

C 和 C++语言程序设计教程

海搏创作室 编著

 **浦东电子出版社**
PeP Pudong ePress

前 言

《C 和 C++ 语言程序设计教程》熔 C 和 C++ 为一体、以基础知识和实例为主、以通俗的实例去讲解抽象的 C、C++ 及它们中的重点和疑难问题，让读者能在较短的时间内，能够深入全面掌握 C 和 C++ 语言的精髓。

创作室在创作时没有刻意去区别 C 和 C++ 语言，而是力求在讲解 C++ 的同时阐述 C 语言，使读者从一开始就把它们作为同一种语言学习。另一较具特色的是从头至尾均使用 Visual C++ 作为编程环境，使读者能够在学习 C 和 C++ 语言基础知识的同时能够掌握 Visual C++ 的编程方法，以便读者今后深入学习 C++ 和其他可视化语言。

《C 和 C++ 语言程序设计教程》以实例讲述概念和语法，提供了大量的实例、思考和练习题（包含答案），重点阐述了 C 和 C++ 语言的基本语法现象，包含以下主要内容：

1. C 和 C++ 语言的基础知识：在这一部分讲解了基本数据类型、表达式语句的定义和使用方法，介绍了数组和指针的作用和意义及应用，阐述了数据结构、枚举和联合的定义与作用，以实例说明了它们在 C 及 C++ 中的应用。

2. 类与对象：重点讲述了类的基本概念、定义和使用，介绍了类的构造函数和析构函数的作用，还介绍了类的继承、封装、重载和派生，讲述了类与对象的区别与联系，提供了大量的实例帮助读者理解语言中的重点和难点。

3. 面向对象的编程技术：这一部分介绍了面向对象的基本概念、作用和意义，以实例方式讲解了面向对象的编程技术。重点介绍了微

软公司的 MFC，阐述了 Windows 的消息、消息循环、消息映射及消息处理，使读者把 C++ 与面向对象的编程技术进行有机地结合。

实例中提供的程序短小实用，可作为读者今后编程的资源，如实例：矩阵旋转、数据的二进制与十六进制的转换、字符串的处理和文件流输入输出等。

C 和 C++ 语言相对 Basic 及 FoxBASE 等语言是一种比较难学的语言，创作室根据自身经验，向读者建议在学习 C 和 C++ 语言时，可采用下面的方法和步骤：

简单阅读你要学习的某一章或节；

上机调试实例，可以直接打开光盘中的实例文件(*.cpp) 进行练习；

重新阅读你学习的章或节，体会和掌握书中的难点和重点。

创作室相信：通过我们创作室和读者的共同努力，一定会使读者以最快的速度 and 效率掌握 C 和 C++ 语言。

多媒体交互式光盘分为四部分 (1) 等级考试 (包括选择题和填空题)；(2) 问答题：精选 C 和 C++ 语言的重难点问题并一一解答；(3) Examples：提供所有实例的源程序，读者可直接调用调试 (4) 出版社简介。光盘中所有问题都为交互式的，并配备了答案，以利于读者巩固所学的知识点。

读者的成功是我们的追求！

海搏创作室

内 容 提 要

《C 和 C++ 语言程序设计教程》流畅地把 C 与 C++ 溶为一体，详细阐述了 C 和 C++ 语言的基本数据类型、结构和语句；讲述了类及类的继承、封装和重载；重点介绍了 C 和 C++ 中的一些重点和疑难点，如多维数组、指针和流等；特别阐述了 C++ 语言一些新的语法现象，如 new 和 delete 的使用。

《C 和 C++ 语言程序设计教程》以通俗易懂的语言，向读者展示了 C 与 C++ 语言的方方面面，同时提供了大量的实例、思考和练习（附有答案），便于读者巩固学习效果；另一特点是使用 Visual C++ 作为编程环境，使读者从开始学习语言就接触可视化编程环境，为今后学习 Visual C++ 等语言打下了良好的基础。

《C 和 C++ 语言程序设计教程》是众多作者教学和编程经验的结晶，文字流畅、深入浅出、实例丰富，可以作为普通高校本、专科学生的教学用书，也可供计算机爱好者自学使用。

多媒体交互式光盘分为四部分（1）等级考试（包括选择题和填空题）（2）问答题：精选 C 和 C++ 语言的重难点问题并一一解答（3）Examples：提供所有实例的源程序，读者可直接调用调试（4）出版社简介。光盘中所有问题都为交互式的，并配备了答案，以利于读者巩固所学的知识点。

