

内附光盘

范例的源程序代码及结果文件，让您学习更方便

Visual C++

程序设计实务入门

庄益瑞 梁仁楷 编著

- 本书以 Microsoft Visual C++ 为工具，运用较多的范例详细讲解 C++ 程序的设计过程及技巧
- 每一章都配有精心设计的习题，读者可以循序渐进体会 C++ 程序设计的精髓，非常适合初次学习程序语言或想彻底了解 C++ 语言的读者阅读



中国铁道出版社

CHINA RAILWAY PUBLISHING HOUSE

Visual C++

程序设计实务入门

庄益瑞 梁仁楷 编著



中国铁道出版社

2001年·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2001-1327 号

版 权 声 明

本书中文繁体字版由台湾松岗电脑图书资料股份有限公司出版(2000)。本书中文简体字版经台湾松岗电脑图书资料股份有限公司授权由中国铁道出版社出版(2001)。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

图书在版编目(CIP)数据

Visual C++ 程序设计实务入门 / 庄益瑞, 梁仁楷编著. —北京: 中国铁道出版社, 2001. 4

ISBN 7-113-03943-X

I. V… II. ①庄…②梁… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 19402 号

书 名: Visual C++ 程序设计实务入门

作 者: 庄益瑞 梁仁楷

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街 8 号)

策划编辑: 苏 茜

特邀编辑: 袁秀珍

封面设计: 冯龙彬

印 刷: 北京市兴顺印刷厂

开 本: 787×1092 1/16 印张: 27.5 字数: 666 千

版 本: 2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-03943-X/TP·479

定 价: 45.00 元

MSZP1/01

版权所有 盗版必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

出版说明

本书以适合初学者阅读的浅显文字，配合渐进式的范例引导练习，再辅以详细的图解，精确阐述 C++ 程序设计的精华，适合读者自学或作为面向对象程序设计的教科书。书中每章结尾都有配合章节内容设计的自我突破习题，可作为自我练习或课后实习的题目，并附有简单的提示作为解题的参考。随书所附光盘的内容为书中完整范例文件及结果文件。

本书由台湾松岗电脑图书资料股份有限公司提供版权，经中国铁道出版社计算机图书项目中心审选，王小海、周延、易鹏博、彭涛、宁夕、何伟、邓雄、王涛、李谨、严尔顺、廖康良、萧志军、孟丽花等同志完成了本书的整稿及排版工作。

中国铁道出版社

2001 年 6 月

Visual C++ 程序设计实务入门

目 录

第1章 C语言的基本概念.....	1
1-1 程序语言.....	1
1-1-1 为什么需要程序语言.....	1
1-1-2 高级语言.....	2
1-2 C语言的演进.....	3
1-3 Microsoft Visual C++ 6.0 的安装.....	3
1-4 启动程序与画面简介.....	16
1-4-1 启动 Visual C++	16
1-4-2 Visual C++画面简介.....	17
1-4-3 启动 MSDN Library	19
1-4-4 MSDN Library 画面简介.....	19
1-5 C语言程序结构.....	21
1-5-1 程序设计的步骤.....	21
1-5-2 C语言程序结构.....	22
1-6 如何编写C语言程序.....	25
1-6-1 C语言程序的格式.....	25
1-6-2 编译式语言 and 解释式语言.....	25
1-6-3 Visual C++程序的编写与编译方法.....	26
温故知新.....	29
自我突破习题.....	29
第2章 数据类型与表达式.....	31
2-1 C语言的数据类型与变量声明.....	31
2-1-1 常量与变量.....	31
2-1-2 整型.....	32
2-1-3 浮点型.....	33
2-1-4 字符型.....	34
2-2 输入与输出.....	35
2-3 运算符和表达式.....	40
2-3-1 数学运算表达式.....	40
2-3-2 赋值表达式.....	41
2-3-3 逻辑关系表达式.....	42
2-3-4 位运算表达式.....	43
2-3-5 数据类型的转换与强制转型运算符.....	47

