

北京希望电脑公司计算机技术丛书

Borland C++ 3.1



软件集成技术与范例

蔡明志 編著

海洋出版社

北京希望电脑公司计算机技术丛书

Borland C++ 3.1

软件集成技术与范例

蔡明志 原著

廖彬山等 改编

海 洋 出 版 社

1993年·北京

内 容 提 要

本书详细介绍了 Borland C++ 3.1 的使用方法,并结合具体实例对 C++ 的一些主要特性做了详细阐述。全书共分十章,分别讲述 Borland C++ 3.1 的安装与使用方法、C++ 特性与函数、类与数据封装、运算符重载、继承、虚拟函数与多态性、C++ I/O、模板(Templates)等。本书内容丰富、资料新颖,既可作为广大计算机用户的培训教材,又可作为计算机软硬件开发人员的参考手册。

需要此书的读者可与北京希望电脑公司资料部联系。邮政编码:100080,电话:2562329,通信地址:北京市 8721 信箱。

版 权 声 明

本书繁体字中文版原书名为《Borland C++ 3.1 软件 IC 不是梦》,由松岗电脑图书资料股份有限公司出版,版权归松岗公司所有。本书简体字中文版版权由松岗公司授予北京希望电脑公司,由北京希望电脑公司和海洋出版社独家出版、发行。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

(京)新登字 087 号

Borland C++ 3.1

软件集成技术与范例

蔡明志 原著

廖彬山等 改编

*

海洋出版社出版(北京复兴门外大街 1 号)

海洋出版社发行 双青印刷厂印刷

*

开本: 787×1092 1/16 印张:28.31 字数:637 千字

1993 年 9 月第 1 版 1993 年 9 月第 1 次印刷

印数: 1—5000 册

ISBN 7-5027-3762-6/TP·213

定价:59.00 元

编 者 序

面向对象技术的日益普及导致了許多支持面向对象的程序设计语言的出现。在C语言基础上发展起来的C++语言就是一种支持面向对象开发方法的程序设计语言。由于C++提供了把数据和在数据之上的操作封装在一起的类、对象和方法的机制,并通过派生、继承、重载和多态性等特性,实现了人们追求已久的软件重用和程序自动生成,使得软件,特别是大型复杂软件的构造和维护变得更加有效和容易,并使软件开发能更自然地反映事物的本质,从而大大提高软件开发的效率和质量。

美国 Borland 公司在 Turbo C++ 基础上推出了全新的 Borland C++ 系列软件。其 3.1 版除实现了 AT&T C++ 版本的全部功能外,还支持 ANSI C 和 Microsoft Windows 应用程序的开发,因此已成为国际上最受欢迎的面向对象程序设计软件。

本书详细介绍了 Borland C++ 3.1 的使用方法,并结合具体实例对 C++ 的一些主要特性做了详细阐述。全书共分十章,分别讲述 Borland C++ 3.1 的安装与使用、C++ 特性与函数、类与数据封装、运算符重载、继承、虚拟函数与多态性、C++ I/O、模板(Templates)等。

本书不同于一般的使用手册,注重编程与实例,集知识性与技术性于一体,既可作为广大计算机用户的培训教材,也可作为计算机软硬件开发人员的参考手册。

本书由蔡明志先生编著。改编由廖彬山负责。此外参加本书编写工作的还有刘彬、王强、王林、陈虹、李拥民、谢立文、封明朗、连红兵。本书的录排工作由章强、王晓敏和李小宁负责,在此对她们表示感谢。

改编者

1993 年 9 月

目 录

第一章	Borland C++ 3.1 安装与使用	1
1.1	起步:安装 Borland C++ 3.1 & Application Frameworks	1
1.2	设置 Borland C++ 3.1 在硬盘中的目录	2
1.3	设置 Borland C++ 3.1 & Application Frameworks 的安装	2
1.4	其它安装选项的设置	2
1.5	开始安装	3
1.6	进入 Borland C++ 3.1	3
1.7	如何利用 IDE 来设计与编写程序	4
1.8	存储程序	5
1.9	执行程序	5
1.10	利用 Project 文件来编译并开发大型的应用程序	5
1.11	一些好用的功能	6
1.12	进入 Windows 3.x	7
第二章	C++特性	31
2.1	注解(comments)	31
2.2	C++的简单输出与输入	32
2.3	C++的动态存储器分配	34
2.4	定义与声明	35
2.5	参考类型(reference type)	41
2.6	范围运算符(Scope resolution operator)	44
2.7	const 变量	46
2.8	void 指针	47
2.9	sizeof 运算符	48
2.10	C++中所定义的与关键字	48
2.11	C++的结构(structure)数据类型	48
第三章	C++函数	53
3.1	前言	53
3.2	函数原型(function prototype)	53
3.3	为什么需要函数原型	54
3.4	函数信息的传递	57
3.5	Inline 函数	79
3.6	Inline 函数不是宏(Macros)	82
3.7	函数缺省参数初值(Default Arguments)	83
3.8	函数名称的重载(Overloaded Function)	86
3.9	函数重载(Function Overloading)时可能发生的问题	89