书 名：C 和 C++ 语言程序设计教程

文本著作者：海搏创作室

C D 制 作 者：海搏多媒体制作中心

责 任 编 辑：舒红梅

出版、发行者：浦东电子出版社

地 址：上海浦东郭守敬路 498 号上海浦东软件园内 201203

电话：021-38954510，38953321，38953323（发行部）

经 销：各地新华书店、软件连锁店

排 版：四川中外科技文化交流中心排版制作中心

C D 生 产 者：东方光盘制造有限公司

文本印刷者：成都地图出版社印刷厂

开本 / 规格：787×1092 毫米 16 开本 16.5 印张 280 千字

版次 / 印次：2002 年 5 月第一版 2002 年 5 月第一次印刷

印 数：0001—8000 册

本 版 号：ISBN 7—900346—53—8

定 价：29.80 元（1CD）

技术支持热线（028）85412516

说明：凡我社光盘配套图书有缺页、倒页、脱页、自然破损，本社发行部负责调换。

目 录

第 1 章 C++和 VC 开发环境的简介	1
1.1 C 语言的背景与特征	1
1.2 C++的产生背景和特征	2
1.3 VC 开发环境的简介	3
1.3.1 Visual C++6.0 概述	3
1.3.2 Visual C++开发环境	3
第 2 章 基础知识	12
2.1 标识符	12
2.2 常量	13
2.3 变量	15
2.4 基本数据类型	16
2.5 运算符与表达式	22
2.5.1 算术运算符和算术表达式	22
2.5.2 赋值运算符和赋值表达式	24
2.5.3 关系运算符和关系表达式	26
2.5.4 逻辑运算符和逻辑表达式	27
2.5.5 逗号运算符和逗号运算表达式	28
2.5.6 位运算符	29
2.5.7 大致归纳各类运算符的优先级	30
2.6 C++中的语句	30
2.6.1 C++语句的基本类型	30
2.6.2 控制语句在 C++中的用法	31
2.6.3 循环的嵌套	39
2.7 函数	40
2.7.1 函数的定义与声明	40
2.7.2 函数的调用	40
2.7.3 函数的返回值	41
2.7.4 函数的参数	41
2.7.5 作用域	42
2.8 综合实例	42
练习题	46
答 案	47
第 3 章 数组和指针	49
3.1 一维数组的定义和引用	49
3.1.1 一维数组的定义	49
3.1.2 一维数组元素的引用	50
3.1.3 一维数组的初始化	50
3.1.4 一维数组程序举例	51
3.2 多维数组的定义和使用	53
3.2.1 二维数组的定义	53
3.2.2 二维数组的引用	54
3.2.3 二维数组的初始化	54

3.2.4	二维数组程序举例	55
3.3	字符数组	58
3.3.1	字符数组的定义	58
3.3.2	字符数组的初始化及引用	58
3.3.3	字符串的输入和输出	60
3.3.4	字符数组应用举例	62
3.4	指针的概念	64
3.4.1	指针用法概论	65
3.4.2	指针在内存中的存取	66
3.5	变量的指针和指向变量的指针变量	67
3.5.1	指针变量的定义	67
3.5.2	指针变量的引用	68
3.6	数组的指针和指向数组的指针变量	70
3.6.1	数组的指针变量的定义	70
3.6.2	利用指向数组元素的指针变量引用数组元素	71
3.6.3	指向多维数组的指针和指针变量	74
3.7	字符串的指针和指向字符串的指针变量	76
3.7.1	字符串及其指针变量	76
3.7.2	字符串处理函数	78
3.8	指针数组和指向指针的指针	80
3.8.1	指针数组的概念及其引用	80
3.8.2	指向指针的指针	82
3.8.3	指针在 C++ 中的特殊用法	83
3.9	对象数组	87
3.10	C/C++ 中的动态分配和释放	88
3.10.1	C 语言中的动态分配函数	88
3.10.2	C++ 中 new 和 delete 概述	88
3.10.3	new 和 delete 的使用	89
	练习题	92
	答案	93
第 4 章	结构、联合、枚举	95
4.1	结构类型	95
4.1.1	结构类型变量的定义、初始化及其引用	95
4.1.2	指向结构类型数据的指针	98
4.1.3	C++ 中的结构与 C 语言中的差异	100
4.2	联合类型	102
4.2.1	联合类型的定义	102
4.2.2	联合类型变量的引用及其特点	103
4.2.3	C++ 中的联合类型	105
4.3	枚举类型	106
4.3.1	枚举类型的定义	106
4.3.2	C++ 中的枚举类型	107
	练习题	107
	答案	108
第 5 章	类	109
5.1	类与对象	109

