21世纪高等学校计算机规划教材

COMPUTER

# Visual C++ 程序设计教程

Visual C++ Programming Course

陈浩杰 主编

王金玲 衣进韬 刘振 副主编

- 符合认知规律，合理安排内容，循序渐进
- 程序和原理相结合，以“程”说“理”
- 说理透彻，程序简洁

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

COMPUTER

# Visual C++ 程序设计教程

Visual C++ Programming Course

陈浩杰 主编

王金玲 衣进韬 刘振 副主编



人 民 邮 电 出 版 社

北 京

**图书在版编目（CIP）数据**

Visual C++程序设计教程 / 陈浩杰主编. -- 北京 : 人民邮电出版社, 2017.5
21世纪高等学校计算机规划教材
ISBN 978-7-115-44864-4

Ⅰ. ①V… Ⅱ. ①陈… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2017)第024371号

## 内 容 提 要

本书以 Visual C++ 6.0 为基础，详细阐述了 Visual C++程序设计的基本原理和内容。全书循序渐进，构筑了 Visual C++程序设计的几个模块，包括 C++语法、Windows C 程序设计、简单应用程序框架及以此为基础的相关类和资源的使用、文档/视图结构及以此为基础的相关类和资源的使用等。

本书可作为普通高等院校、高职高专院校计算机及相关专业的教材，也可供应用开发人员和自学者参考。

◆ 主　　编　陈浩杰
副 主 编　王金玲　衣进韬　刘　振
责任编辑　张　斌
责任印制　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：12　　2017 年 5 月第 1 版
字数：314 千字　　2017 年 5 月北京第 1 次印刷

定价：38.00 元

**读者服务热线：(010)81055256　印装质量热线：(010)81055316**
**反盗版热线：(010)81055315**
**广告经营许可证：京东工商广字第 8052 号**

# 前言

Visual C++是 Windows 程序设计的利器，它深入 Windows 程序设计的核心，与 Windows 系列操作系统有天然的亲合性，可以说，掌握了 Visual C++，就掌握了 Windows 操作系统的核心。

不过，由于 Visual C++体系庞大而且具有一定的深度，要顺利掌握 Visual C++进行程序设计还是有一定难度的。因此，作者针对 Visual C++入门难的特点，凭借多年的教学经验和心得，以及学习者对 Visual C++的认知规律，精心编写了本书。作者在书中构造了几个台阶，合理安排内容。首先从简单应用程序框架讲起，在简单应用程序框架的基础上分析了 MFC 类库中几个比较重要的类；为了使读者能顺利理解文档/视图结构，在简单应用程序框架的基础上，分析了菜单资源并对对话框资源进行介绍；在学习过文档/视图结构后又深入分析了对话框和控件的使用。

本书讲解透彻，以程说理。本书的程序简洁，摒弃了大多数 Visual C++书籍程序庞大的特点。另外，程序与原理密不可分，尤其是对比较难的原理能做到通过例程来说明。

从内容体系上来说，本书可分为以下几个部分。

（1）第 1 章介绍了 Visual C++的开发环境和工程。

（2）第 2、3 章介绍了 C++的语法。

（3）第 4 章通过 C 语言设计 Windows 程序，介绍了 Windows 程序设计的几个重要概念。

（4）第 5～8 章介绍了简单应用程序框架及几个比较重要的 MFC 类和菜单资源，并介绍了对话框资源。在第 8 章还分析了几个 Windows 公用对话框，将 MFC 的文件类和文件公用对话框放在一起讲解。

（5）第 9～11 章分析了文档/视图结构，介绍了 AppWizard 和 ClassWizard 这两个辅助工具的使用，深入分析了对话框及控件的使用。

（6）第 12～14 章可看作提高部分。

全书以简洁的程序来说明复杂的原理，程序和原理相结合，力求说理透彻，希望本书能给 Visual C++的初学者以有益的启发。

编者

2017 年 1 月

# 目录

# 绪论

## 0.1 面向对象程序设计导论

C 语言是世界上应用最广泛的几种计算机语言之一，C 语言作为一门程序设计语言为广大的计算机专业和非专业人员所喜欢。传统的“C 语言程序设计”的主要内容是用 C 语言进行 DOS 程序设计，比较流行的开发环境是 DOS 操作系统下的 Turbo C。

