21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Visual C++ 应用教程

Visual C++ Programming

郑阿奇 丁有和 主编

- 深入阐述Windows程序设计思想
- 展现小综合到大综合的教学模式
- 提供可编译运行测试的具体实例

精品系列

人民邮电出版社
POSTS & TELECOM PRESS

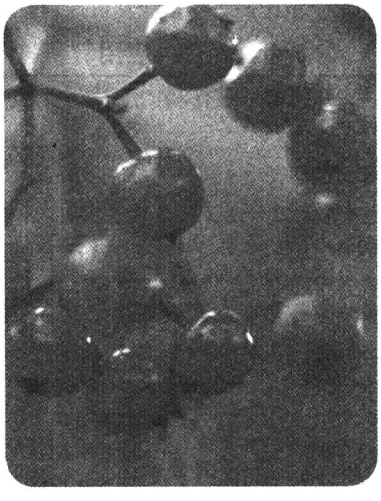
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Visual C++ 应用教程

Visual C++ Programming

郑阿奇 丁有和 主编



精品系列

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

Visual C++应用教程 / 郑阿奇，丁有和主编. —北京：人民邮电出版社，2008.10
21世纪高等学校计算机规划教材
ISBN 978-7-115-18208-1

Ⅰ. V… Ⅱ. ①郑…②丁… Ⅲ. C语言—程序设计—高等学校—教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第077106号

内 容 提 要

本书在复习C++基础知识后，先从Windows编程入手，然后引入MFC编程，再分别介绍MFC编程环境下的基本概念，进而一步一步展开。Visual C++内容安排突出基本概念和基本内容，通过一个一个的小综合，把小的知识点串起来，从而深化理解。实验部分前面突出知识点实际训练，后面是本书的小综合实践，最后的大综合突出数据库和图形应用。同时，本书把内容介绍和应用技术有机地结合起来，为使用Visual C++解决问题时可能遇到的困难提供简单的解决方案。

本书可以作为大学本科、高职高专的教材，也可供Visual C++应用开发人员参考。

21世纪高等学校计算机规划教材

Visual C++应用教程

◆ 主　　编　郑阿奇　丁有和
责任编辑　张立科
执行编辑　武恩玉

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092　1/16
印张：19.25
字数：506千字　　2008年10月第1版
印数：1－3 000册　　2008年10月河北第1次印刷

ISBN 978-7-115-18208-1/TP

定价：32.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

出版者的话

计算机应用能力已经成为社会各行业最重要的工作要求之一，而计算机教材质量的好坏会直接影响人才素质的培养。目前，计算机教材出版市场百花争艳，品种急剧增多，要从林林总总的教材中挑选一本适合课程设置要求、满足教学实际需要的教材，难度越来越大。

人民邮电出版社作为一家以计算机、通信、电子信息类图书与教材出版为主的科技教育类出版社，在计算机教材领域已经出版了多套计算机系列教材。在各套系列教材中涌现出了一批被广大一线授课教师选用、深受广大师生好评的优秀教材。老师们希望我社能有更多的优秀教材集中地呈现在老师和读者面前，为此我社组织了这套“21世纪高等学校计算机规划教材-精品系列”。

“21世纪高等学校计算机规划教材-精品系列”具有下列特点。

（1）前期调研充分，适合实际教学需要。本套教材主要面向普通本科院校的学生编写，在内容深度、系统结构、案例选择、编写方法等方面进行了深入细致的调研，目的是在教材编写之前充分了解实际教学的需要。

（2）编写目标明确，读者对象针对性强。每一本教材在编写之前都明确了该教材的读者对象和适用范围，即明确面向的读者是计算机专业、非计算机理工类专业还是文科类专业的学生，尽量符合目前普通高等教学计算机课程的教学计划、教学大纲以及发展趋势。

（3）精选作者，保证质量。本套教材的作者，既有来自院校的一线授课老师，也有来自IT企业、科研机构等单位的资深技术人员。通过他们的合作使老师丰富的实际教学经验与技术人员丰富的实践工程经验相融合，为广大师生编写出适合目前教学实际需求、满足学校新时期人才培养模式的高质量教材。