2-3-6 运算符优先级.....	48
温故知新.....	54
自我突破习题.....	54
第3章 程序流程控制	57
3-1 循环结构.....	57
3-1-1 for 语句.....	57
3-1-2 while 语句.....	66
3-1-3 do...while 语句.....	71
3-2 条件流程控制.....	75
3-2-1 if 语句.....	75
3-2-2 else if 语句.....	78
3-3 多重分支结构 (switch...case).....	81
3-4 中断分支与循环跳转语句.....	87
3-4-1 中断分支 break 语句.....	87
3-4-2 循环跳转 continue 语句.....	90
3-5 条件运算符 (? :).....	95
3-6 无条件跳转语句 (goto).....	99
温故知新.....	100
自我突破习题.....	101
第4章 数组与指针	105
4-1 数组.....	105
4-1-1 数值数组的赋值.....	105
4-1-2 字符数组的赋值.....	112
4-2 多维数组.....	117
4-2-1 数值二维数组的赋值.....	117
4-2-2 字符二维数组的赋值.....	123
4-3 指针.....	125
4-3-1 指针的意义.....	125
4-3-2 指针的运算.....	126
4-4 动态内存配置.....	129
4-4-1 new 配置内存.....	129
4-4-2 delete 释放内存.....	133
4-5 指针与数组.....	135
4-5-1 指针与一维数组.....	135
4-5-2 指针与多维数组.....	138
4-5-3 指针与字符数组.....	139
温故知新.....	140
自我突破习题.....	141

第 5 章 函数与函数库	145
5-1 函数的定义	145
5-2 参数传递	147
5-2-1 传值调用	149
5-2-2 传址调用	151
5-3 C++常用函数	153
5-3-1 数学函数	154
5-3-2 随机数函数	159
5-3-3 字符串函数	162
5-3-4 数据转换函数	165
5-3-5 输入输出函数	169
5-4 递归函数 (Recursive Functions)	176
5-5 内联函数 (Inline Functions)	179
温故知新	180
自我突破习题	180
第 6 章 特殊数据类型	185
6-1 枚举 (enum)	185
6-2 类型定义语句 (typedef)	188
6-3 结构体 (struct)	189
6-3-1 结构体的定义与引用	189
6-3-2 函数中结构体数据的传递	193
6-4 预处理指令 (Preprocessor Directives)	198
6-4-1 #define 与 #undef	198
6-4-2 #include	200
6-4-3 #if...#else...#endif 与 #elif	204
6-4-4 #ifdef 与 #ifndef	206
6-5 变量的作用范围	206
6-5-1 自动变量 (auto)	207
6-5-2 静态变量 (static)	208
6-5-3 外部变量 (extern)	210
6-5-4 静态外部变量 (static extern)	212
6-5-5 寄存器变量 (register)	213
温故知新	213
自我突破习题	213
第 7 章 类与对象	217
7-1 面向对象基本概念	217
7-1-1 什么是对象	217
7-1-2 面向对象设计的优点	219

7-2 加上行为的结构体一类(class)	219
7-2-1 类的意义	219
7-2-2 类的设计方法	221
7-2-3 类中成员的种类	224
7-2-4 类的聚集	226
7-3 构造函数(Constructor)与析构函数(Destructor)	228
7-3-1 构造函数的基本概念	228
7-3-2 设定对象的初始值	229
7-3-3 构造函数重载	239
7-3-4 析构函数(Destructor)	240
7-4 对象的定义与操作	241
7-4-1 对象的定义	241
7-4-2 对象的赋值	241
7-4-3 动态配置对象	243
温故知新	244
自我突破习题	245
第8章 继承与类派生	249
8-1 类派生	249
8-1-1 继承的概念	249
8-1-2 类派生—公有派生	252
8-1-3 类派生—私有派生	258
8-1-4 类派生—保护派生	260
8-1-5 派生的原则	261
8-1-6 派生类的构造函数与析构函数	262
8-2 多重继承	264
8-2-1 基本概念	264
8-2-2 构造函数的定义	265
8-2-3 成员函数的访问	267
8-3 类的层次结构	268
8-3-1 单一类层次结构	268
8-3-2 多重类层次结构	272
8-4 虚基类	274
8-4-1 使用时机	274
8-4-2 构造函数的调用	276
8-4-3 析构函数的调用	281
8-4-4 虚基类成员的调用	282
温故知新	283
自我突破习题	284