3.10	函数指针(Pointers to functions)	90
3.11	小结	96
第四章	类与数据封装(Class & Encapsulation)	97
4.1	什么是类(class)	97
4.2	什么是对象(object)	97
4.3	如何设计出一个类	98
4.4	Constructor 构造函数	111
4.5	Destructor 构造函数	113
4.6	成员初设序列(member initializer list)	116
4.7	Constructor 与成员初设序列的深入探讨	122
4.8	Copy constructor 复制建立者	127
4.9	一个特殊的类类型—Structure	138
4.10	小结	140
第五章	类的应用	141
5.1	类的友元(Friend Function)	141
5.2	类的静态成员(Static class members)	146
5.3	this 指针	153
5.4	类成员指针(Pointers to Class Members)	160
5.5	类数组	168
5.6	面向对象程序开发	173
5.7	小结	177
第六章	运算符重载(Overloading Operators)	179
6.1	数据类型的转换	211
6.2	设计转换函数时容易发生的问题	224
6.3	小结	228
第七章	继承(Inheritance)	229
7.1	基础类与派生类(Base class & derived class)	229
7.2	基础类下的数据隐藏	230
7.3	派生类的定义方法	243
7.4	公用(Public)基础类与私用(Private)基础类	244
7.5	派生类之数据成员与成员函数的定义与使用	245
7.6	继承下派生类的 constructor 的设计	249
7.7	派生类的 constructor 的进一步讨论	253
7.8	派生类的成员函数及数据成员与类范围的关系	259
7.9	Overriding functions 并非 overloading functions	272
7.10	类的 friend 没有被继承的性质	273
7.11	扩展您的程序	273
7.12	多重继承下的不确定性	287
7.13	小结	292

第八章	虚拟函数与多态性(Virtual function and Polymorphism)	293
8.1	派生类与基础类的转换	293
8.2	静态连结(Static Binding)与动态连结(Dynamic Binding)	297
8.3	虚拟函数(Virtual Function)	301
8.4	如何定义虚拟函数	306
8.5	虚拟函数的定义	308
8.6	虚拟函数的调用	310
8.7	虚拟函数与继承的关系	320
8.8	虚拟函数的数据封装	328
8.9	两个特殊的虚拟函数	331
8.10	虚拟 destructor	336
8.11	另一种虚拟观念的应用——虚拟基础类(Virtual Base Class)	341
8.12	小结	351
第九章	再论 C++ 的 I/O(Input/Output)	353
9.1	iostream.h 的内容	353
9.2	类 ios 的其他成员函数与 Manipulators	369
9.3	其他的格式标志	372
9.4	派生类 ostream 与输出(Output)	375
9.5	派生类 ostream 中有关输出的其他成员函数	375
9.6	输出运算符 << 的 Overloading	377
9.7	派生类 istream 与输入(Input)	384
9.8	派生类 istream 中有关输入的其他成员函数	384
9.9	输入运算符 >> 的 Overloading	386
9.10	有关文件的 I/O	391
9.11	ofstream 与文件的输出	391
9.12	ifstream 与文件的读取	392
9.13	文件数据流的操作	392
第十章	Templates	413
10.1	Template 函数	413
10.2	如何定义 template 函数	415
10.3	template 函数的 overloading	421
10.4	特殊状况的处理	424
10.5	多重类型参数	428
10.6	Template 类	431
10.7	Template 类的定义	431
10.8	Template 类的 friend 函数	438
10.9	特殊化的 tempate class	443
10.10	小结	445

第一章 Borland C++ 3.1 安装与使用

1.1 起步：安装 Borland C++ 3.1 & Application Frameworks

首先将 Borland C++ 3.1 第一张软盘置入 A 或 B 盘中，然后在 DOS 下键入：

A: \>INSTALL

按 Enter 键后，便开始安装 Borland C++ 3.1。首先会见到图一的情形。

在图一中，你将可以看见，如果是安装 Borland C++ 3.1 & Application Frameworks，则硬盘空间至少需要49MB，并外加5MB的工作空间。如果确定有如此大的硬盘空间，则请再按下 Enter 键继续以下的安装程序，否则请按 Esc 键回到 DOS，将硬盘“清干净”些再说吧！按下 Enter 键后，则进入图二步骤。

在图二中须告诉安装程序在安装时，将使用哪一台磁盘机来读入 Borland C++ 3.1 的原版软盘至硬盘中，也就是所谓的源磁盘机 (SOURCE drive)，输入 A 或 B，然后按 Enter 键。读者可以由屏幕下方的信息区知道当前安装程序正要求你做哪些动作。

(如果读者在安装的过程中发现有任何不当的情形，只要按下 Esc 键，即可中止安装程序，回到 DOS 环境下。)

输入磁盘机名称后 (如图二中的 A)，按 Enter 键，便可进入如图三的步骤。

图三的部分是整个安装过程中最重要的一环，你必须耐心而仔细地地完成每一部分，这样 Borland C++ 3.1 才会有最佳的执行状况。

在图三，中央部分 (白底黑字) 为工作区，在工作区中必须逐一地选择各个安装项目，然后进行设置的工作，直到每一项都完成设置，再选择工作区中最下方的 Start Installation 项目，开始将 Borland C++ 3.1 由软盘安装到硬盘中。