20 世纪 60 年代，为了应对软件危机，诞生了结构化的编程思想。C 语言就是一种结构化的编程语言，结构化的语言能比较容易地写出中等复杂的程序。然而，随着计算机技术的发展，应用软件变得越来越庞大，许多软件采用结构化的程序设计方法已经无能为力。结构化程序设计方法存在的最主要的问题是代码的重用性差。代码重用是提高生产率的关键，采用传统的面向过程的程序设计，每次进行软件开发，除了一些标准的库函数可以调用外，程序员几乎总是从零做起。

或许读者已经听说过面向对象程序设计（Object-Oriented Programming）或 OOP，面向对象程序设计的确是一种新的程序设计思想，它的出现主要是为了应对日益严重的软件危机。

面向对象的程序设计方法是以人们通常描述现实世界的方法来描述软件问题的，现实世界由各种各样的对象构成，比如计算机、汽车等，每个对象都有自己独特的属性和功能（方法），比如汽车的属性有汽车的颜色、型号等，其功能有前进、后退等。在软件世界里把对象的属性抽象为数据，把对象的功能抽象为函数，这样根据现实世界的对象就可以构造出软件世界的对象，如图 0-1 所示。

面向对象程序设计最重要的工作是构造各种现实世界对象的软件模型。这样的模型一次构造，可以多次使用。因此，面向对象的程序设计思想提高了程序代码的重用性，比如计算机的 CPU 是一个对象，则 CPU 做好后就可以在各种计算机上使用。在面向对象程序设计中，程序员所做的工作就是像搭积木一样把各种对象组合起来，构成一个功能完备的程序。

在这里要引入类的概念，类是一类对象的抽象，或者反过来说类的实例化就是对象。比如前面讲到的汽车对象，一辆红色的汽车，一辆蓝色的汽车，是否需要分别构造这两辆汽车对象呢？这样做显然是不科学的，可以先构造一个汽车类，这个汽车类都具备颜色这一属性，在实例化时指明具体对象的颜色即可，如图 0-2 所示。

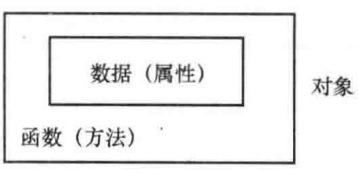


图 0-1 对象的概念

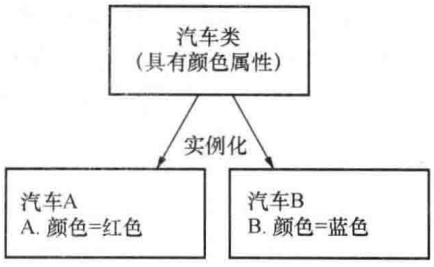


图 0-2 类与对象的关系

可见类与对象的关系就如同 C 语言中数据类型与具体的变量之间的关系。

在程序设计中可以自己构造对象，也可以直接使用别人已构造好的对象，还可以在别人所构造的对象的基础上进行修改、补充产生新的对象（类的继承性），Visual C++程序设计就是在 MFC（微软基础类库）的基础上派生自己的类，正是因为类具有继承性，程序设计中代码的重用性得到大大地提高，本书将在第 3 章将讲述有关类与对象的基本概念。

为了适应新的面向对象编程思想，C 语言也在改进，发展到 C++语言。C++是由 Bjarne Stroustrup 在美国新泽西州的贝尔实验室发明的，起初这种新语言称为“带类的 C”，1983 年正式更名为 C++。有人把 C++语言称为 C 的超集，因为 C++语言包括了 C 语言的全部内容，并增加了“类”以适应面向对象的程序设计。但这并不意味着 C++语言只能进行面向对象程序设计，它也可以进行面向过程的程序设计。