第 9 章 虚拟函数与友元函数	289
9-1 多态	289
9-2 虚函数	290
9-2-1 定义与声明	290
9-2-2 访问关系	293
9-2-3 动态链接	300
9-2-4 纯虚函数	304
9-3 友元函数与友元类	306
9-3-1 友元函数	306
9-3-2 友元类	308
9-4 this 指针	311
温故知新	313
自我突破习题	314
第 10 章 重载	317
10-1 函数重载	317
10-1-1 一般函数重载	317
10-1-2 构造函数重载	318
10-1-3 拷贝构造函数	322
10-2 运算符重载	326
10-2-1 运算符函数	326
10-2-2 二元运算符的重载 (一)	328
10-2-3 二元运算符的重载 (二)	333
10-2-4 一元运算符的重载	338
10-2-5 等号运算符的重载	341
10-3 类型转换	343
10-3-1 利用构造函数进行类型转换	344
10-3-2 转换成其它类型	346
温故知新	347
自我突破习题	347
第 11 章 模板	353
11-1 什么是模板	353
11-2 函数模板	355
11-2-1 应用一般数据类型的函数模板	355
11-2-2 应用自定义类型的函数模板	358
11-3 类模板	362
11-3-1 定义及使用方法	362
11-3-2 具有固定类型的类模板	364
11-3-3 一个类模板范例——串行队列	367

温故知新	380
自我突破习题	380
第 12 章 输入与输出流	383
12-1 流	383
12-2 输出流 cout 的运作	385
12-2-1 串联输出	385
12-2-2 字符与字符串输出	386
12-2-3 清除缓冲区 flush()	386
12-2-4 设定输出格式	387
12-3 输入流 cin 的运作	392
12-3-1 串联输入	392
12-3-2 单一字符输入 get()	392
12-3-3 字符串输入 getline() 和 ignore()	394
12-4 文件的输入与输出	396
12-4-1 简单的文件 I/O 处理	397
12-4-2 高级文件处理	400
12-4-3 随机存取文件	412
温故知新	417
自我突破习题	418
附录 A C++关键字	421
附录 B AscII 字符对照表	423

第 1 章

C 语言的基本概念

1-1 程序语言

自从有计算机以来，程序语言就一直存在着，它肩负着人类与计算机沟通的桥梁，默默耕耘了大半个世纪。许多人赞叹着计算机的无所不能，确不知背后有程序语言这个大功臣，协助程序设计师实现炫丽的美梦，也成就了无数科技的神奇创造。

以往，似乎学习计算机的人，都得懂一点程序语言，才能把计算机搞定。现在，程序语言退居幕后，你不需要学习任何计算机程序语言，就可以把计算机操作的得心应手，甚至堪称专家。说的更清楚些，这种专家可以称是计算机应用专家，在公司的任何部门都吃的开；但是，如果身处信息公司或信息相关部门，或者负责公司某项信息项目，想在老板的威严下游刃有余，可得下点功夫。其中，程序语言就是必修的课程。

现在的高级语言颇多，语法和概念大同小异，只要学习过其中一种语言，想要学习另一套语言时，可有事半功倍的效果。本书将介绍的 C++ 语言，是近代计算机语言较受重视及一直沿用的程序语言，属于中高级语言，但也可以做低级的控制，想要写系统的人士，应该好好学习它喔！