图三的最下方有一行信息说明了可使用按键的功能。

F1: 帮助键

如果你安装至某个选项而不知该如何继续时，除了再仔细阅读中央下方蓝底白字的信息区外，也可以按下 F1 键而得到额外的帮助信息，相信可以顺利地引导你完成安装程序。

F9: 开始安装

除了上述利用工作区中的 Start Installation 选项来真正进入安装步骤外，也可以直接按下 F9 键来进行。

Enter: 选择安装项目

在图三中您应可以发现工作区中有一个可移动的光条，将此光条利用键盘的方向键移动到某个选项，按下 Enter 键即可选择或设置该项目。

Esc: 回到上一个安装步骤

利用 Esc 键可以回到上一个安装程序的画面。

1. 2 设置 Borland C++ 3. 1 在硬盘中的目录

将工作区中的黑色光棒移动到安装选项 Directories... 上 (如图三的情形), 按下 Enter 后, 便进入目录设置步骤, 如图四。

一般而言, 我们不需要更改任何设置, 但如果您希望安装在 D 或其它的硬盘或目录中, 则只要将黑色光棒移动到 Borland C++ 3. 1 Directory 项 (如图四), 按下 Enter, 然后输入新的目录, 如图五。

将可以发现安装程序为您做了整体性的更改, 如图六中的各目录都已改成安装在硬盘 D。

回到图三的安装画面, 如果您已安装 Windows 3. 0 或 3. 1 在硬盘上, 也请设置其所在的正确目录 (缺省为 C: \WINDOWS), 这样程序将会把 Borland C++ 3. 1 安装在 WINDOWS 环境下, 这样一来, 您将可以在 DOS 或 Windows 下使用 Borland C++ 3. 1。将黑色光棒移动到 Windows Dir... 选项上, 按下 Enter 键后再输入您的 Windows 所在目录即可, 如图七。

至于其它部分, 则能不改就不要改是最好不过了。

1. 3 设置 Borland C++ 3. 1 & Application Frameworks 的安装

将黑色光棒移动到 Install Options... 的选项上, 此时您将会发现屏幕中下方的信息区会显示 Borland C++ 3. 1 & Application Frameworks 中各部分安装时所需要的硬盘空间, 如果您不希望全部安装以节省硬盘空间, 则按下 Enter 键后, 进入图八所示的步骤。

每个选项的最右方都将呈现 Yes/No 的状态, 其中 Yes 代表该项将被安装至硬盘, 如果您希望取消某选项的安装, 只要移动光棒至该项上, 按下 Enter 键, 将原为 Yes 状态的选项转为 No 的状态即可。再按一次则可再恢复成 Yes 的状态, 如图九中的状况便表示共有三项应用软件不会被安装到硬盘。

1. 4 其它安装选项的设置

另外还有三项主要的安装设置:

Examples Options: 设置要安装哪些范例程序。

Windows Options: 设置要安装哪些软件到 Windows 下。

DOS Library Models: 设置要安装哪几种模块大小的 DOS Library (函数库)。

这三个安装项目的设置方式可以参照上述第三点的设置方法。

1.5 开始安装

完成工作区内的所有设置后,按下 F9 或将工作区中的光棒移至 Start Installation 项,如图十。

接着便出现如图十一的安装画面,您只要耐心地等待,然后按照安装程序的指示顺序将 Borland C++ 3.1 或 Borland C++ 3.1 & Application Frameworks 的软盘置入源磁盘机即可。如图十二,便要求您将第二片软盘置入 A 磁盘机中。

在漫长的等待后,如果有幸看到图十三的画面,则表示已经顺利将 Borland C++ 3.1 或 Borland C++ 3.1 & Application Frameworks 依照设置安装至硬盘上而可以开始使用了。不过您首先得浏览安装程序为您安排的 readme 文件,见图十四。

但如果读者急欲瞧瞧安装的成果,则按下 Esc 键先离开 readme 文件回到 DOS,等以后有空再仔细好好阅读。

1.6 进入 Borland C++ 3.1

安装 Borland C++ 3.1 后,作者便要带领读者进入 Borland C++ 3.1 的集成开发环境,也就是所谓的 IDE。

首先进入您安装 Borland C++ 3.1 硬盘的磁盘机代号,如 C: 或 D: 然后按图十五的方式进入 Borland C++ 3.1 的子目录(如果当初没有更改缺省目录,则情形便如图十五一样),然后键入 bc 即可进入 Borland C++ 3.1 的集成环境。然而 bc 命令后还可加上许多选择性的参数,如果您不太熟悉,则可利用 bc/? 得到详细的辅助说明,见图十五。

当您第一次进入 Borland C++ 3.1 时,会出现如图十六的情形,为 Borland C++ 3.1 的版权画面。

按下任何一键后便可正式进入 Borland C++ 3.1 的 IDE,如图十七。

在图十七中,最上方为主功能选单表,中部分为程序编辑区或工作区,最底部一行为信息区。

如果您对于菜单中的选项用法不太明了,则可以将光棒移动至该选项,然后仔细阅读信息区中的文字。若还想进一步的了解该选项的意义或用途,则可以按下 F1 键,立刻会出现有关该功能选项的辅助信息。