（4）一纲多本，适应面宽。在本套教材中，我们根据目前教学的实际情况，做到“一纲多本”，即根据院校已学课程和后续课程的不同开设情况，为同一科目提供不同类型的教材。

（5）突出能力培养，适应人才市场要求。本套教材贴近市场对于计算机人才的能力要求，注重理论技术与实际应用的结合，注重实际操作和实践动手能力的培养，为学生快速适应企业实际需求做好准备。

（6）配套服务完善，共促提高。对于每一本教材，我们在教材出版的同时，都将提供完备的PPT课件，并根据需要提供书中的源程序代码、习题答案、教学大纲等内容，部分教材还将在作者的配合下，提供疑难解答、教学交流等服务。

在本套教材的策划组织过程中，我们获得了来自清华大学、北京大学、人民大学、浙江大学、吉林大学、武汉大学、哈尔滨工业大学、东南大学、四川大学、上海交通大学、西安交通大学、电子科技大学、西安电子科技大学、北京邮电大学、中国林业大学等院校老师的大力支持和帮助，同时获得了来自信息产业部电信研究院、联想、华为、中兴、同方、爱立信、摩托罗拉等企业和科研单位的领导和技术人员的积极配合。在此，人民邮电出版社向他们表示衷心的感谢。

我们相信，“21世纪高等学校计算机规划教材-精品系列”一定能够为我国高等院校计算机课程教学做出应有的贡献。同时，对于工作欠缺和不妥之处，欢迎老师和读者提出宝贵的意见和建议。

人民邮电出版社

前　言

目前，Visual C++教材很多，但对初学者来说感觉难学难懂，用它解决问题更难。本书作者结合近年来 Visual C++教学和应用开发的经验，以 Visual C++ 6.05（中文版）为平台组织教材内容，采用新的切入点，既吸取现有教材中合理的内容，又在对主要内容的介绍上有所创新。

本书第一部分（第 1～9 章）内容首先回顾了 C++语言的基础知识，其中，面向对象的程序设计部分为 Visual C++的介绍奠定基础。MFC 编程和操作从 Windows 编程入手，然后引入 MFC 编程，再分别介绍 MFC 编程环境下的基本概念，从而为后面理解 Visual C++的环境和开发所遇到的问题做了初步准备。从第 3 章开始一步一步展开，内容安排突出基本概念和基本内容，通过一个一个的小综合，把小的知识点串起来，从而深化理解，为解决实际问题逐步积累基础。本书所有示例的完整源程序代码均已上机调试通过。全书各部分内容既相互联系又相对独立，方便用户根据自己的需要进行选择。

本书第二部分配合教学内容选取了典型的实验题目，实验部分前面突出知识点实际训练，后面是 Visual C++教程中的小综合实践。通过具体实例一步一步引导读者进行操作和编程（先领进门），然后提出问题思考并让读者自己进行操作和编程练习，旨在帮助学生更好地利用本书学到更多的知识。

第三部分是“综合应用”，突出数据库和图形应用。通过实验和综合应用，读者能够轻松自如地使用 Visual C++设计开发一个小的应用系统。

为了便于教学，本书为授课教师提供教学课件、所有的实例源程序文件、大综合应用源程序文件，有需要者请登录人民邮电出版社（www.ptpress.com.cn）免费下载。另外，本书还配有 Visual C++应用演示系统：在人民邮电出版社教育资源网上同步提供包含教程和实验中形成的学生成绩管理系统的所有源文件。教师可据此在课堂演示，学生可据此上机模仿。

本书把内容介绍和应用技术有机结合，为在用 Visual C++解决问题时可能遇到的困难提供简单的解决方案。所以本书既是学习 Visual C++的教材，又是 Visual C++应用的参考书。

本书由南京师范大学郑阿奇、丁有和主编，由于作者水平有限，不当之处在所难免，恳请读者批评指正。

作　者

2008 年 4 月

目 录

第一部分 Visual C++教程

第二部分 实 验

第三部分 实 习

第一部分
Visual C++教程

- 第 1 章　C++基础
- 第 2 章　MFC 编程和操作
- 第 3 章　对话框和常用控件
- 第 4 章　菜单、工具栏和状态栏
- 第 5 章　图标、光标和窗口框架
- 第 6 章　数据和文档
- 第 7 章　图形、文本和位图
- 第 8 章　视图应用框架
- 第 9 章　数据库应用

第1章 C++基础

目前，Visual C++是C++解决问题的最佳开发平台，这已经成为大家的共识。不熟悉C++，当然谈不上学习Visual C++。只有学好Visual C++，才能解决实际问题。