5.2	构造函数和析构函数	114
5.2.1	无参数的构造函数和析构函数	115
5.2.2	参数化的构造函数	116
5.2.3	重载构造函数和拷贝构造函数	117
5.3	静态成员	119
5.4	this 指针	122
5.5	友元	123
5.6	类型常量	125
5.7	类对象作为成员	126
5.8	对象的数组	127
5.9	对象指针	128
5.10	传递对象到函数	129
5.11	类属单向链表的例子	131
	练习题	134
	答 案	136
第 6 章	函数及重载	140
6.1	函数的定义与说明	140
6.2	函数参数和函数的值	142
6.2.1	函数的形式参数和实际参数	142
6.2.2	函数的值	143
6.3	函数的调用	145
6.3.1	函数调用的概念	145
6.3.2	函数调用的形式	146
6.4	函数参数的传递	147
6.4.1	按值传递	147
6.4.2	按地址传递	148
6.5	函数参数的种类	149
6.5.1	引用类型参数	149
6.5.2	指针变量作为函数参数	151
6.5.3	数组作为函数参数	153
6.5.4	字符串指针作为函数参数	155
6.5.5	函数的指针和指向函数的指针变量	157
6.6	局部变量和全局变量	160
6.6.1	局部变量	160
6.6.2	全局变量	161
6.7	递归函数	162
6.8	内联函数	163
6.9	重载	164
6.9.1	函数重载	164
6.9.2	运算符的重载	167
6.9.3	new 和 delete 的重载	171
	练习题	173
	答 案	174
第 7 章	继承性和派生类	176
7.1	继承及派生概述	176
7.2	派生类设计	177

7.3	单继承	180
7.3.1	单继承概述	180
7.3.2	继承中的构造函数	185
7.3.3	继承和组合	188
7.3.4	继承中的几种特殊用法	192
7.3.5	继承的一个实例	198
7.4	多继承	203
7.4.1	多继承概述	203
7.4.2	多继承定义	205
7.4.3	虚基类	207
7.4.4	虚函数	215
	练习题	222
	答案	224
第 8 章	面向对象编程技术在 VC 中的应用	226
8.1	MFC 类库简介	226
8.2	MFC 应用程序框架	228
8.3	消息循环、消息映射及消息处理	229
8.3.1	事件及消息概述	229
8.3.2	MFC 消息循环	229
8.3.3	消息映象	230
8.3.4	消息处理函数	231
8.4	实例及解说	231
8.4.1	使用 MFC AppWizard 创建一个新项目	231
8.4.2	实现画一个椭圆的效果	232
第 9 章	C++ 的 I/O 流库	237
9.1	C++ 流库的结构	237
9.1.1	streambuf 类及其派生类	238
9.1.2	ios 类及其派生类	238
9.2	输入和输出	239
9.2.1	istream	239
9.2.2	ostream	241
9.3	输入运算符和输出运算符	241
9.4	文件 I/O	244
9.4.1	文件的打开和关闭	244
9.4.2	文件的读写	246
9.5	格式控制	247
9.5.1	用 ios 类成员函数进行格式控制	247
9.5.2	用操纵函数控制格式	249
	练习题	251
	答案	252

第 1 章 C++和 VC 开发环境的简介

本章重点及主要内容：

C 语言的特点及发展过程

C++语言的优势及产生背景

使 Visual C++创建 C++应用程序的方法

本章首先介绍了 C 和 C++语言的背景和特征。在概述时，虽将 C 和 C++分开以示重点；但在讲述时，尽量将两者密切联系起来，避免单纯的讲述，竭力使读者对 C 和 C++有全局的和新的认识。在解说 Visual C++的开发环境时，紧紧结合实例操作（编译、链接、运行、调试），以达到让读者轻松上手的目的。

1.1 C 语言的背景与特征

C 语言是当前流行语言 C++的基础，在其中具有十分重要的地位。之前，C 语言的应用也风行一时；它兼顾了高级语言和低级语言方面的特性，使用户既可编写系统软件，又可编写应用软件；加之 C 语言自身的诸多优点，使它好评如潮。

C 语言是 1970 年由 Dennis Ritchie 设计，它是在 DEC PDP-11 上的 Unix 系统环境下实现的。C 语言是在 B（全称：basic combined programming language）语言的基础上发展起来的，后来 C 语言经过多次的改进，直到以 1978 年发表的 Unix 第七版中的 C 编译程序为基础，Brian W.kernighan 和 Dennis M.Ritchie（合称 K&R）合著了影响深远的名著《The C programming Language》，这本书便成了后来广泛使用的 C 语言的基础，被称为标准 C。在 1983 年，美国国家标准协会(ANSI)又制定了新的标准：ANSI C；在 1990 年，国际标准化组织 ISO 接受了 87ANSI C 的标准为 ISO C。在新的标准下，C 语言为了尽量和 C++中的规定相符合，和传统的版本有不少的差别。例如：ANSI C 中，初始化更一般化了，现在可以对自动数组和结构直接初始化；在传统的版本里，必须是静态数组才能直接进行初始化。