1-1-1 为什么需要程序语言

观察计算机的演进史可以发现每一阶段的硬件发展，其目的都是在改进人类控制计算机的方法，以及执行的效率。近代美国普林斯敦大学的数学家范纽曼（von Neumann）提出一种新的方法，将控制计算机的程序指令存储在计算机内存中，当计算机要执行时，可从内存中读取指令，执行指令，甚至可以修改程序，使程序设计的方法往前迈进一大步，这就是著名的“存储程序概念”（stored-program concept）。这个概念对后代的计算机发展影响很大，使计算机不再只是实验室内的工具，变成每个人都可以应用在日常生活中的必需品。

由于这个程序设计的概念，计算机制造厂商也开始进行系统软件的设计，让硬件在操作系统的指挥下，得以有效地运作。近代最流行的操作系统，UNIX 最为普及，其稳定性最被赞赏，而且有很多免费的资源可利用，所以从最小的工作站到可以容纳数百位人员同时使用的超级计算机，都有它的足迹。另外，在个人计算机方面，微软（Microsoft）的 MS-DOS 和视窗 WINIDOWS 操作系统最为普遍，其所设计的图形使用界面，让一般人可以很容易地操



作及设计计算机软件。

说了这么多，到底什么是程序语言？简单地说，程序语言是包含一组指令及一套语法规则，使用程序语言所规定的指令，并依照其语法规则组合之，经过翻译系统将此语言翻译成计算机可以辨认的符号，这种语言称为程序语言。而程序设计师的目的，就是将想法转成该程序语言的语法规则，产生计算机程序，便可以指挥计算机做事了。就如同学习一般生活上的语言，除了有单字外，还得有文法来组合单字，才能让人听得懂、看得懂。

为什么要写程序？其实，计算机终究不是个活的机器，要让计算机能帮我们处理资料，变成有用的信息，得靠程序设计者的巧思及创意，把计算机变聪明，甚至变成活的机器。例如：全班学生的成绩登记后，如果只是看看每一个学生的成绩几分，这只能说是一堆资料（data）而已，似乎没有什么特别意义。但是，如果能将成绩予以排序，或者计算出平均分，以便和其它同学比较，看看自己在全班的排名，看看自己哪一科成绩在班上是比较优秀的，才是有意义的信息（information）。

那么，我们该选择哪一种程序语言？

在计算机刚发明不久时，程序设计师编写程序是用人工方法，将一个个的指令或程序刻录在计算机芯片中，相当耗时费力。后来演进成使用代码输入计算机，称为低级语言，例如，机械语言和汇编语言，这种方法虽然可以省些时间和精力，但是人的记忆力是有限的，低级语言比较不容易学习，许多程序的细节都得考虑得清清楚楚，一个个的设计在程序中；许多复杂的程序，需要耗费大量人工输入程序代码，且不易侦错及维护，于是高级语言逐渐被开发出来，在计算机软件设计上帮了不少忙。

1-1-2 高级语言

高级语言的设计理念，是要模拟人类的语言及逻辑思考的方式，让程序设计者可以通过自己熟悉的语言来设计程序，提高程序的可读性，以便日后侦错及维护；同时，高级语言所设计的程序有较高的可移植性。在不同的计算机操作系统下，只要有该计算机系统版本的编译器，便可以编译及执行这个程序。因为一般低级计算机程序语言所设计的程序，其程序指令大部分针对特定系列的微处理器（microprocessor）而设计，故只能在特定的计算机执行，不能转移到其它机种的计算机编译与执行。例如，Intel 80x86 的汇编程序，不能在 Macintosh 680x0 计算机上执行。