本章首先介绍C++面向对象的一些基础内容，以便更好地理解以后讨论的Visual C++应用程序框架。

1.1 C++概述

20世纪80年代初期，贝尔实验室的Bjarne Stroustrup在C语言的基础上创建了C++语言。C++根除了C中存在的问题，同时加入了**面向对象**的程序设计，从而引入了**类**的机制。最初的C++被称为“**带类的C**”，1983年正式命名为C++（C Plus Plus）。以后经过不断完善，形成了目前的C++。

1.1.1 C++程序创建

使用C++高级语言编写的程序称为**源程序**。由于计算机执行由0和1组成的二进制指令（称为**机器代码**），因而C++源程序是不能被计算机直接执行的，只有转换成机器代码才能被计算机执行。这个转换过程就是编译器对源代码进行**编译**和**连接**的过程，如图1-1所示。

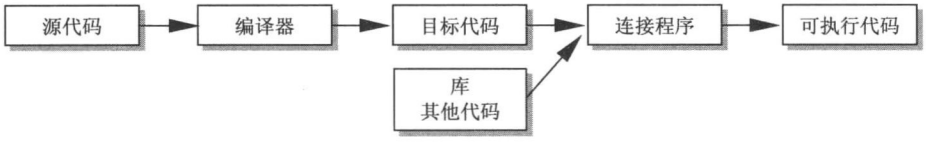


图1-1 C++程序创建过程

事实上，许多C++编程工具软件商都提供了各自的C++集成开发环境（Integrated Development Environment，IDE），使C++程序的源代码**编写**、**编译**和**连接**一体化。常见的有：Microsoft Visual C++、各种版本的Borland C++（如Turbo C++、C++ Builder等）、IBM Visual Age C++、bloodshed免费的Dev-C++等。但Visual C++在项目文件管理、调试以及操作的亲和力等方面都略胜一筹，故这里仅介绍Visual C++开发环境。

Visual C++是微软公司推出的目前使用极为广泛的基于Windows平台的可视化编程环境。Visual C++ 6.0是在以往版本不断更新的基础上形成的，由于其功能强大、灵活性好、完全可扩展以及具有强有力的Internet支持，在各种以C/C++语言为内核的Windows开发工具中脱颖而出，成为目前流行的Windows应用程序开发的集成开发环境，如图1-2所示。

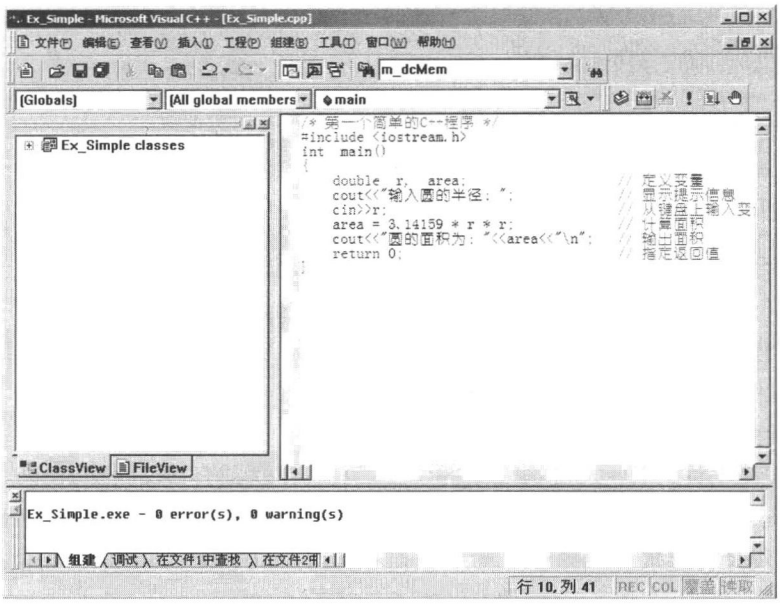


图 1-2　Visual C++ 6.05 SP6 开发环境

Visual C++ 6.0 分为标准版、专业版和企业版三种，但其基本功能是相同的。Visual C++ 6.05 中文版是在 Visual C++ 6.0 基础上进行汉化的一个版本，本书以此版本作为编程环境。为统一起见，仍称之为 **Visual C++ 6.0**，并以 Windows XP（经典桌面主题）作为操作系统。