C 语言相比于其它语言具有突出的优点：

1. 编写自由、简洁、灵活，变通性强：例如：++、--、+=、if(判别式)语句和指针的广泛应用等，在 C 语言中，这一特征随处可见。建议初学者在编写风格上，尽量多使用 C 语言自身的编写风格，使程序在可读性和运行效率的优势突出体现出来；

2. 运算符和数据类型丰富：除了具有其他语言的通用运算符和数据类型，增添了不少种类，能用来实现各种复杂的数据处理；

3. C 语言是理想的结构化语言：它具有灵活的结构化的控制语句（if ...else 语句、while 语句、do...while 语句、switch 语句、for 语句）和结构化的模块；

4. C 语言既可编写系统软件，又可编写一般应用软件：兼有高级语言和低级语言的特性；

5. C 语言的一个重要的特性便是语法限制不太严格，程序设计的自由度大：然而，在



C++ 中对语法检查更规范,更严密些。在 C 语言的编译环境下的可以顺利通过的程序,而在 C++ 的编译系统中可能会错误百出。当程序代码很长时,往往虽然通过编译后,但程序的执行结果仍然不可预见,因为尚有很多潜在的、隐藏的错误没有被编译系统检查出来,从而使程序调试的难度增大。这正是“限制”和“灵活”的矛盾。但总的来说,C 语言使程序编写者有了较大的自由度,使编写更加的灵活,这是它的一个显著的特点。

1.2 C++ 的产生背景和特征

程序设计方法经历了过程化程序设计(procedural programming)、结构化程序设计(structured programming)、面向对象程序设计(object-oriented programming)几个阶段。

过程化程序设计,是在一个数据集合上进行的一系列操作;而结构化程序设计提供了系统的组织这些过程和能够处理大量数据的有效手段。C 语言便是理想的结构化语言,当它在处理较小规模的程序时,较得心应手,但是当程序的规模较大时,结构化程序设计方法就显出它的不足,数据和处理这些数据过程的分离使程序变得模糊难解,C 程序的编写者必须细致地分析程序中的每一个细节和运行中数据每时每刻的变化情况,就其中的众多变量的值在不同过程中变化的情况,也就纷繁复杂,往往使人感到力不从心。于是新的软件危机出现了,为了解决这一软件设计危机,在 1980 年贝尔实验室的 Bjarne Stroustrup 当时为了仿真课题的研究,在 C 语言的基础上编写了称为“带类的 C”语言版本,提出了面向对象的程序设计(Object -Oriented Programming ,简称 OOP),经过多年修改和扩充,便形成了现在的 C++ 语言。

C++ 语言的特征体现在 C++ 程序设计语言在 C 语言的“增强”方面:

1. C 语言的扩充

例如:C++ 的输入和输出(cout 与 cin) 函数的重载(同名函数的可重复使用,实现一名多义)、变量的引用类型(type &a)、内置函数(关键字 inline)、动态内存分配、撤销(new , delete)等。

2. 面向对象的机制

面向对象的程序设计的实质是把数据和处理这些数据的过程或行为合并为一个单独的“对象”,一个具有明确特性的完整实体。在理解面向对象的程序设计方法时,应从一种系统方法论的角度来理解,它如同世界上万事万物一样,既有联系又有独立性、完整性。面向对象程序设计语言的特征表现在:

(1) 封装和数据隐藏

C++ 通过增加一种新的数据类型“类”定义对象来支持封装和数据隐藏。类作为一种类型,描述了众多对象共同的特性(由数据和行为组成)。类作为一个封装实体,可以完整使用,其内部的成员可以被隐藏起来。

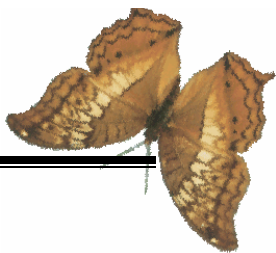
(2) 继承与重用

C++ 通过继承来支持重用并产生新的子类,称为派生类型。

(3) 多态性

多态性是另一个与类层次有关的重要概念,即公共消息可以发送给父类对象和所有的子类对象。多态性允许每个子对象以适合自己的形式来响应消息,来支持“不同类型有各自的响应”的思想,以达到提高已有代码的可扩充性的目的。





1.3 VC 开发环境的简介

1.3.1 Visual C++6.0 概述

Windows 应用程序开发系统 Visual C++6.0 是 Visual Studio98(Visual Studio 6.0)家族的成员之一,起源于 Microsoft C/C++,是当前 Windows 应用开发系统中与 Windows 平台本身最具有亲缘关系的开发平台。特别是 Visual C++6.0 支持 Windows 95、NT4.0 以及 Windows98 的 Win32 SDK(Software Development Kit - 软件开发包),能非常方便的使用其中提供的 API 来进行 Windows 应用程序的开发。Visual C++6.0 同 Visual C++5.0 一样彻底摒弃了对 16 位操作系统的支持,作为 32 位应用程序的开发工具。