举例说明:假设读者不了解如何开始一个新的程序文件,则可以将光棒移至 File/New 的位置,然后查看屏幕下的信息区,如图十八。

或者再按下 F1 键,以便得到更详尽的辅助信息,见图十九。

要关闭辅助信息窗口只要按下 Esc 键,或将鼠标器光标(如果您有鼠标器)移到视窗左上角的关闭钮上,按下鼠标器左键即可。

1. 7 如何利用 IDE 来设计与编写程序

要设计程序,首先必须利用 File/New 或 File/Open 功能来打开文件,假使我们用 File/New 功能,则会出现以下图二十的情形。

在图二十中,读者可以发现工作区中多出了一个窗口,该窗口代表我们打开的程序文件,缺省文件名为 NONAME00. CPP,其中. CPP 代表该程序文件为 C++ 语言所编写,IDE 将以

C++ 的语法来编译 (Compile) 此程序。每个程序编辑窗口的左上角皆会有一个关闭钮,用来关闭窗口并使之由屏幕上消失。而程序编辑窗口的右上角则有一个指向上的箭头,代表极大钮,可以将编辑窗口的编辑区放至最大。在极大钮的左方还有一个数字,当前显示为1,这是窗口打开的顺序,由于 Borland C++ 3.1 的 IDE 一次可以编辑多个程序文件,因此每一个打开在工作区中的编辑窗口都含有一个代表数字,由1到9。如果工作区中打开超过九个编辑窗口,则从第十个编辑窗口以后不再给予编号。利用 Alt+窗口编号,可以直接将作用编辑窗口移到指定的窗口上。

所谓作用编辑窗口指可以输入文字、编写程序以及编译程序的窗口,工作区中可以同时打开多个程序文件,每一个文件具有一个独立的编辑窗口,但一次只能有一个编辑窗口可以被使用来编写或编译程序。如图二十一中,同时打开两个编辑窗口,但只有编号为2,也就是程序 NONAME01. CPP 才是使用编辑窗口。

让我们再回到图二十的状况中。请按下面图二十二的情形将程序输入到窗口中:

现在我们要开始将图二十二中的程序编译成在 DOS 下的可执行文件,在进行编译前,必须先完成几项工作。

首先选择 Options 功能下的 Application 选项,如图二十三。

接着会出现图二十四的情形。

请确定是否您的屏幕也同图二十四中的内容完全相同,如果不是则请选择最左方的 [Dos Standard] 选项,这是设置我们的程序要编译成哪一种环境下的可执行文件,如果您想设计在 Windows 下的执行文件,则可以选择 [Windows App] 或 [Windows DLL]。

如果您现在的状况与图二十四中相同,则请选择 OK。接下来请在选择 Options 下的 Directories 选项,如图二十五。

紧接着会出现图二十六的情形。

图二十六的情形是要我们设置编译程序时,所需要的包含文件 (included files) 与函数库所安装的目录。如果系统在 Borland C++ 3.1 安装后并无任何改动,则您应该不必在此重新设置,但如果系统或目录有所变更,特别是硬盘的磁盘机代号如果已经更改,则请重新设定装入文件与函数库所在的正确目录,一切设置完成后,请选择 OK。

现在我们可以开始编译程序,选择 Compile 下的 Compile 选项,或直接按下 Alt+F9 即可,然后 IDE 便开始编译程序,同时屏幕上还会出现如图二十七中的编译状态窗口。

编译状态窗口的底部在编译终止时会告诉您是否已经编译成功。如果失败,则出现 Error: Press any key 的信息,且编译窗口中也会指出有几个错误,如图二十八中便指示出有二个错误。

按下任一键后随即会出现指示错误与警告的信息窗口 (Message),在 Message 窗口中,如

图二十九，可以看见程序发生错误的地方。

假使我们已经将程序编译成功，再来便是要将程序编译后产生的目标文件（.OBJ）与函数库链结（Link）成可执行文件。选择 Compile 下的 Link 功能，如图三十，便是程序链结成功的情形。

完成以上的步骤，您已经能写出一个可以在 DOS 下执行的程序了。（利用 Compile 下的 Make 功能，或直接按下 F9 键，则可以一次完成程序的编译与链结）

1. 8 存储程序

辛苦设计出的程序，如果没有存储下来，相信会有不太好受的感觉。Borland C++ 3.1 的 IDE 中提供了三个存储文件的方法。

1. File/Save 或利用 F2 键。

存储当前作用窗口中的程序，其文件名便是该使用窗口上所显示的文件名。

2. File/Save as。

存储当前作用窗口中的程序，但希望更换文件名，因此会出现如图三十一的窗口并要求您输入新的文件名。

3. File/Save all。

存储工作区中所有曾经被更改的文件。

1. 9 执行程序

可以有两种方法来执行您设计的程序而不必离开 Borland C++ 3.1 的集成开发环境：

1. 先利用 File 下的 DOS shell 功能，暂时先回到 DOS 环境下，然后执行您的程序（已经编译及完成链结），等程序执行结束后，在 DOS 下键入 exit 指令则可以立刻回到 IDE 中。
2. 直接在 IDE 中按下 Ctrl+F9，则 IDE 便会为您执行当前作用编辑窗口中的程序（如果已经产生可执行文件），然后再按下 Alt+F5 来观察程序的结果（如果程序中有输出信息至屏幕上）。

