可视化程序设计 Visual C++（第2版）

主 编 ⊙ 杨喜林 王方杰

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

可视化程序设计

Visual C++

（第2版）

杨喜林　王方杰　主　编
荣宝义　艾　铭　史学武　副主编

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内容简介

Visual C++是一个功能强大的可视化软件开发工具，是高等院校计算机及相关专业主要核心课程。

本书对可视化程序设计 Visual C++的应用与开发进行了详细、系统的介绍，内容主要包括：Visual C++程序的建立，菜单、工具栏和状态栏的创建，对话框和常用控件，窗口、文档与视图，图形绘制，数据库应用，多媒体技术等。

本书的特点是以案例为主，各章节都附有大量的实例，实例中的语句代码几乎都标有注释和说明，并且操作步骤详细，有利于引导读者更好的消化、理解和实际应用所学的知识。

图书在版编目（CIP）数据

可视化程序设计 Visual C++ / 杨喜林，王方杰主编. —2 版. —北京 ：北京理工大学出版社，2016. 5

ISBN 978-7-5682-2129-0

Ⅰ. ①可…　Ⅱ. ①杨… ②王…　Ⅲ. ①C 语言-程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2016）第 094792 号

出版发行 / 北京理工大学出版社有限责任公司
社　　址 / 北京市海淀区中关村南大街 5 号
邮　　编 / 100081
电　　话 / (010)68914775(总编室)
　　(010)82562903(教材售后服务热线)
　　(010)68948351(其他图书服务热线)
网　　址 / http：//www. bitpress. com. cn
经　　销 / 全国各地新华书店
印　　刷 / 三河市天利华印刷装订有限公司
开　　本 / 787 毫米×1092 毫米　1/16
印　　张 / 30
字　　数 / 698 千字
版　　次 / 2016 年 5 月第 2 版　2016 年 5 月第 1 次印刷
定　　价 / 75. 00 元

责任编辑 / 高　芳
文稿编辑 / 高　芳
责任校对 / 陈玉梅
责任印制 / 王美丽

前　言

Visual C++1.0 是 Microsoft 公司 1993 年推出的一个功能强大的可视化软件开发工具。随着其版本的不断更新，1998 年推出的 Visual C++6.0 有了较大的改进，得到了广泛的应用，并已成为目前专业程序员进行软件开发的首选工具。

Visual C++6.0 不仅是一个 C++编译器，而且是一个基于 Windows 操作系统的可视化集成开发环境（Integrated Development Environment，IDE）。它提供了一个高效的 Windows 编程环境，将程序和资源的编辑、编译、调试和运行融为一体，具有其独特的优越性能。由于它任何时候都可以调用所有的 Win32 函数，因而能灵活处理像 Java 等不能处理的写磁盘和串口访问这样的任务。

Visual C++的最大特点就是提供对面向对象技术的支持，它利用类把大部分与用户界面设计有关的 Windows API 函数封装起来，通过 MFC（Microsoft Foundation Class）类库的方式提供给开发人员使用，大大提高了程序代码的重用性。它还提供一个功能强大的应用程序生成向导——AppWizard，它能使用户编程省去繁琐的初始化代码，自动生成一个运行程序框架，用户只需按自己的意图添加相关的代码，就能得到一个满意的应用程序。

本书实例中的每条语句代码几乎都标有注释，便于读者阅读和理解程序的设计思维，更好地掌握程序设计方法。一般在提出问题前都会首先举出实际例题，让读者先有感性认识，再进行理论上的讲解，使读者感到清晰、易懂。

本书适合高等学校作为教材使用，也可作为计算机专业人员和非专业人员研究可视化程序设计的实用参考书。

本书由烟台南山学院杨喜林教授、徐州工业职业技术学院王方杰担任主编，南通科技职业学院荣宝义、吉林城市职业技术学院艾铭、史学武担任副主编。

由于编者水平有限，有不当之处，请读者批评指正。

编　者

目　录

第 1 章

Visual C++程序的建立

C++面向对象程序设计语言是在 C 语言的基础上发展起来的，它与传统的程序设计方式不同，是一种新的程序设计范型。它对降低软件的复杂性，改善其重要性和维护性，提高软件的生产效率，有着十分重要的意义。因此面向对象程序设计被普遍认为是程序设计方法的一场实质性的革命。

Visual C++支持面向对象程序设计，是 Microsoft 公司推出的目前应用最为广泛的基于 Windows 平台的可视化编程工具。

使用 Visual C++的强大功能，可以开发 Windows 应用程序，设计完成色彩亮丽的可移动的图形图像及千变万化的文字信息和广泛流行的企事业管理、银行、电信、商业、交通、航空航天、教育、游戏等众多的实用软件。