下面以一个简单的 C++程序为例来说明在 Visual C++中创建和运行程序的一般过程。

1. 创建工作文件夹

创建 Visual C++ 6.0 的工作文件夹“D:\Visual C++应用”，以后所有创建的 C++程序都在此文件夹下，这样既便于管理，又容易查找。在文件夹“D:\Visual C++应用”下再创建一个子文件夹“第 1 章”用于存放第 1 章中的 C++程序；对于第 2 章程序就存放在子文件夹“第 2 章”中，依次类推。

2. 启动 Visual C++ 6.0

选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”，运行 Visual C++ 6.0。第一次运行时，将显示如图 1-3 所示的“每日提示”对话框。单击“下一条”按钮，可看到有关各种操作的提示。如果在“启动时显示提示”复选框中单击鼠标，去除复选框的选中标记“✔”，那么下一次运行 Visual C++ 6.0，将不再出现此对话框。单击“关闭”按钮关闭此对话框，进入 Visual C++ 6.0 开发环境。

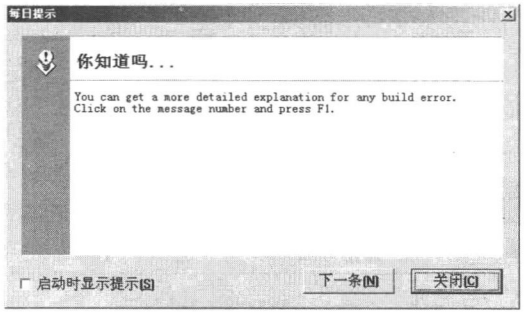


图 1-3　“每日提示”对话框

3. 添加 C++程序

（1）单击标准工具栏 上的“新建”（）按钮，打开一个新的文档窗口，在这个窗口中输入下列 C++代码。

例 Ex_Simple　一个简单的 C++程序。

```
/* 第一个简单的 C++程序 */
#include <iostream.h>
int   main( )
```

```
{
    double   r,   area;                          // 定义变量
    cout<<"输入圆的半径：";                      // 显示提示信息
    cin>>r;                                      // 从键盘上输入变量 r 的值
    area = 3.14159 * r * r;                      // 计算面积
    cout<<"圆的面积为："<<area<<"\n";            // 输出面积
    return 0;                                    // 指定返回值
}
```

（2）选择“文件”→“保存”菜单或按快捷键 Ctrl+S 或单击标准工具栏的按钮，弹出“保存为”文件对话框。将文件定位到“D:\Visual C++应用\第 1 章”文件夹中，文件名指定为“Ex_Simple.cpp”（注意扩展名.cpp 不能省略，cpp 是 C Plus Plus 的缩写，即“C++”的意思）。

此时在文档窗口中所有代码的颜色都发生改变，这是 Visual C++ 6.0 的文本编辑器所具有的语法颜色功能，绿色表示注释，蓝色表示关键词等。

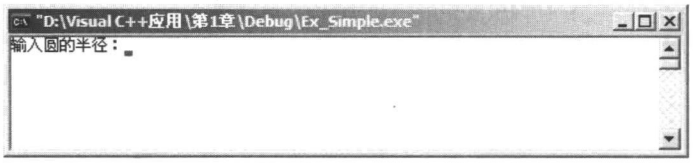
4. 编译和运行

（1）单击编译工具条上的生成工具按钮或直接按快捷键 F7，系统弹出一个对话框，询问是否为该程序创建默认的活动工作区间文件夹，单击“是”按钮，系统开始对 Ex_Simple 进行编译、连接（以下称编连），同时在输出窗口中显示编连的有关信息，当出现：

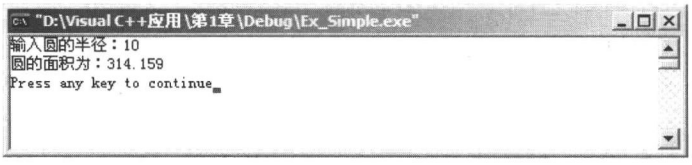
```
Ex_Simple.exe - 0 error(s), 0 warning(s)
```

表示 Ex_Simple.exe 可执行文件已经正确无误地生成了。

（2）单击编译工具条上的运行工具按钮或直接按快捷键 Ctrl+F5，就可以运行刚刚生成的 Ex_Simple.exe 了，结果弹出以下控制台窗口（其属性已被修改过）信息：