## 0.2 Windows 程序设计导论

随着 Windows 操作系统的出现，现在的应用软件大都是基于 Windows 操作系统的，因此开发 Windows 操作系统下的应用软件成为必需。

基于 DOS 操作系统的程序设计，其特点是顺序过程驱动的程序设计方法，一个程序会有一个明显地开始、明显地执行过程和明显地结束。

Windows 程序设计是基于事件（消息）驱动的。事件的含义很广泛，例如最常见的有鼠标事件。当单击鼠标左键时，产生鼠标左键单击事件；当单击鼠标右键时，产生鼠标右键单击事件。Windows 应用程序运行时都会打开一个窗口，并随时检测有无消息产生，程序员所要做的工作就是对产生的各种消息进行处理，如图 0-3 所示。

由于消息的产生是不可预期的，因此 Windows 程序设计是一种非顺序的消息驱动的程序设计方法。

C 语言可以直接进行 Windows 程序设计，称为 Windows API 程序设计，所谓 API 就是 Application Interface（应用程序接口），也就是大量的 Windows 函数。用 C 语言进行 Windows 程序设计就是对 API 函数的直接调用，这种程序设计方法思路清晰、原理简洁，但由于 API 函数数量众多且杂乱无章，因此掌握起来比较困难。而且更重要的一点是，没有采用面向对象程序设计方法，代码没有重用性。本书第 4 章将简单介绍 API 程序设计的一些基本知识。

因此有必要采用 C++进行 Windows 程序设计，称为 Visual C++程序设计。Visual C++的类库 MFC 把 API 函数进行了合理的分类封装，程序员可以从 MFC 类库派生自己的类，使 Windows 应用程序开发过程大大简化。

综合面向对象的开发思想和 Windows 程序设计的消息驱动模型，把图 0-1 和图 0-3 综合，形成图 0-4 所示的 Visual C++面向对象的 Windows 程序设计。从图 0-4 可以看到，对象之间通过消息进行通信联系，而且对象内增加了消息处理函数。

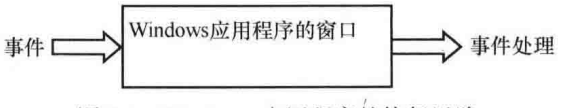


图 0-3　Windows 应用程序的执行思路

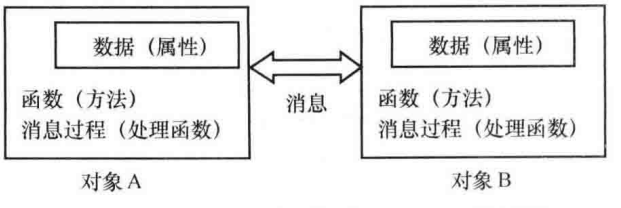


图 0-4　Visual C++面向对象的 Windows 程序设计

因此，C 语言家族可以进行四种类型的应用程序开发。

（1）用 C 语言开发基于 DOS 的应用程序。

（2）用 C++开发基于 DOS 的应用程序。

（3）用 C 语言开发基于 Windows 应用程序。

（4）用 C++开发基于 Windows 的应用程序。

本书的重点内容是介绍用 C++开发基于 Windows 的应用程序，即 Visual C++面向对象的 Windows 程序设计，但 Visual C++的开发环境同样可以开发其他类型的应用程序。

## 0.3 Visual C++的发展历史与技术特点

Visual C++这套功能强大的 Windows 应用程序开发系统是从 Microsoft C/C++演化而来的，从 Microsoft C/C++ 8.0 开始改称为 Visual C++（简称 VC++）1.0，发展到本书所讲的 Visual C++ 6.0，该版本发行至今一直被广泛地用于大大小小的项目开发。目前最新的版本是 Visual C++ 2015。

Visual C++ 6.0 有一套集成开发工具，包括各种编辑器、编译工具、集成调试器等，其主要技术特点有两个。

（1）Visual C++支持面向对象编程技术，这一技术包装了 Windows 内在的复杂的运行机制，使 Windows 编程变得简单易学。

提到面向对象编程技术，就不得不提 MFC。MFC 是由微软的 AFX 小组开发的，MFC 提供了一整套用于 Windows 应用程序开发的类。1992 年 3 月，MFC 的第一个版本随同 Microsoft C/C++ 7.0 诞生了，可见 MFC 比 Visual C++还要早，也许正是 MFC 的诞生促使 Microsoft C/C++演变成 Visual C++。