1.1 C 程序和 C++程序

只有在具备 C 和 C++编程基础后，才能更好地运用 Visual C++编程工具开发 Windows 应用程序。下面先用 C 语言、C++语言编写一个同样的显示学生信息的程序，来熟悉一下它们的编程方法。特别是通过 C++程序设计，进一步掌握面向对象的编程技术，这将对学好可视化的 Visual C++程序设计起到促进作用。

[例 1.1] 用 C 语言编写显示学生信息程序

将 Visual C++6.0 软件安装到本机器后，创建其桌面快捷方式。在常用的 e 盘上建一个文件夹，名为：vcpp。

双击桌面 Visul C++6.0 快捷方式图标，在打开的窗口中选择 File→New 命令，在弹出的对话框中展开 File 选项卡，选择 C++source File 选项，然后在右边 File 处写要建的文件名：学生信息 1，在其下面的 Location 即路径名处写：e:\vcpp，然后单击 OK 按钮，如图 1.1 所示，在出现的界面（文档窗口）上可输入 C 程序，之后在弹出的对话框中依次单击 Build→Build→是→是→！（编译运行）按钮，便出现图 1.2 的结果。输入的 C 程序如下：

```
#include <stdio.h>
#include <string.h>
struct student              //定义 student 结构体类型
{  long num;                //学号
   char name[20];           //年龄
   char sex[10];            //性别
```

```
    float score;        //成绩
}stu,*p;        //直接定义 student 结构体类型变量 stu 和 student 结构体类型指针*p
int main()
{    p=&stu;                          //student 结构体的起始地址赋给指针变量 p
     stu.num=89101;                   //将学号赋给 stu 结构体变量中的成员 num 中
     strcpy(stu.name,"李明");         //将姓名赋给 stu 结构体变量中的成员 name
     strcpy(stu.sex,"男");            //将性别赋给 stu 结构体变量中的成员 sex
     stu.score=89.5;                  //将成绩赋给 stu 结构体变量中的成员 score
//以下两个 printf()函数输出的结果是相同的
printf("学号 No.:%ld\n 姓名 name:%s\n 性别 sex:%s\n 成绩 score:%2.1f\n",
                                        stu.num,stu.name,stu.sex,stu.score);
                                        //stu.num 表示 stu 结构体变量中的成员 num
printf("\n 学号 No.:%ld\n 姓名 name:%s\n 性别 sex:%s\n 成绩 score:%2.1f\n",
                                          p->num,p->name,p->sex,p->score);
                                  //p->num 是代替(*p).num 的书写形式，也就是 p->num
                              //等价于(*p).num，它表示 p 指向结构体变量 stu 中的成员 num
return 0;
}
```

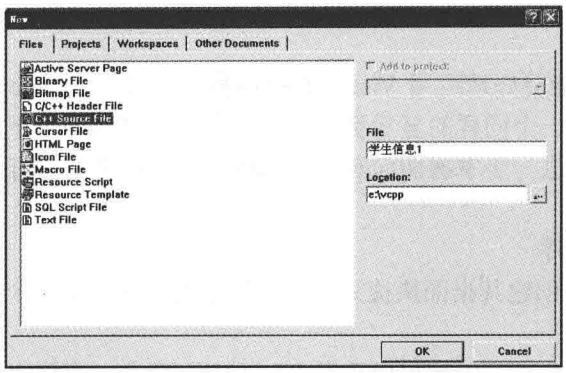

图 1.1 建立 C 语言程序对话框

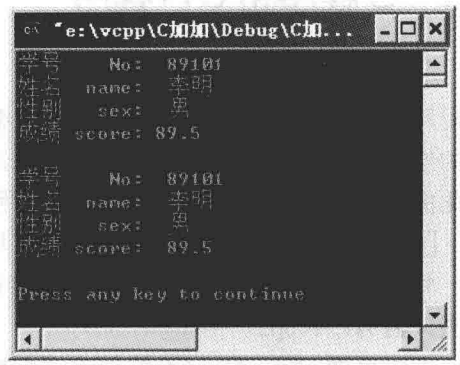

图 1.2 C 语言程序运行结果

[例 1.2] 用 C++编写显示学生信息程序

（1）双击桌面 Visul C++6.0 快捷方式图标，在打开的窗口中选择 File→New 命令，在弹出的对话框中展开 projects 选项卡，选择 Win32 Console Applicator 选项，然后在 Project name 处写文件名：学生信息 2，在 Location 处写 e:\vcpp，在弹出的对话框中依次单击 OK→Finish→OK 按钮。