此时等待用户输入一个数。当输入 10 并按 Enter 键后，控制台窗口显示如下：



其中，“Press any key to continue”是 Visual C++自动加上去的，表示 Ex_Simple 运行后，按一个任意键将返回到 Visual C++开发环境，这就是 C++程序的创建、编连和运行过程。

上述生成的程序又可叫“控制台应用程序”，它是指那些需要与传统 DOS 操作系统保持程序的某种兼容性，同时又不需要为用户提供完善界面的程序。简单地讲，就是指在 Windows 环境下运行的 DOS 程序。控制台窗口就是一个 DOS 屏幕！

1.1.2 C++程序结构

一个程序是由若干个程序源文件组成的。为了与其他语言相区别，每一个 C++程序源文件通常是以.cpp（C Plus Plus，C++）为扩展名，它是由编译预处理指令、数据或数据结构定义以及若干个函数组成。下面就以 Ex_Simple.cpp 的程序代码（见图 1-4）为例来分析 C++程序的组成和结构。

1. main 函数

代码中，main 表示**主函数**，由于每一个程序执行时都必须从 main 开始，而不管该函数在整

```
1// [例Ex_Simple] 一个简单的C++程序
2/* 第一个简单的C++程序 */
3#include <iostream.h>
4int  main()
5{
6    double  r,  area;                          // 定义变量
7    cout<<"输入圆的半径：";                    // 显示提示信息
8    cin>>r;                                    // 从键盘上输入变量r的值
9    area = 3.14159 * r * r;                    // 计算面积
10   cout<<"圆的面积为："<<area<<"\n";          // 输出面积
11   return 0;                                  // 指定返回值
12}
```

图 1-4　Ex_Simple.cpp 的程序代码

个程序中的具体位置，因此每一个 C++程序或由多个源文件组成的 C++项目都必须包含一个且只有一个 main 函数。

在 main 函数代码中，“int main( )”称为 main 函数的**函数头**，函数头下面用一对花括号“{”和“}”括起来的部分称为 main 函数的**函数体**，函数体中包括若干条语句（**按书写次序依次顺序执行**），每一条语句都以分号“;”结束。由于 main 函数名的前面有一个 int，它表示 main **函数的类型**是整型，须在函数体中使用关键字 return，用来将其后面的值作为函数的返回值。由于 return 语句运行后，函数体 return 后面的语句不再被执行，因此除非想要函数提前结束，否则 return 语句应写在函数体的最后。

main 函数体的第 1 条（**行号为 6**）语句是用来定义两个双精度实型（double）变量 r 和 area，第 2 条（**行号为 7**）语句是一条输出语句，它将双引号中的内容（即字符串）输出到屏幕上，cout 表示标准输出流对象（屏幕），“<<”是插入符，它将后面的内容插入到 cout 中，即输出到屏幕上；第 3 条（**行号为 8**）语句是一条输入语句，cin 表示标准输入流对象（键盘），“>>”是提取符，用来将用户键入的内容保存到后面的变量 r 中；“return 0;”之前的最后一条（**行号为 10**）语句是采用多个“<<”将字符串和变量 area 的内容输出到屏幕中，后面的“\n”是换行符，即在内容输出后回车换行。

2. 头文件包含

行号为 3 的代码是 C++文件包含#include 的编译指令，称为**预处理指令**。#include 后面的 iostream.h 是 C++编译器自带的文件，称为 **C++库文件**，它定义了标准输入/输出流的相关数据及其操作，由于程序用到了输入/输出流对象 cin 和 cout，因而需要用#include 将其合并到程序中，又由于它们总是被放置在源程序文件的起始处，所以这些文件被称为**头文件**（Header File）。C++编译器自带了许多这样的头文件，每个头文件都支持一组特定的“工具”，用于实现基本输入输出、数值计算、字符串处理等方面的操作。

在 C++中，头文件包含两种格式。一是将文件名用尖括号“< >”括起来，用来包含那些由编译系统提供的并放在指定子文件夹中的头文件，这称为**标准方式**。二是将文件名用双引号括起来的方式，称为**用户方式**。这种方式下，系统先在用户当前工作文件夹中查找要包含的文件，若找不到再按标准方式查找（即再按尖括号的方式查找）。所以，一般来说，用尖括号的方式来包含编译器自带的头文件，以节省查找时间；而用双引号来包含用户自己编写的头文件。