在高级语言的演进中，较著名的程序语言列举如表 1-1 所示：

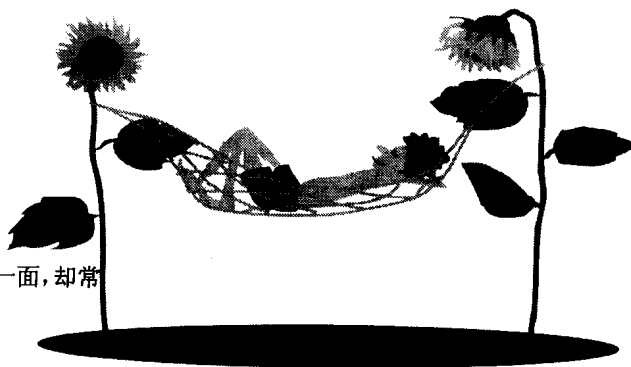
表 1-1

程序语言名称	主 要 用 途
FORTRAN	应用在科学运算、数学、理工等方面
COBOL	应用在数据管理及商业方面
PASCAL	应用在科学计算及教学方面
LISP	人工智能语言
Modula-2、Modula-3	应用在科学计算及一般用途
C、C++	应用在工程及科学方面
BASIC	应用在一般用途及教学方面
JAVA	应用于网络及通讯方面

虽然高级语言比较容易学习及设计，但却没有低级语言执行效率高，因为低级语言可以直接控制计算机基本的运算，而高级语言所提供的函数库，主要是提供一般性的使用，并没有针对某项特别任务而设计，没有做最佳化的考虑，也就无法满足要求效率的程序设计；就如同我们所穿的衣服，大量制造的成衣穿起来就没有量身订制的衣服来得舒服。

因此，设计软件时，要选择何种语言来设计，是在系统规划时一项重要的抉择。本书所介绍的程序语言是 C++ 语言，它是近代程序语言的明星，加上它有面向对象设计的概念，可以很方便设计出面向对象程序。至于什么是面向对象？在本书第 7 章将有详细的介绍。

我们经常只看到别人快乐的一面，却常忽略他们的努力与付出。



1-2 C 语言的演进

C 语言是 1972 年 Dennis Ritchie 和 Ken Thompson 两人一起设计 UNIX 操作系统时发展出来的，由于语法简单，可以作高级应用，也可低级控制计算机基本功能，功能强大，成为最近几年来程序设计师的最爱。有关 C 语言的版本很多，例如：Turbo C、Borland C、ANSI C 或 Microsoft C 等，除了函数库有所差异外，基本的语法都是差不多的。

比 C 语言还多两个加号的 C++，是 AT&T 贝尔实验室的 Bjarne Stroustrup 博士设计的，他在 1980 年将面向对象的概念加到 C 语言中，产生了“C with Classes”程序语言。1985 年，Stroustrup 博士在这个基础上，加入了更多面向对象的特性，包括运算符重载（operator overloading）、虚函数（virtual function）等，使 C 语言融合了面向对象及传统的面向过程的程序设计机制，应用层面更广，而完成 C++ 语言的设计。于是，各家软件厂商纷纷设计出不同版本的 C++ 程序语言，如：USL C++(AT&T)、HP C++、Borland C++、Microsoft C++、Visual C++ 等。本书将使用 Visual C++ 作为范例设计的语言版本。

1-3 Microsoft Visual C++ 6.0 的安装

微软公司出版的 Visual C++ 6.0 版是 Visual Studio 6.0 版的成员之一。Visual Studio 6.0 是一套用来开发软件的工具，内容包含五套各有特色的程序语言（Visual Basic、Visual C++、Visual FoxPro、Visual InterDev 和 Visual J++），以及两套辅助的工具（Visual SourceSafe 和 MSDN Library）。因此，安装 Visual C++ 必须使用 Visual Studio 的第一张光盘来安装。

另外, Visual Studio 6.0 版有两种版本, 一个是 Professional 专业版, 为计算机专业人员提供了一套功能完整的工具, 以便开发解决方案, 适合一般窗口系统或相关软件包的开发者使用; 另一个版本是 Enterprise 企业版, 提供开发大型应用软件, 需要管理重要资料的系统, 并提供许多特别的开发与管理工具, 适合企业界开发软件使用。

不论您现在手上的 Visual Studio 版本为何, 只需要第一张光盘即可安装 Visual C++, 且本书所介绍及使用到的功能, 两种版本都会提供。

以下将以企业版为例, 安装的步骤如下:

步骤 1: 将 Visual Studio 6.0 版的第一张光盘放入光驱中, 系统会自动执行, 出现如图 1-1 所示的窗口, 直接选择 **Next** 钮。