（2）Visual C++支持可视化编程，这正是 Visual 名字的由来。Visual C++提供了 AppWizard 和 ClassWizard 这两个功能强大的可视化编程向导，AppWizard 负责建立应用程序的框架，ClassWizard 则在框架的基础上负责给应用程序添加代码。

同 Microsoft Visual Studio 家族中的 Visual Basic 相比，Visual C++功能更加强大，开发人员对程序的控制更强，生成的代码效率更高，程序运行的速度更快，但要求开发人员所做的工作也就更多。从使用语言的角度来说，Visual C++比 Visual Basic 低级，正如同 C 语言与 Basic 语言的关系，因此 Visual C++一般来说适合于专业的软件开发人员使用。

# 第1章 Visual C++概述

**本章要点**

本章主要讲述 Visual C++的工作环境、工程的概念，并以 Win32 Console Application 工程为例讲述了工程的建立，源文件的添加，工程的编译、连接与执行。

本章的主要内容如下。

（1）Visual C++的工作环境。

（2）工程的概念。

（3）工程的建立，源文件的添加，工程的编译、连接与执行。

## 1.1 Visual C++的工作环境

本书选择中文版 Visual C++ 6.0 作为工作环境。打开 Visual C++，其工作环境如图 1-1 所示。

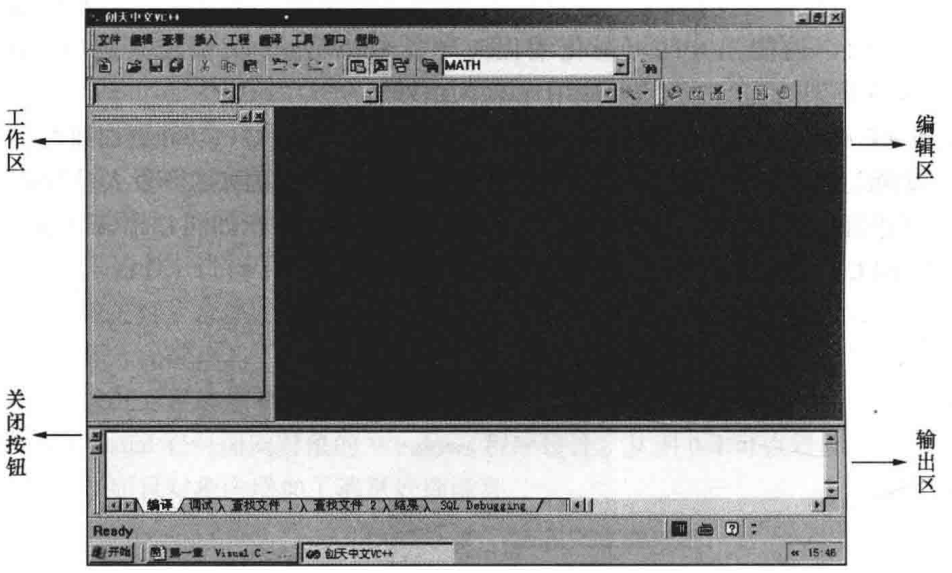


图 1-1 Visual C++的工作环境

Visual C++的工作环境可以划分为三块区域，最左边的区域是工作区，最下面的区域是输出区，最右面的区域是编辑区。

编辑区用来对源文件进行编辑，现在的编辑区是灰色的，表示没有源文件在进行编辑。

输出区（Output）的作用是：在对工程进行编译和连接后，如果程序有错误或警告，则显示在输出区，可以对照错误或警告提示进行程序修改。

这里要讲一下工程（Project）的概念：Visual C++把要完成的某个程序设计任务称之为工程。进行程序设计要先建立一个工程，当然刚建立的工程还只是个空架子，要完成具体的程序设计任务还必须往工程里添加源文件（如：.cpp 文件、.c 文件、.h 文件等）。工程管理添加在工程下的各种源文件中。这些源文件有机结合，通过编译和连接生成可执行文件。

因此工程就相当于盖房子的总体图纸，要把房子盖起来还必须购买具体的原材料（钢筋、水泥等）。

工作区（Workspace）的作用就是用来管理工程及工程下的各种源文件，在它的管理下，可以有条不紊地进行各种源文件的编辑。

可以单击工作区或输出区的关闭按钮关闭这两个区域，图 1-1 示意了输出区的关闭按钮。这两个区域关闭后能增加编辑区的面积，当对某个源程序进行编辑时，可以关闭这两个区域，需要的时候再打开。