3. 注释

程序 Ex_Simple 中的“/*...*/”之间的内容或“//”开始一直到行尾的内容是用来注释的，它的目的只是为了提高程序的可读性，对编译和运行并不起作用。正是因为这一点，所注释的内容既可以用汉字来表示，也可以用英文来说明，只要便于理解即可。

一般来说，注释应在编程的过程中同时进行，不要指望程序编制完成后再补写注释。那样只会多花好几倍的时间，更为严重的是，时间长了以后甚至会读不懂自己编写的程序。

通常，必要的注释内容应包含以下内容。

- 在源文件头部进行必要的源程序的**总体注释**：版权说明、版本号、生成日期、作者、内容、功能、与其他文件的关系、修改日志等，头文件的注释中还应有函数功能简要说明。
- 在函数的头部进行必要的**函数注释**：函数的目的/功能、输入参数、输出参数、返回值、调用关系（函数、表）等。
- **其他的少量注释**：如全局变量的功能、取值范围等。千万不要陈述那些一目了然的内容，否则会使注释的效果适得其反。

需要说明的是，“/*...*/”用来实现多行注释，它是将由“/*”开头到“*/”结尾之间所有内容均视为注释，称为**块注释**。块注释（“/*...*/”）的注解方式可以出现在程序中的任何位置，包括在语句或表达式之间。而“//”只能实现单行注释，它是将“//”开始一直到行尾的内容作为注释，称为**行注释**。

1.1.3 C++程序组成

下面再看两个C++程序。

例 Ex_Simple1　在屏幕上输出一个由星号形成的三角形。

```
// 输出星号的三角形阵列
#include <iostream.h>
void DoDraw(int num);                          // 声明一个全局函数
int   main( )
{
      int   num = 5;                           // 定义并初始化变量
      DoDraw(num);                             // 函数的调用
      return 0;                                // 指定返回值
}
void DoDraw(int num)                           // 函数的定义
{
      for (int   i = 0;   i < num;   i++)      // 循环语句
      {
            for (int   j = 0;   j <= i;   j++)
                  cout<< '*';
            cout << '\n';
      }
}
```

本程序包括两个函数：主函数main和被调用的函数DoDraw。DoDraw函数是在屏幕上输出星号的三角形阵列，这个阵列的行数以及每行星号的个数由num决定。程序运行的结果如下：

```
*
**
***
****
*****
```

在以后的C++程序运行结果中，本书不再完整显示其控制台窗口，也不再显示“Press any key to continue”，仅将控制台窗口中运行结果部分裁剪下来列出，并加以单线阴影边框，**本书作此约定**。

例 Ex_Simple2　用类的概念重写例 Ex_Draw。

```
#include <iostream.h>
class CDrawArray                               // 定义一个类
{
public:
      void DoDraw(int num);                    // 声明类的公有成员函数
};
```

```
void CDrawArray::DoDraw(int num)                    // 成员函数的实现
{
    for (int i=0; i<num; i++)
    {
        for (int j=0; j<=i; j++)
            cout<<'*';
        cout<<'\n';
    }
}
int  main( )
{
    int  num = 5;
    CDrawArray myDraw;                              // 定义类的一个对象
    myDraw.DoDraw(num);                             // 调用此对象的成员函数
    return 0;                                       // 指定返回值
}
```

虽然本程序的作用和例 Ex_Simple1 是一样的，但它引用了类的概念，是一个面向对象的 C++程序。程序中 class 后的名称是要定义的类名，该类仅声明了一个公共类型的成员函数 DoDraw。调用时，先定义该类的对象，然后像 myDraw.DoDraw(num)语句那样调用。程序运行的结果同前。

从以上几个例子可以看出，一个 C++程序往往由预处理命令、语句、函数、变量和对象、输入与输出以及注释等几个基本部分组成。

1.2　类 和 对 象

Visual C++以 C++面向对象的**类**机制作为 Windows 应用程序框架的核心。**类**是面向对象程序设计的核心，它实际上是一种新的数据类型，也是对某一类对象的概括（抽象），而类的对象就是类的一个**实例**，因此，**类**和**对象**是密切相关的。

1.2.1　面向对象程序设计

那么，什么是面向对象的程序设计呢?