1. 10 利用 Project 文件来编译并开发大型的应用程序

以 Make 的方式来编译当前作用窗口的程序文件进而产生可执行的应用程序，虽然极为方便，但是唯一的缺点便是一次只能编译一个文件。

对于一个大型应用程序的开发，相信没有任何程序设计者会将所有的程序码集中在同一个文件中然后进行编译与链结。优秀的程序设计者应会尽量将应用程序的开发分成若干文件来编写，并分别编译，最后再链结在一起产生可执行的文件。但是这便造成编译与链结的过程过于复杂。于是 Borland C++ 3.1 之 IDE 提供了 Project 文件（或称为计划文件）的功能以解决这个困扰，使得在 Borland C++ 3.1 中设计应用程序更为得心应手。

假设有一个应用程序共分为七个程序文件来开发,则设计者便可以建立一个 Project 文件来编译并链结这七个程序文件。

首先选择 Project 选单下的 Open project...选项,准备打开一个 project 文件,如图三十二的情形。

图三十二中,在此窗口内您必须输入一个要打开的 project 文件名称,附加文件名缺省为 .PRJ,窗口内的 Files 栏中列出所有已经建立的 project 文件,假设程序设计者产生一个新的 project 文件并命名为 NEW.PRJ,如图三十三。

输入完成后,选择 OK,接下来会出现如图三十四的画面。

打开新的 project 文件时,起初并无任何程序文件,现在设计者可以开始将应用程序所需要的程序文件加入到 NEW.PRJ 中,选择 Project 下的 Add Item...选项或直接按下 Ins 键,则立即会出现图三十五的情形。

利用图三十五的窗口,设计者可以将所有为开发应用程序所编写的程序文件加入到 NEW.PRJ 中,全部加入后,选择 Done,接着便可以见到如图三十六的画面。

与图三十四比较,读者将可以发现 project 文件 NEW.PRJ 中已经具有开发应用程序时所需要的程序文件了,以后设计者只要执行 Make 或 Compile、Link,则 IDE 便会依据当前打开的 project 文件中有哪些程序文件,然后逐一进行编译或链结,最后产生一个与 project 文件同名称的可执行程序。

利用 Project/Delete item (或直接按下 Del 键)可以用来删除 project 文件中的多余的程序文件。

此外必须特别注意,由于 IDE 中一次只能打开并使用一个 project 文件,因此当您希望编译其他程序文件时,必须先关闭当前使用中的 project 文件,因为当 IDE 中某个 project 文件被打开时,则编译器 (Compiler) 永远不会去理会不包含在此 project 文件内的程序文件。

1. 11 一些好用的功能

下面作者将介绍在 Borland C++ 3.1 IDE 中一些特别好用的功能选项:

1. Edit/Undo 与 Edit/Redo.

Edit/Undo 可以恢复您在编辑窗口中所作过的任何改变,而 Edit/Redo 则是把利用 Edit/Undo 功能所恢复的部分再恢复回来。

2. Edit/Copy example.

Borland C++ 3.1 IDE 中,辅助信息内藏着丰富的范例程序,利用此功能选项可以将这些范例程序复制到剪贴板 (Clipboard) 中,然后再利用 Edit/Paste 功能便可以将复制到剪贴板中的范例程序传送到您的编辑窗口中。

如图三十七中,右下方为从辅助信息中所找到的有关函数 exit 如何使用的范例程序,这时便可以利用 Edit/Copy example 功能将此范例程序复制到剪贴板中。

3. Compile/Information...

Information 的功能早在 2.0 版中便已经出现,但现在的 Information 选项则移至 Compile 选单下。利用 Information 功能可以查看当前的 IDE 的所有状况,如图三十八。

4. Window/Tile 与 Window/Cascade.

这两个功能皆是用来将工作区中的编辑窗口排列整齐。Tile 功能是以不重叠的方式排列，如图三十九的情形。

而 Cascade 则是以重叠的方式排列编辑窗口，如图四十。

当您一次编辑多个程序文件时，由于不断地搬移与切换编辑窗口，很容易弄乱窗口的排列，造成编辑上与浏览上的不便。利用此二功能选项可以随时将您工作区中所有的编辑窗口排列整齐。

1. 12 进入 Windows 3. x

如果您在硬盘上已经安装 Windows 3. x，且在安装 Borland C++ 3.1 & Application Frameworks 时曾设置要安装在 Windows 3. x 中，则在安装 Borland C++ 3.1 成功后，再进入 Windows 3. x 时（以下将以 Windows 3.1 为例）将会出现如图四十一的信息画面。

这个信息中要求您决定是否要让 Windows 3.1 自动替您产生 Borland C++ 3.1 的程序组 (program groups)，请按下 OK 钮，稍待一会儿便可以看到建立好的 Borland C++ 3.1 程序组及其各项程序的图标 (ICONS)，如图四十二。