打开方法：选择菜单命令“查看”→“工作区”或“查看”→“输出”，如图 1-2 所示，分别打开这两个区域。

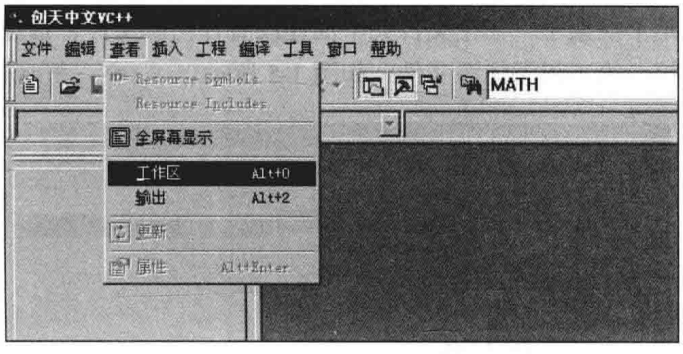


图 1-2　工作区和输出区的打开

# 1.2　工程的建立与编译、连接

### 1. 工程的建立

为了熟悉 Visual C++的工作环境，首先选择用 C 语言编写 DOS 程序来掌握工程的建立。选择菜单命令“文件”→“新建”，打开图 1-3 所示的对话框。

对话框打开时会自动选择“工程”页面，在此可以选择要创建的工程类型。要编写 DOS 程序，应选择“Win32 Console Application”（Win32 控制台应用）工程。

在“位置”栏里选择新建立的工程所存放的路径，为了便于对工程进行管理，建议新建一个文件夹集中管理工程，在“工程”栏里输入工程的名称，单击“确定”按钮，完成工程的建立。

工程建立完成后，在工作区会出现工程的名字，如图 1-4 所示。可以看到工程名下连接了 3 个文件夹，分别为源文件、头文件和资源文件文件夹，每个文件夹下面可以添加多个源文件。

需要说明的是，在工作区下面出现了两个页面，一个是“Class View”页面（对类进行管理），另一个是“File View”页面（管理工程下的各种文件）。图 1-4 为“File View”页面中的内容。

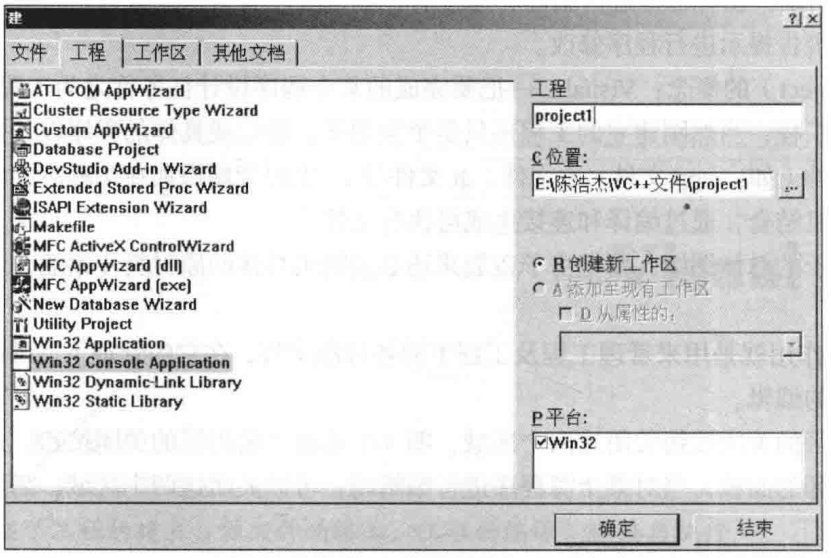


图 1-3　工程类型的选择和建立


图 1-4　工作区出现工程名

## 2. 源文件的加入与工程的编译、连接

新建的工程还只是个空架子，三个文件夹下面还没有任何源文件，选择菜单命令“工程”→“添加工程”→“新建”，给工程添加源文件，如图 1-5 所示。


图 1-5　给工程添加源文件

这时会打开图 1-6 所示的对话框，对话框打开时会自动选择“文件”页面。在“文件”页面里可以选择源文件的类型，选择“C++ Source File”，表示要往工程里添加一个 C++源文件，在“文件”栏里输入要新建的源文件名，单击“确定”按钮完成源文件的加入。