在以**过程**为主的程序设计中，**问题**被看作一系列需要完成的任务，而**函数**是用于完成这些任务的主要手段。但函数是面向过程的，即它关注如何根据规定的条件完成指定的任务，强调的是**算法**。尽管每个函数都可以有自己的局部数据，但在多函数程序中，有许多重要的数据仍然需要被放置在全局数据区中，以便能被所有的函数访问。这种结构很容易造成全局数据在无意中被其他函数改动，因而程序的正确性不易保证。

另外，在大型程序中，模块化的系统是实现模块功能的函数和子程序的集合。由于用户的需求和软、硬件技术的不断发展变化，按照功能划分设计的系统模块必然是易变的和不稳定的，这样开发出来的模块可重用性不高。

为了解决面向过程程序设计中的上述问题，20 世纪 80 年代末兴起了**面向对象**程序设计（Object-Oriented Programming，OOP）方法。它把世界看成是独立**对象**的集合，在程序中，**对象是程序的基本元素**，它将数据和操作紧密地连结在一起，并保护数据不会被外界的函数意外地改动。**对象之间通过消息而相互作用**，当一个对象为完成其功能需要请求另一个对象的服务时，前者就向后者发出一条消息，后者在接收到这条消息后，识别该消息并按照自身的适当方式予以响应。

面向对象的程序设计有三个主要特征：**封装**、**继承**和**多态**。

（1）**封装**。封装是将数据和代码捆绑到一起，避免了外界的干扰和不确定性。**在 C++中，封**

装是通过类来实现的。类是用来描述具有相同属性和方法的对象的集合，它定义了该集合中每个对象所共有的属性和方法。

（2）**继承**。继承是让某个类型的对象获得另一个类型的对象的特性。在C++面向对象程序设计中，继承是指一个子类继承父类（或称为基类）的特征。通过继承可以实现代码的**重用**：从已存在的类派生出的一个新类将自动具有原来那个类的特性，同时，它还可以拥有自己的新特性。

（3）**多态**。对于相同的消息，不同的对象具有不同的反应能力。多态机制使具有不同内部结构的对象可以共享相同的外部接口，通过这种方式减少代码的复杂度。

总之，面向对象的程序设计是将问题抽象成许多类，将数据与对数据的操作封装在一起，各个类之间可以存在着继承关系，对象是类的实例，程序是由对象组成。因此，在C++面向对象程序设计时，首先设计类，定义类的属性和可执行的操作（方法），然后设计使用这些类的对象的程序。这种从低级（如类）到高级（如程序）的处理过程称为**自下向上**的编程方式。

1.2.2 类的声明

C++中，声明一个类的一般格式如下：

```
class  <类名>                                    // 声明部分
{
    private:
        [<私有型数据和函数>]
    public:
        [<公有型数据和函数>]
    protected:
        [<保护型数据和函数>]
};
<各个成员函数的实现>                             // 实现部分
```

（类体）

其中，class是类声明的关键字，class的后面是要声明的**类名**。类中的数据和函数都是类的**成员**，分别称为**数据成员**和**成员函数**。

数据成员用来描述类状态等的属性，由于数据成员常用变量来定义，所以有时又将这样的数据成员称为**成员变量**。

成员函数是用来对数据成员进行操作，又称为**方法**。注意，类体中**最后一个花括号后面的分号“;”不能省略**。

在Visual C++中，常用大写的C字母开始的标识符作为类名，C用来表示类（Class），以与对象、函数及其他数据类型的名称相区别。

类中关键字public、private和protected声明了类中的成员与类外之间的关系，称为**访问权限**。对于public成员来说，它们是公有的，可以在类外访问。对于private成员来说，它们是私有的，不能在类外访问，数据成员只能由类中的函数所使用，成员函数只允许在类中调用。而对于protected成员来说，它们是受保护的，具有半公开性质，可在类中或其子类中访问（后面还会讨论）。

从类的声明格式可以看出，类的声明可分为**声明部分**和**实现部分**。简单地说，**声明部分**将告诉使用者“做什么”，而**实现部分**是告诉使用者“怎么做”。格式中，**各个成员函数的实现**是类定义中的**实现部分**，它包含所有在类体中声明的函数的定义（即对成员函数的实现）。如果一个成员