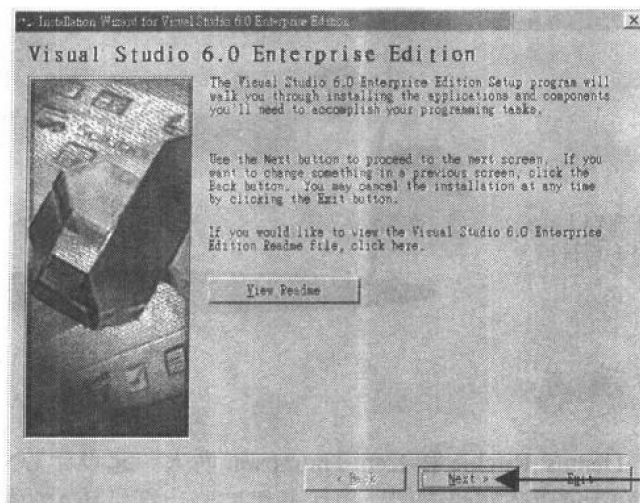


图 1-1

步骤 2: 出现 End User License Agreement 窗口后, 选择 I accept the agreement, 再选择 **Next** 钮 (见图 1-2)。

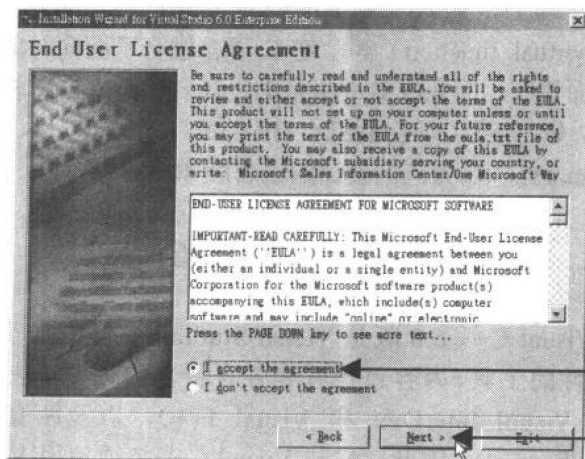


图 1-2

步骤 3: 出现 Product Number and User ID 后, 输入软件识别码 (如果是 MSDN 中所附带的 Visual Studio 6.0, 则无须输入), 接着输入您的名字及公司名称, 最后选择 **Next** 钮 (见图 1-3)。

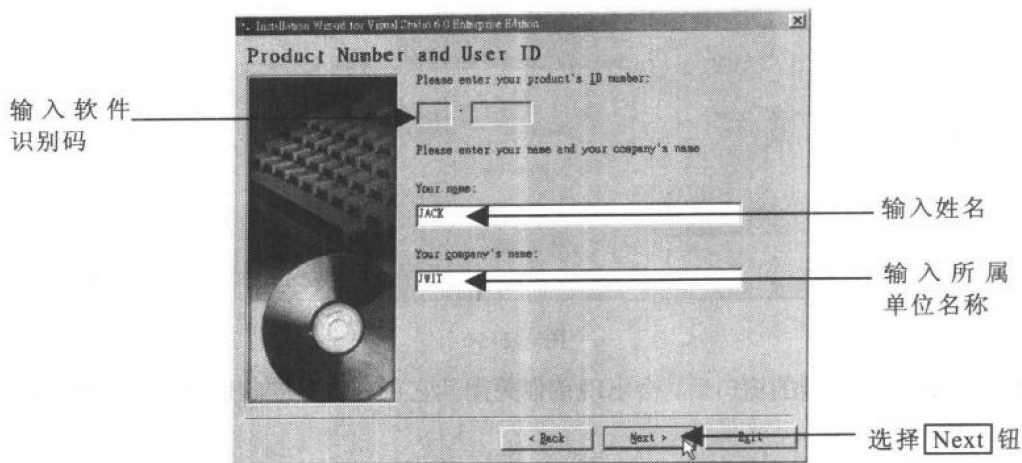


图 1-3

步骤 4: 出现 Visual Studio 6.0 Enterprise Edition 窗口后, 因为我们只要安装其中的 Visual C++ 而已, 所以选择 Enterprise Setup Option 栏中的 Custom 选项, 再选择 **Next** 钮。见图 1-4。