Visual C++主要的特点是支持可视化编程和支持面向对象的编程技术,它提供了大量的可视化编程的工具 如 App Wizard 和 Class Wizard 等。此外,可使用 Activex 技术,支持 Internet 编程;支持 ODBC 数据库编程等。

本书的编写侧重于对 C 和 C++的基础的讲解,由于考虑到当前流行的 Visual C++开发系统而采用了 VC 的编译系统,其目的在于向读者指引一个方向,使读者在以后对 VC 的学习中能够轻松上手。

1.3.2 Visual C++开发环境

安装 Visual C++6.0 硬件配置的最低要求:486CPU、100MB 的硬盘空间、16MB 内存 (Windows95 下可为 8MB);软件配置要求:32 位操作系统 Win95,Windows NT 3.5,Win98,Windows NT 4.0 及更高版本)。

如果安装成功 Visual C++6.0 后,用鼠标点击 开始 ” “ 程序 ” “ Microsoft Visual Studio 6.0 ” “ Microsoft Visual C++6.0 ”。随后,便是 Visual C++6.0 的集成环境界面,如图 1.1 所示。

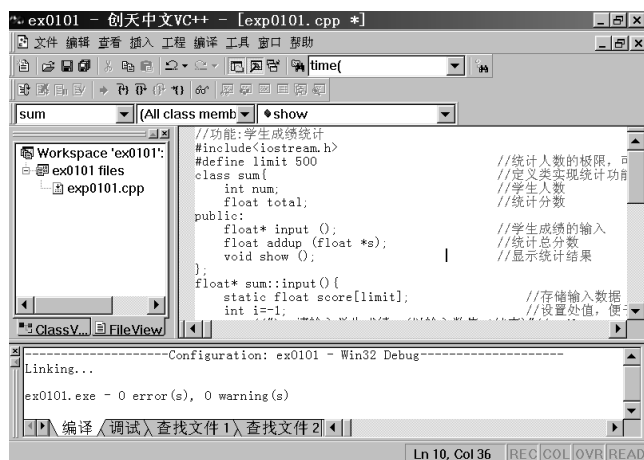


图 1.1 VC++6.0 集成环境界面

在集成环境中,最重要的是工程(project)的概念。工程是相关源文件的集合,包括源程序、头文件及资源定义文件,是编程的主要场地。Visual C++平台是自动化很高的编译系

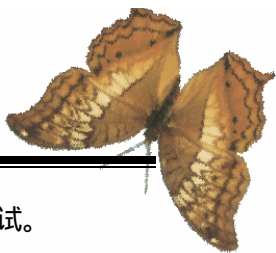


统，它能自动处理源文件之间的关系，利用其内在的推理规则来激活编译器、连接器和资源编译器，最后生成可执行文件。

下面，通过一个例子演示一般工程的创建、编译及调试过程，并结合相关要点予以说明。为了便于调试和 C++ 的特征相契合，在举例时便应用了类的设计（注：假若读者是第一次接触 C++ 的内容，则大可不必在意程序本身，而应学会其过程中所使用的程序编写、编译和调试的方法，这也是创作室的最大的心愿）。在此，编写了一个实现学生成绩统计的简单程序。

实例 Exp0101: 实现学生成绩统计功能，源程序文件名为 exp0101.cpp。

```
#include<iostream.h>
#define limit 500 //统计人数的极限，可修改
class sum{ //定义类实现统计功能
    int num; //学生人数
    float total; //统计分数
public:
    float* input (); //学生成绩的输入
    float addup (float *s); //统计总分数
    void show (); //显示统计结果
};
float* sum::input(){
    static float score[limit]; //存储输入数据
    int i=-1; //设置初值，便于数组下标运用与计数
    cout<<"\n <<请输入学生成绩：(以输入数值-1 结束)"<<endl;
    do {
        i++;
        cin>>score[i];
    }while (score[i]!=-1); //以-1 作输入结束标志
    num=i;
    return score; //返回数组存储首地址
}
float sum::addup(float* s){
    total=0; //赋初值
    while (*s!=-1)
        total+=*s++; //统计分数
    return total; //返回统计分数
}
void sum::show(){
    cout<<"学生统计人数："<<num<<endl;
    cout<<"学生成绩统计分数："<<total<<endl;
}
void main(){
    float *p; //定义指针传递地址
    sum student ; //定义对象
    p=student.input(); //将数据存储区首地址传递给 P
    student.addup(p); //统计学生成绩
    student.show(); //显示统计结果
}
```