在 Borland C++ 3.1 的程序组中，执行 Borland C++ 标像可以进入先前所讨论的在 DOS 之下的 Borland C++ 3.1 IDE。

注意！如果您执行后无法顺利进入 Borland C++ 3.1 的 IDE，则很可能是路径的设置发生问题，此时您必须修改 BC.PIF 文件中的设置，如图四十三的设置状况，在 Program File-name 的设置中明确指出 BC.EXE 执行文件的安装路径：

（关于 PIF 文件的修改方法，请参考有关 Windows 3. x 的操作手册）

另外在程序组中的还有 BORLAND C++ FOR WINDOWS (请执行 BCW 图标)，如图四十四，以及 Resource Workshop (请执行 Workshop 图标)，这些应用程序对开发 Windows 3.1 环境下的应用程序都非常好用。图四十五即为 Resource Workshop 的进入后的画面。

好了，现在您可以开始慢慢地享用辛苦安装完成的 Borland C++ 3.1 & Application Frameworks，并准备开始进入本书的重点：面向对象程序设计。

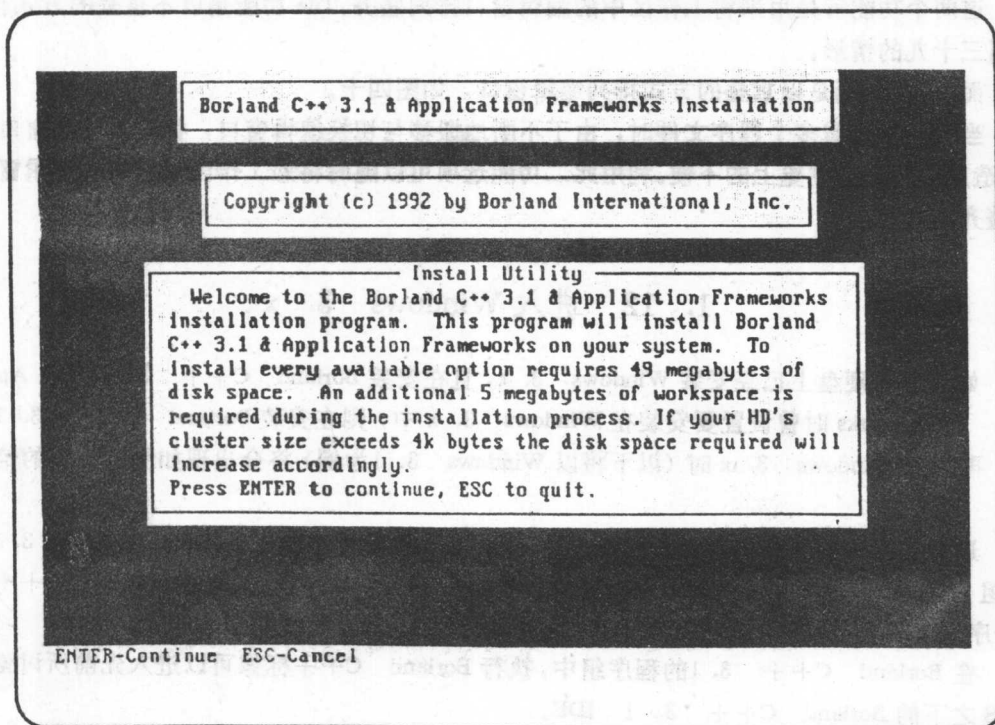


图 一

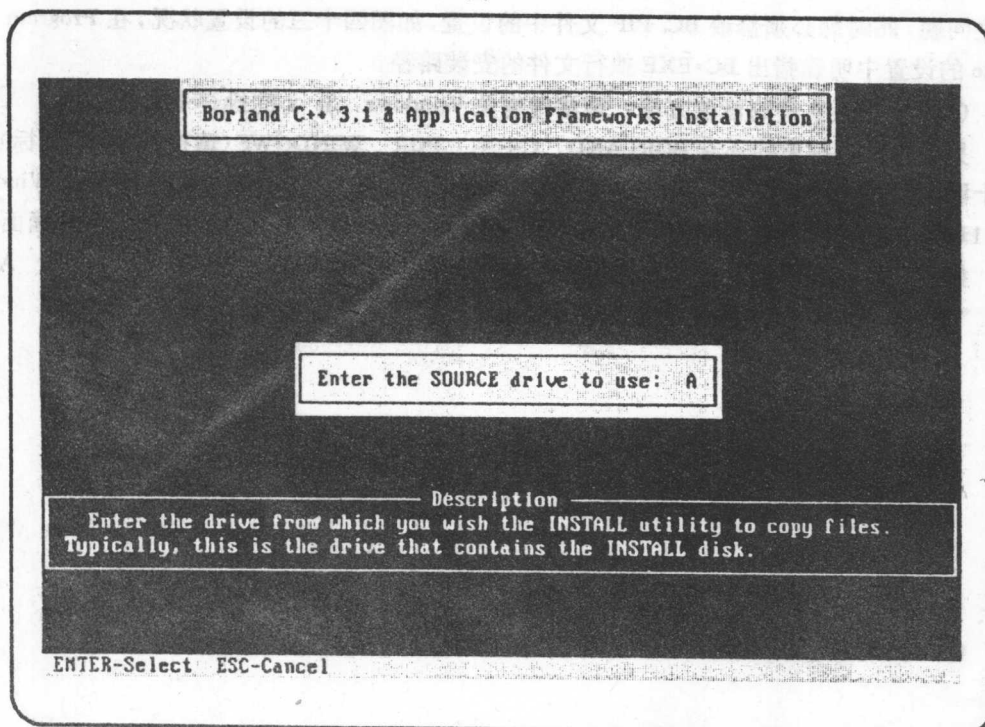


图 二

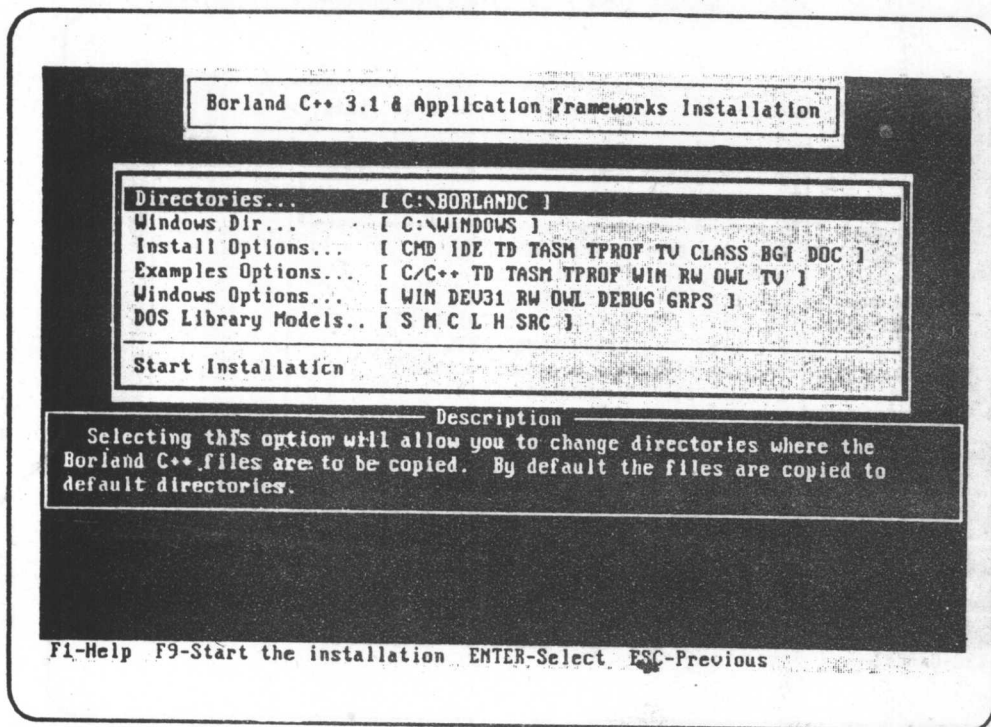


图 三

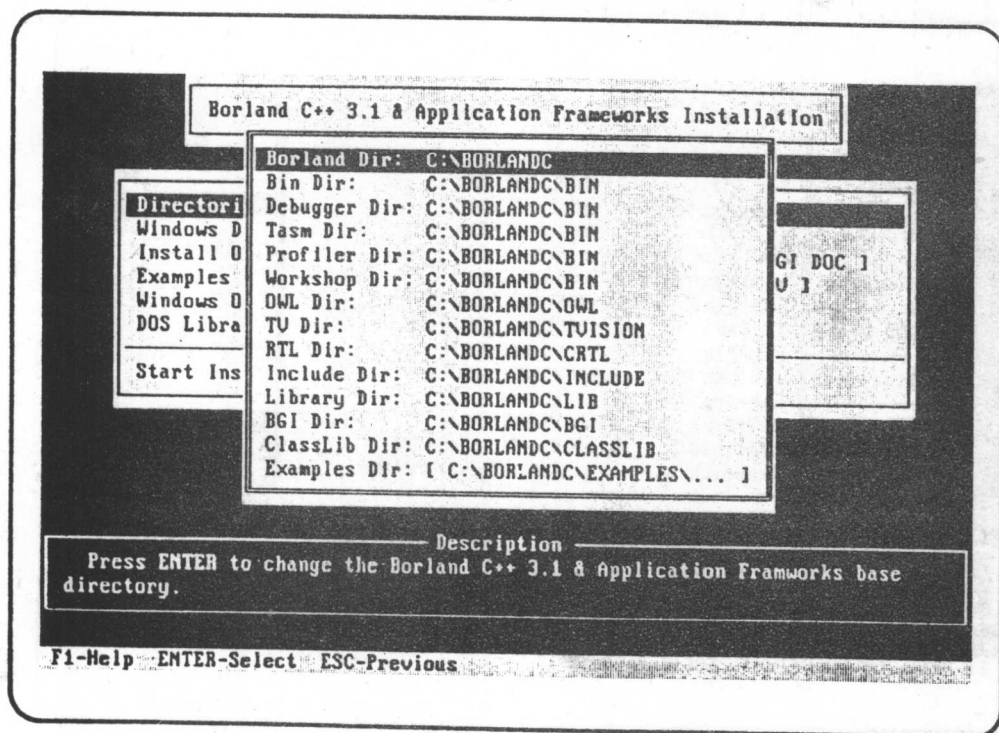


图 四

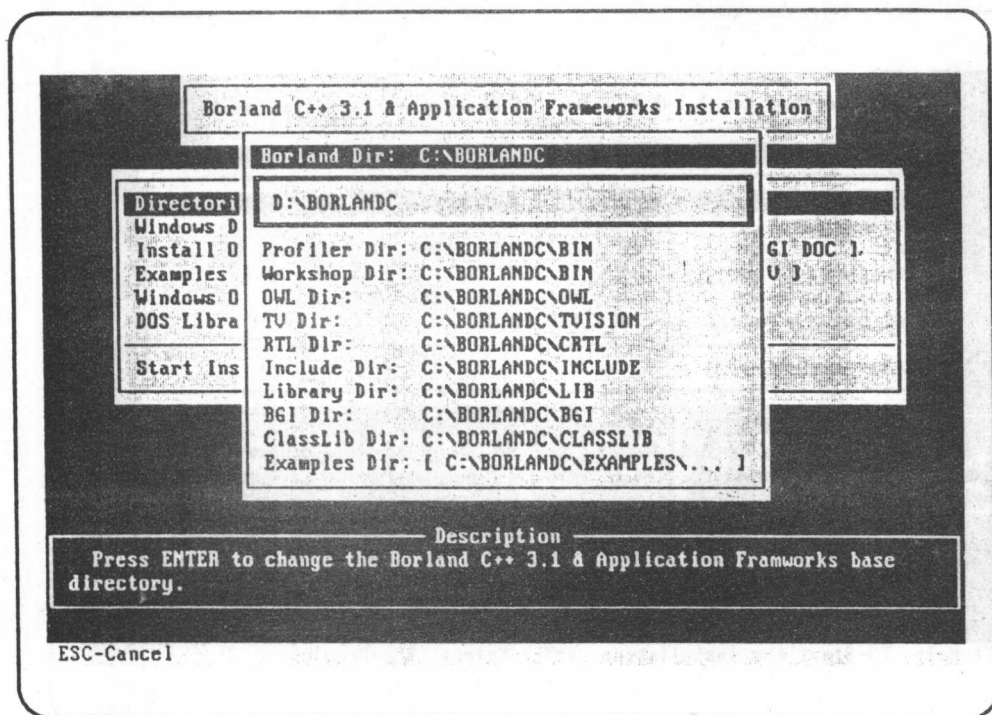


图 五

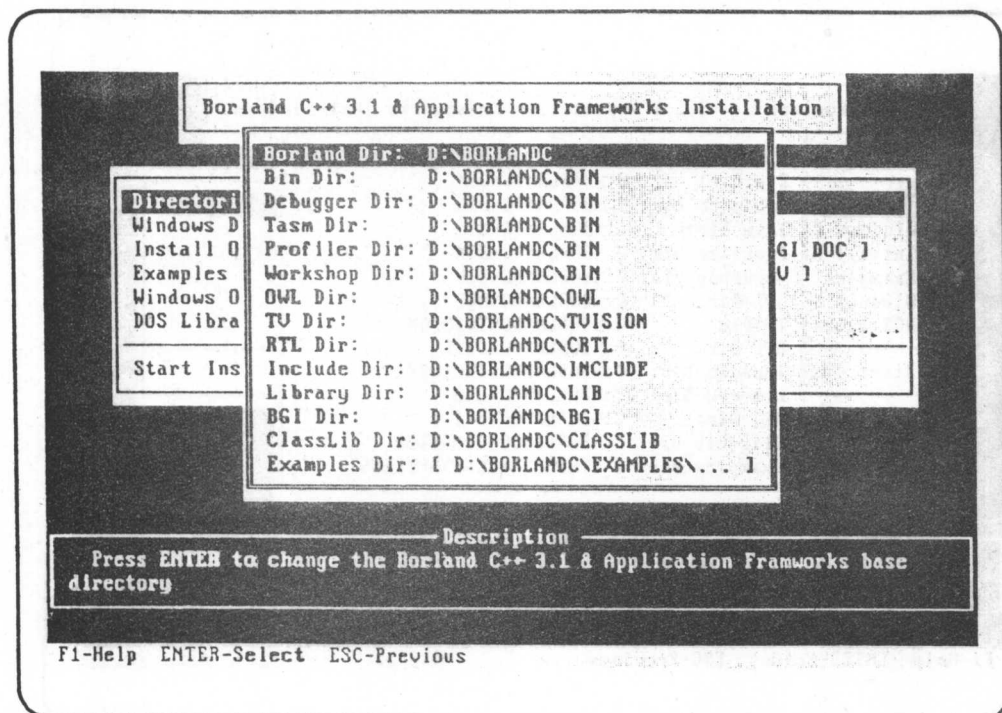


图 六