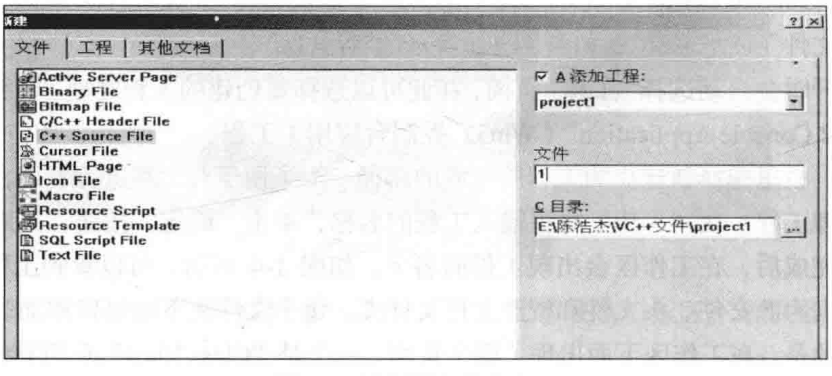


图 1-6　选择源文件的类型

另外，在图 1-6 所示的“添加工程”栏里可以看到刚建立的工程名，这表示新建的源文件将自动添加到刚建立的工程中。系统会按照源文件的类型，把建立的文件添加到工程中适当的文件夹里，选择“C++ Source File”，则新建的源文件将以.cpp 为后缀自动添加到工程中的源文件夹里。如果选择的源文件类型为“C/C++ Header File”，则将建立一个以.h 为后缀的头文件自动添加到工程中的头文件夹里。

现在编辑区会出现新建的.cpp 源文件，编辑这一文件，输入一个简单的 C 语言程序，如图 1-7 所示。


图 1-7　文件的编辑与编译、连接

编辑完成后，选择菜单命令“编译”→“编译”对工程进行编译，编译完成后会生成以.obj 为后缀的目标文件；然后选择菜单命令“编译”→“构件”对工程进行连接，连接完成后生成可执行文件。这两个步骤可以合为一个，即单击图 1-7 所示的 Visual C++工具栏上的 build 工具，一次性完成编译和连接。

当输出区显示“0 errors, 0 warnings”时，表示没有错误和警告，反之则会按序号列出错误和警告。双击错误或警告，编辑标志会出现在源文件可能出错的位置，当然有时候提示的位置不一定很准确。

### 3. 程序的执行

可以使用菜单，也可以单击工具栏上的“红色感叹号”工具，Win32 控制台工程执行后会出现一个 DOS 界面，如图 1-8 所示。

### 4. 工作区的关闭

本次工程完成后，可以关闭工程，进行下一次工程的建立。选择菜单命令“文件”→“关闭工作区”，即可完成工程的关闭。

可以根据工程建立时的路径，在磁盘中找到刚建立的工程，如图 1-9 所示，其中文件 1.cpp 就是刚编辑的源文件。应当注意 project1.dsw 文件。双击这一文件可以把整个工程打开，双击 1.cpp 文件则仅仅打开源文件。project1 工程目录下还有一个“Debug”文件夹，在里面有刚生成的可执行文件。