下面分三步，讲述一个 C++工程的创建、编译与连接，最后便是程序的调试。

1. 第一步：工程的创建与源文件的编辑

(1) 打开 VC 的集成环境如图 1.1 所示。

(2) 单击“文件”中“新建”项，打开“新建”对话框，在其中选择“工程”项，便可新建一个工程，如图 1.2 所示。（注：在图中，可看见 VC 支持多种应用程序的开发，为了便于对 C++语言的理解，便于使用命令行操作和类似于 DOS 风格的文本显示的应用程序，称为控制台应用程序（Console Application）。在这种应用程序下，可以使用大部分 DOS 环境下的库函数，而关键在于使用这种模式便于学习新的语言，更容易理解和操作程序的执行过程。）

(3) 按图 1.2 的提示，输入新建工程的名字和存储工程的路径。（如图 1.2：在工程下输入 exp0101；在位置下输入 D:\exp）

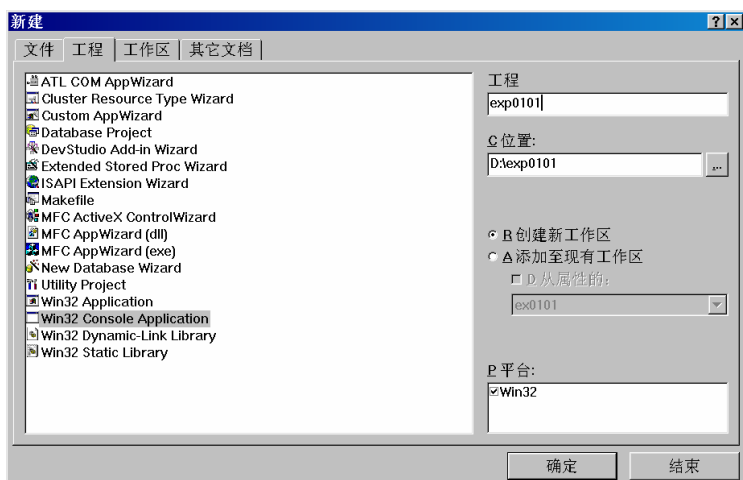


图 1.2 File 菜单的 new 对话框

(4) 然后出现两个对话框按回车键确定，随后工程工作区便有了“Class View”和“File View”两项，用来浏览工程的文件和类，当然其内容目前还是空的。

(5) 由于对程序进行调试，所以要将输出文件定为 Debug 版。打开“编辑”菜单，选择“配置”命令，在对话框中选择相应工程的 Win 32 Debug 项。但一般默认设置都有 Debug 版和 Release 版，假若要发布应用程序则可将输出文件设置指定为 Release 版。其区别在于：后者比前者少几个文件，更简短，运行更快，但前者多了一些调试的信息，便于调试程序。

(6) 选择“文件”菜单中的“新建”项，再选中“文件”页，这里有多种文件的编辑。选中 C++Source File。注意右边的“添加工程”下拉列举框中选中相应的工程，将此文件加入到该工程里去。如图 1.3 所示。

(7) 按“确定”后，便可以在相应的工程空间的编辑区中编辑源文件。将例程 exp0101.cpp 输入编辑区，如图 1.4 所示。

(8) 至此，便可以非常方便的编写程序代码，完成了工程的创建过程。希望读者能够从创建工程一直到能编辑源代码的整个过程明白 VC 开发系统整体思路和全局意义。想一想与其他的开发系统的区别在哪里？当你在开发很大的项目时，那时你就能更好的体会到 VC 开