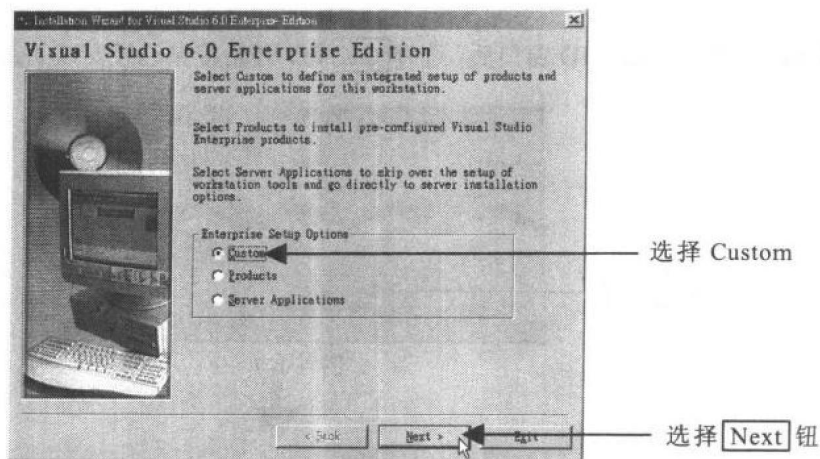
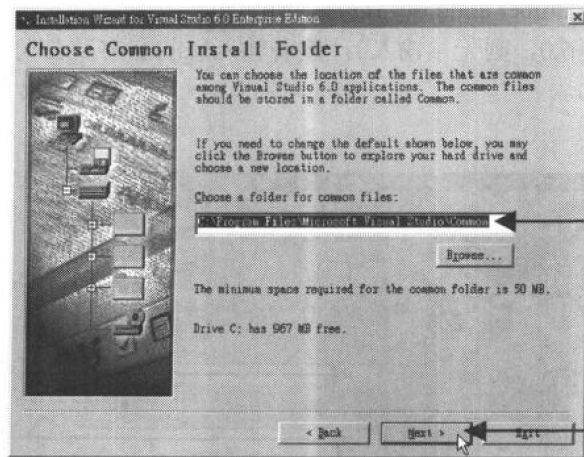


图 1-4

步骤 5: 出现 Choose Common Install Folder 窗口后, 输入 Visual Studio 常用的文件安装路径, 或选择 **Browse...** 钮找寻路径 (通常安装程序会自动显示预设的路径, 不需要修改), 接着选择 **Next** 钮。见图 1-5。