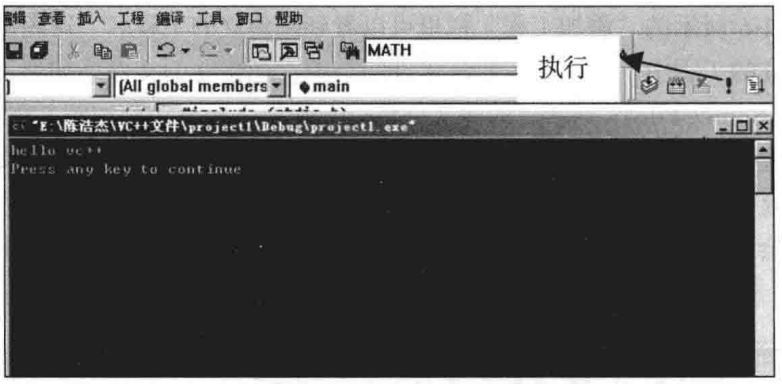


图 1-8　程序的执行

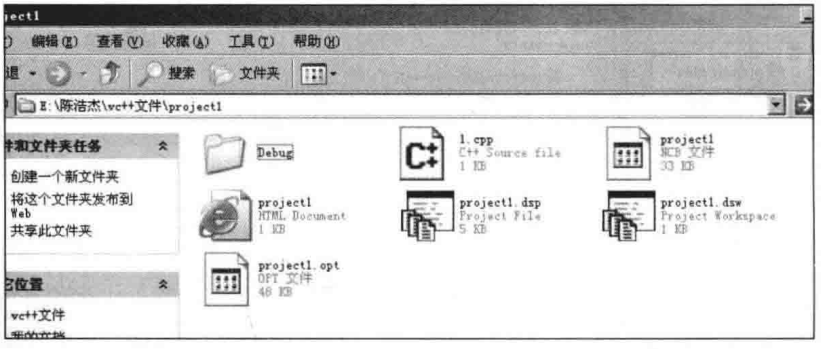


图 1-9　工程目录下的各种文件

### 5. 工程的打开

选择菜单命令“文件”→“打开工作区”，找到刚建立的工程，选择工程目录下的“project1.dsw”文件，单击“打开”按钮，即可打开工程，如图 1-10 所示。

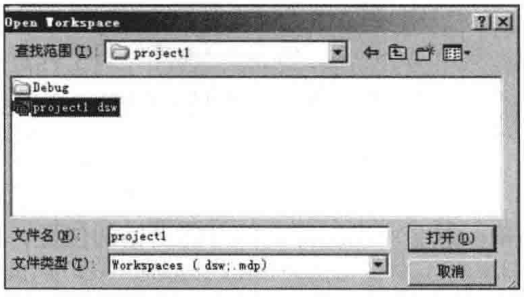


图 1-10　工程的打开

## 思考与练习

1. 在 Visual C++中，工程与各种源文件之间是什么关系？

2. 一个 Visual C++工程下会有 3 个文件夹，这 3 个文件夹分别是什么？以.cpp 为后缀的文件会添加到哪个文件夹中？以.h 为后缀的文件会添加到哪个文件夹中？

3. 以 Win32 Console Application 工程为例，简述工程的建立，文件的添加，工程的编译、连接及执行。

# 第 2 章 C++对 C 的补充

**本章要点**

本章主要讲述 C++语言对 C 语言的语法补充。内容包括 C++的输出、输入，C++的函数重载、默认函数参数与引用函数参数，C++的内存动态分配与撤销，C++的范围公解符。

本章的主要内容如下。

（1）C++的输出、输入。

（2）函数重载、默认函数参数与引用函数参数。

（3）C++的内存动态分配与撤销。

（4）范围公解符。

C++对 C 语言的补充主要体现在两个方面：一是在语言方面增加了函数重载、默认函数参数和引用等概念，二是增加了类（Class）这种新的数据类型。用类生成的变量（或叫做类的实例化）称为对象，因此，又称 C++语言为面向对象的程序设计语言。

本章将介绍 C++语言的函数重载等概念，下一章将介绍 C++的类。

## 2.1 C++的输出和输入

**1. C++的输出**

**例程 2.1** C++的输出。

建立一个 Win32 Console Application 类型的工程，然后按如下步骤处理。

（1）编辑源文件（.cpp 文件）如下。

```
#include <iostream.h>        //头文件与 C 语言不同
void main()
{
cout<<"this is "<<endl<<"the first c++ program"<<endl;
int a =5;
cout<<a<<endl;
}
```