发系统的方便和功能的强大。

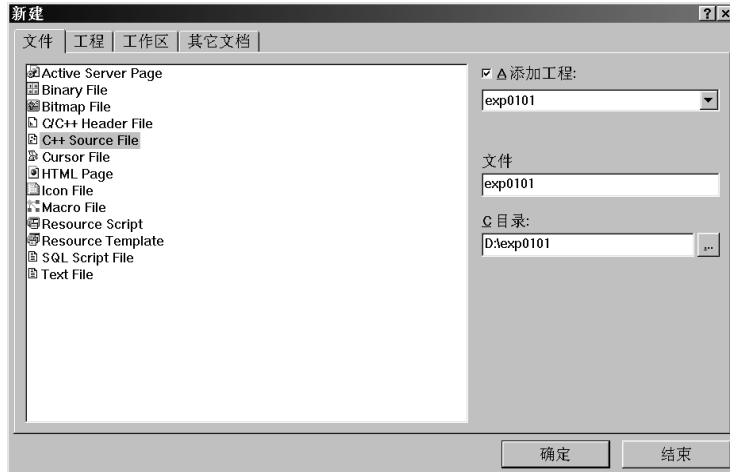


图 1.3 “新建”对话框创建源文件

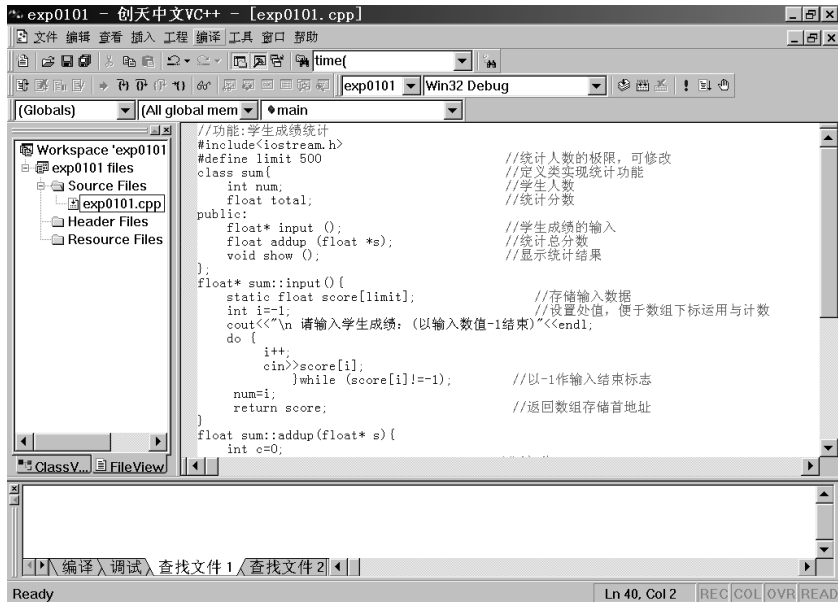


图 1.4 在工程编辑区编辑源文件

相关说明：

1. 假若已建立有某个工程，需要打开时可以直接在“文件”中单击“打开工作区”、“最近的文件”或“最近的工作区”来打开自己已经存在的或最近访问的文件或工程。
2. 工程包括了用户在开发、编译、连接和调试应用程序时所需的所有文件。工程是在工程区(Workspace)中组织完成的。一个工程空间可以包含多个工程，可以是相关的，也可以是独立的工程。工程空间是一个开发工程的容器。





3. 当用户创建了一个工程后,系统便自动的产生一个工程空间,生成一个工程空间文件,工程扩展名为.dsw,用来存储位于工程空间一级的信息和一些其他的相关文件(.opt,.dsp等)。

4. 当工程工作空间的某些部分给隐藏了,可在图 1.1 中单击 View 菜单中的相关项将隐藏部分显示。(例如:工作空间、输出区和在调试时相关的调试信息的窗口等)

5. 在图 1.1 中,单击“插入”菜单,可以通过“新建类”项创建一个新的类,显得很方便。

6. 在图 1.1 中,“工程”菜单是 Visual C++工作台进行工程管理的主要工具。它包括以下菜单项:

- 设置活动工程命令:在多个工程同时打开时,用来选择要被激活的工程;
- 添加工程命令:像插入菜单一样,但功能更全面,它既可向工程里添加各种新的文件,又可添加工程等;
- 设置命令:将激活 project setting 对话框。通过它可以对工程的编译,连接的模式进行设置,如图 1.5 所示;
- 输出制作文件命令:生成.mak 文件,其中记述了工程编译连接的基本信息;
- 插入文件到工作区命令:该命令是将需要的工程加入到某个工程空间,一个工程空间可以有多个工程,可以是相关的,也可以是独立的。

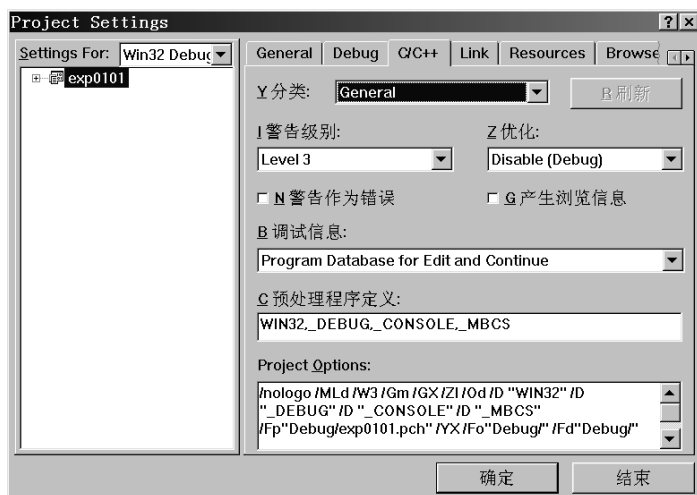


图 1.5 “工程设置”对话框

2. 第二步:源程序的编译和连接,生成可执行文件

在编译源程序之后,便该编译它了。在图 1.6 的界面,单击“编译”菜单中“编译”(compile)项或直接按快捷键(Ctr+F7)用来编译当前工作区的 C 和 C++文件,在工程空间的输出区便输出编译相关的信息。有时,源程序通不过编译,编写者会根据输出的信息进行程序源代码的检查做出相应的修改,直到程序能够顺利地通过编译。

