體處理函數。(其中有許

多函數的定義也出現在 `stream.h` 中)

	<code>process.h</code>	<code>spawn...</code> 和 <code>exec...</code> 等函數的宣告及其有關的結構。
ANSI C	<code>setjmp.h</code>	宣告 <code>longjmp</code> 和 <code>setjmp</code> 函數及其有關的資料型態 <code>jmp_buf</code> 。
ANSI C	<code>share.h</code>	利用檔案共享 (file-sharing) 的函數所用到參數的定義。
ANSI C	<code>signal.h</code>	<code>signal</code> 和 <code>raise</code> 函數的宣告及其有關常數的定義。
ANSI C	<code>stdarg.h</code>	可以讀取不定數目引數的巨集定義。(如 <code>vprintf</code> , <code>vscanf</code> 等)
ANSI C	<code>stddef.h</code>	定義數個一般的資料型態和巨集。
ANSI C	<code>stdio.h</code>	K&R 一書中標準的 I/O 和 UNIX V 擴展的 I/O 資料型態及巨集的定義。 標準 I/O 預設的資料流: <code>stdin</code> , <code>stdout</code> , <code>stderr</code> 的定義。 資料流層次 I/O 常式的定義。

ANSI C	stdiostr.h	宣告配合 stdio FILE 結構使用的 C++ 資料流。
ANSI C	(stdlib.h)	宣告常用的常式：轉換函數、搜尋／排序常式... 等等。
C++	stream.h	宣告舊版 C++(VERSION 1.2) 資料流 (I/O)常式。
ANSI C	string.h	字串及記憶體的处理常式。
C++	strstrea.h	配合位元組陣列的 C++資料流類別宣告。
	sys\stat.h	開啓及開閉檔案有關符號常數的宣告。
	sys\timb.h	宣告函數 ftime及其傳回的資料型態 timeb結構。
	sys\types.h	宣告有關時間函數的 time_t 型態。
ANSI C	time.h	宣告有關時間轉換函數 asctime, localtime 和 gmtime 等函數的結構。 有關 ctime, difftime, gmtime和

stime 等函數的型態宣告以上各函數的宣告。

value.h

為了與 UNIX V 可相容需定義的重要常數。其中部分常數是與機器種類有關的。

1.4 分類說明函數庫內的常式

BONANZA 登園 (R)

本章將功能相關的函數庫常式擺在一起，並附註由那一個標題檔所宣告，以便讀者參考。有關進一步完整的資料請參考函數庫參考手冊。

1.4.1 分類常式

以下的常式能夠辨認出 ASCII 字元是否屬於文字、控制字元、標點符號或者大小寫等等。

isalnum	(ctype.h)	isascii	(ctype.h)	isdigit	(ctype.h)
isalpha	(ctype.h)	iscntrl	(ctype.h)	isgraph	(ctype.h)
islower	(ctype.h)	ispunct	(ctype.h)	isupper	(ctype.h)
isprint	(ctype.h)	isspace	(ctype.h)	isxdigit	(ctype.h)

1.4.2 轉換常式

以下的常式能夠將數字轉換成字串，或將字串轉換成數字，以及在大小寫之間作變換。

atof	(stdlib.h)	itoa	(stdlib.h)	_tolower	(ctype.h)
atoi	(stdlib.h)	ltoa	(stdlib.h)	tolower	(ctype.h)
atol	(stdlib.h)	strtod	(stdlib.h)	_toupper	(ctype.h)
ecvt	(stdlib.h)	strtol	(stdlib.h)	toupper	(ctype.h)
fcvt	(stdlib.h)	strtoul	(stdlib.h)	ultoa	(ctype.h)
gcvt	(stdlib.h)	toascii	(stdlib.h)		

1.4.3 目錄管理常式

以下的常式有關目錄與路徑名稱的處理。

chdir	(dir.h)	fnsplit	(dir.h)	mkdir	(dir.h)
findfirst	(dir.h)	getcurdir	(dir.h)	mktemp	(dir.h)
findnext	(dir.h)	getcwd	(dir.h)	rmdir	(dir.h)
fnmerge	(dir.h)	getdisk	(dir.h)	searchpath	(dir.h)
setdisk	(dir.h)				

1.4.4 診斷常式

以下的常式提供了內建的偵測錯誤並加以處理的能力。

assert	(assert.h)
matherr	(math.h)
perror	(errno.h)

1.4.5 繪圖常式

以下的常式能夠在螢幕上繪出圖形和文字。

arc	(graphics.h)	fillellipse	(graphics.h)
bar	(graphics.h)	fillpoly	(graphics.h)
bar3d	(graphics.h)	floodfill	(graphics.h)