输入常用文件
安装路径

选择 **Next** 钮

图 1-5

步骤 6: 出现开始安装的窗口后, 会出现请你关闭其它正在执行的软件窗口, 选择 **Continue** 钮。见图 1-6。

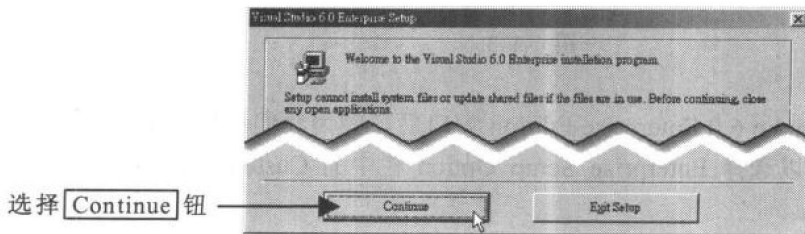


图 1-6

步骤 7: 出现 Product ID 窗口后, 选择 **OK** 钮, 开始搜寻已经安装的组件。见图 1-7、图 1-8。

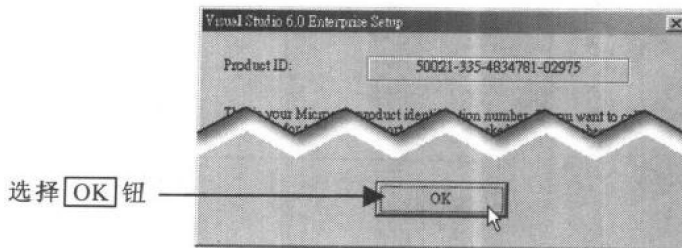


图 1-7

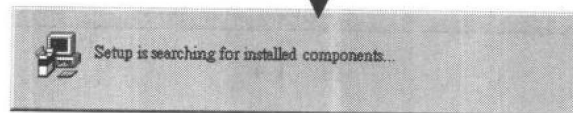
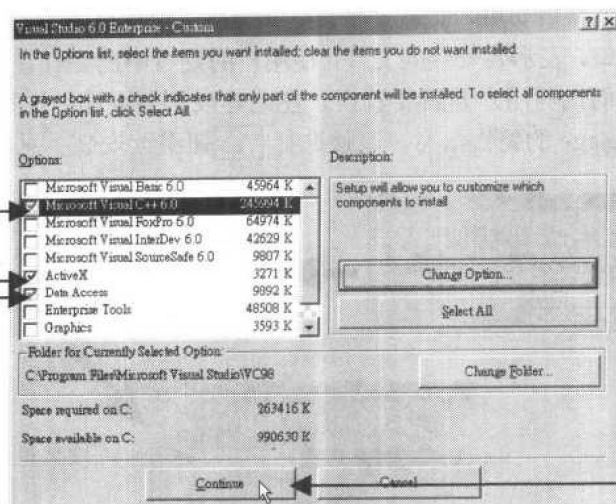


图 1-8

步骤 8: 出现 Visual Studio 6.0 Enterprise-Custom 用户自定义安装的窗口, 由于只需要安装 Visual C++ 6.0, 因此将其它项目前面的 ☐ 删除, 只留下 Microsoft Visual C++6.0、Active X、Data Access 即可, 然后选择 **Continue** 钮。见图 1-9。

勾选 Microsoft
Visual C++ 6.0

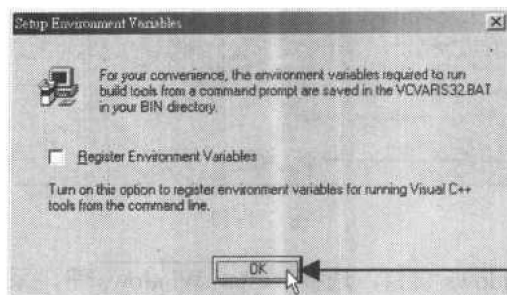
其它选项保
持不变



选择 **Continue**
钮

图 1-9

步骤 9: 出现 Setup Environment Variables 窗口后, 安装程序会询问您是否需要存储环境变量设置的批处理文件 VCVARS32.BAT, 以便将来在 DOS 命令行的环境下执行此批处理文件。由于在 DOS 环境下执行的机会不大, 因此不必勾选, 直接选择 **OK** 钮即可。见图 1-10。



选择 **OK** 钮

图 1-10

步骤 10: 出现 Visual Studio 6.0 Enterprise - Restart Windows 窗口后, 选择 **Restart Windows** 钮, 系统会自动重新启动 Windows。见图 1-11。

选择 **Restart Windows** 钮

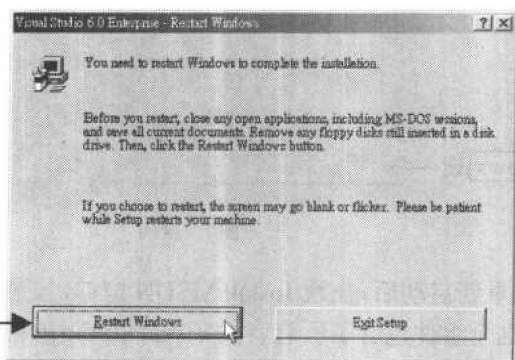


图 1-11