（2）编译、连接后执行程序，输出如下：

```
this is
the frist c++ program
5
```

（3）程序说明：

① C++的输入、输出库是 iostream.h，即输入、输出流类库。

② cout 代表屏幕，“<<”表示输出运算符，endl 即 end of line，表示回车换行。

③ 可以看到，多个输出运算符“<<”可以连用。

④ C++语言一般把没有返回值的函数定义为 void 型。

⑤ 放在“//”后面的语句属于注释语句。C++的注释语句可分为两种，一种如例程 2.1 所示，一般放在一行的末尾，另一种是把注释语句放在“/*”和“*/”之间。

⑥ C++语言中，变量的定义只要放在使用之前即可，如变量定义语句 int a=5，变量定义语句不一定要放在函数体的开头部分。

### 2. C++的输入

**例程 2.2** C++的输入。

建立一个 Win32 Console Application 类型的工程，然后按如下步骤处理。

（1）编辑源文件（.cpp 文件）如下。

```
#include <iostream.h>
void main()
{
   int a,b;
   cin>>a>>b;
   cout<<"a="<<a<<endl;
   cout<<"b="<<b<<endl;
}
```

（2）执行程序，程序要求从键盘输入两个整型数给变量 a 和 b。

输入两个整数 3　4 后回车。

程序的输出如下：

```
a=3
b=4
```

（3）程序说明：

① cin 代表键盘，“>>”表示输入运算符。

② 执行程序时，从键盘输入的两个数据 3 和 4 之间要用空格分隔。

## 2.2 函数重载、默认函数参数和引用

### 1. 函数重载

在 C 语言中，可以看到这样的库函数：int abs( int x)、double fabs(double x)。这两个函数都是求某个数的绝对值，但不同的是，前者是求一个整数的绝对值，后者是求一个浮点数的绝对值。可见，由于函数参数类型的不同，要分别完成求整数和浮点数绝对值的任务，必须分别调用两个不同名的函数 abs()和 fabs()，这无疑增加了记忆负担。

在 C++语言中，可以给这两个函数起相同的名称。在函数调用时，系统会自动根据调用参数类型的不同，来选择正确的函数版本，这就叫做函数的重载。

**例程 2.3** 求两个数中的大数。

```
#include <iostream.h>
int max(int x,int y);          //函数声明，max()函数版本 1
float max(float x,float y);    //函数声明，max()函数版本 2
```

```
void main()
{
   int x1,y1,z1;
   float x2,y2,z2;
   cin>>x1>>y1;
   cin>>x2>>y2;
   z1=max(x1,y1);
   z2=max(x2,y2);
   cout<<"z1="<<z1<<endl;
   cout<<"z2="<<z2<<endl;
}
int max(int x,int y)
{
   return x>y?x:y;
}
float max(float x,float y)
{
   return x>y?x:y;
}
```

编译、运行程序，输入 3　4　3.1　4.1 后，回车。

程序输出如下：

```
z1=4
z2=4.1
```

说明：

（1）C++中一般在函数调用之前要先声明函数原型，除非函数定义在函数调用之前。

（2）例程 2.3 中，定义了两个同名的 max()函数，在调用的时候，会根据调用参数类型的不同选择正确的函数版本，当用参数 x1 和 y1 调用函数时，选择了版本一，当用参数 x2 和 y2 调用函数时，选择了版本二。

（3）函数重载或者要求重载的函数参数类型不同（例程 2.3），还有一种情况的函数重载是函数参数的个数不同也可。

（4）前面讲的输出、输入运算符“<<”和“>>”实际上是运算符的重载，因为这两个运算符默认的用法是作为移位运算符的。

**2. 默认函数参数**

一般情况下函数的实参和形参的个数应一样，但在有的情况下，实际参数个数可少于形式参数。这种情况产生的原因是函数定义时为部分形式参数指定了默认值。在函数调用时，如果默认了相应的实际参数，就用定义函数时形式参数指定的默认值代替；如果没有默认相应的实际参数，就把给定的实际参数赋给形式参数。

**例程 2.4**　默认函数参数的例子。

```
#include <iostream.h>
int max(int x,int y=100);  //指定形参 y 的默认值为 100
void main()
{
   int x1,y1,z1;
    int x2,z2;
   cin>>x1>>y1;
   cin>>x2;
   z1=max(x1,y1);          //调用 max(99,101),输出 101
   z2=max(x2);             //调用 max(99,100),输出 100
```