(1) 单击“构件”(Build)命令对工程进行编译,连接生成可执行文件。可执行文件有两种版本:一种是包含调试信息的调试版本(Debug),文件较大,运行较慢,主要供调试使用;另一种是发行版本(Release),文件紧凑,运行快,通常作为开发的最终的执行文件。



其相关的配置可在图 1.6 中弹出菜单中单击“ 放置可远行配置 (Set Active Configuration) 命令，进行相关的设置，如图 1.7 所示中的设置表示对当前工程进行编译和连接时，表示只以 Debug 形式的相应的工程进行编译和连接，生成可执行文件。

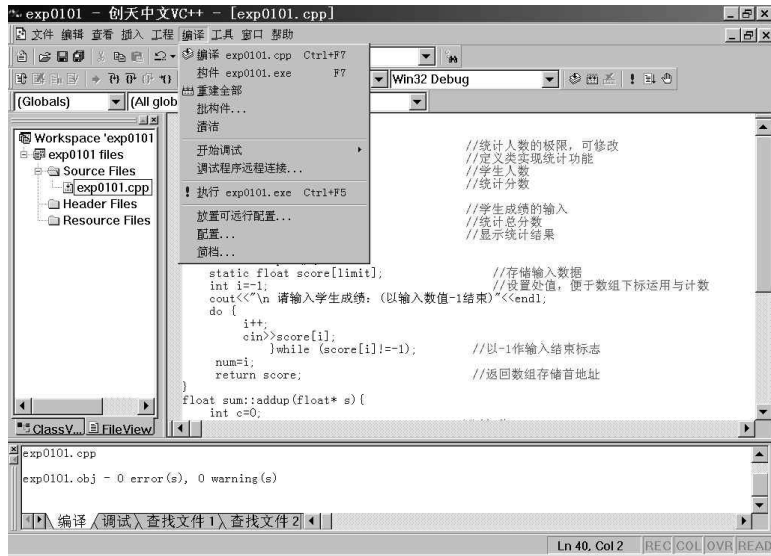


图 1.6 源程序的编译和连接界面

(2) 在生成可执行文件后，单击图 1.6 弹出菜单中“ ! 执行 (Execute) 项或按相应的快捷键 (Ctrl+F5), 执行显示如图 1.8 所示。由于在创建工程时，所选的应用程序的模式是 “ Win32 Console Application ” 它会生成 DOS 风格的应用程序。



图 1.7 输出可执行文件的模式设置

图 1.8 Dos 风格的运行窗口

相关说明：

- (1) “重建全部” (Rebuild All) 命令：将工程中的所有文件进行编译，连接成可执行文件。
- (2) “批构件” (Batch Build) 命令：将有选择地编译，在 Batch Build 对话框中，Project Configuration 列表框列出工程的所有配置形式，Batch Build 表示对指定形式的工程进行编译。



(3) 清除 (Clear) 命令用于删除指定工程的中间文件和输出文件,但并不对工程进行编译。而 重建全部 (Rebuild All) 既删除指定工程的中间文件和输出文件,又对工程进行编译,相当于“构件”加上“清除”两个命令。

3. 第三步: 调试应用程序

(1) 首先,为了让读者了解整个程序运行中相关的变化,从而使读者对全局调试过程有一个清晰的认识。(注:在图 1.9 中没有调试的工具条,可在图 1.9 界面的工具栏空白处鼠标右击,在弹出菜单中选中“调试”项,主界面便会显示调试工具条:





) 在工具条上选中第一个图标按钮,从程序运行的开始处进行显示相关的信息,如图 1.9 所示。



图 1.9 应用程序调试

然后,单击在工具条  中按钮 Step Into (F11) 进行单步向下运行。该命令可以深入到 Call 函数里 (包括库函数), 执行每一步汇编指令,此时可以打开 Registers 窗口,观察各寄存器值的变化。当遇到库函数调用时,往往工作区中显示的便是程序的汇编代码,建议此时按第三按钮  Step Over (F10) 便可结束单步向下运行,越过 Call 函数 (包括库函数)。假若已深入到调用函数中时,可以按第四按钮  Step Out 命令 (Shift+F11) 从 call 函数中跳出来。另外,在观察调试信息时,可以通过单击图 1.10 界面中“查看 (view) 菜单中“调试窗口”子菜单,选择需要的调试窗口类型,点击便可。

调试窗口有以下几种类型 (其中 Watch 和 Variables 窗口经常被使用)

- watch (Alt+3): 该命令用于 Debug 过程中激活观察窗口,可增添需要被观察的变量。
- Variables (Alt+4): 显示调试程序中各变量变量值的窗口。
- Registers (Alt +5): 用于 Debug 过程中显示当前寄存器的值。