（2）再在主菜单处选择 File→New 命令，在弹出的对话框中展开 File 选项卡，选择 C++ Source 选项，右边 Add to Project 要置于选中状态，并显示出刚才输入的文件名，然后在 File 处写：学生信息 2.cpp，单击 OK 按钮，如图 1.3 所示。在出现的窗口界面（文档窗口）中输入 C++程序。再依次单击 Build→Build（编译运行）按钮，结果如图 1.4 所示。输入的 C++程序如下：

```
#include <string.h>
#include <iostream.h>
//类的说明部分
class student   //定义类(class)  student
{   private:  //私有部分
    public:   //公共部分
      long num;           //学号
      char name[20];      //姓名
      char sex[10];       //性别
      float score;        //成绩
      void sc();          //输出函数
    protected:            //保护部分
};
//类的实现部分
void student::sc() //在student类里面定义的函数sc()的具体实现
{   student stu;
    student *p=&stu;
    stu.num=89101;
    strcpy(stu.name,"李明");
    strcpy(stu.sex,"男");
    stu.score=89.5;
   //以下两种输出结果都是相同的
    cout<<"学号    No:  "<<stu.num<<"\n";//一般对象成员
    cout<<"姓名  name:  "<<stu.name<<"\n";
    cout<<"性别   sex:  "<<stu.sex<<"\n";
    cout<<"成绩 score: "<<stu.score<<"\n\n";
    cout<<"学号    No:  "<<p->num<<"\n";//指针对象指向的成员
    cout<<"姓名  name:  "<<p->name<<"\n";
    cout<<"性别   sex:  "<<p->sex<<"\n";
    cout<<"成绩 score:  "<<p->score<<"\n\n";
}
//主函数体
void main()      //主函数，程序从这里开始执行
{   student xx;  //定义student类的对象xx
    xx.sc();//sc()是xx(student)的成员函数，xx.sc()是表示访问xx的成员函数sc()
}
```

可以看到，C 语言是先定义结构体和结构体变量、结构体指针，之后在主函数 main()里用结构体变量和结构体指针输出其变量内容。C++是先定义类，在主函数里用类定义对象，再通过对象里的成员函数输出其变量内容。C 和 C++的最大区别是 C++定义的类里可以有成

员函数，通过成员函数完成所要完成的工作，在程序结构上有了深刻变化，是面向对象的。而 C 的结构体里不能定义函数，它不具备类的功能，程序是结构化的、面向过程的。

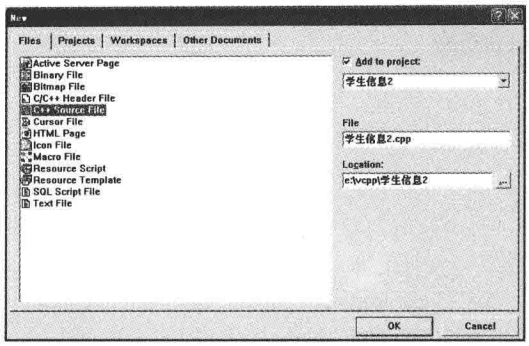

图 1.3　建立 C++程序对话框

图 1.4　C++程序运行结果

1.2　面向对象的编程技术

面向对象包括面向对象分析和面向对象程序设计。面向对象程序设计技术的提出，主要是为了解决传统的结构化程序设计所不能解决的代码重用问题。

1.2.1　类与对象

前面在例 1.2 程序中已经定义了类 student 和其对象 xx。“类”就是对具有相同数据和相同操作的一组相似对象的定义，即类是对具有相同特征和行为的一个或多个对象的描述。在面向对象程序设计中，“类”是一个最重要的概念。

类的结构（也即类的组成，如例 1.2 中的：学号 num、姓名 name、性别 sex、成绩 score、输出函数 sc()）是用来确定一类对象的行为的，而这些行为是通过类的内部数据结构和相关操作来确定的。类通过其成员函数（如例 1.2 中的函数 sc()）来描述这些行为，而成员函数又被称为类的对象向其他对象所提供的服务。

与类对应的就是对象了。对象就是由某个特定的类所描述的一个具体的实例。例如，“车”是一个类，一种具体的车如“客运车”就是一个具体的对象了。对象是类的实例，类是具有公共特性的对象的抽象。

对象的定义：一个类定义后，就可以定义该类的对象，如：例 1.2 中的 student xx；

其定义的格式为：<类名> <对象表名>；

其中，类名是用户定义过的类的标识符，对象名可以有一个或多个，多个时要用逗号分隔。被定义的对象即可以是一个普通对象，也可以是一个数组对象或指针对象。如：

```
CMeter myMeter, Meters[5], *Meter;
```

这时，myMeter 是类 CMeter 的一个普通对象，Meters［5］和 Meter 分别是类的一个数组对象和指针对象。

一个对象的成员就是该对象的类所定义的数据成员（成员变量）和成员函数。访问对象的成员变量、成员函数和访问变量和函数的方法是一样的，只不过要在成员面前加上对象名

和成员运算符“.”，例如：例 1.2 中的 xx.sc()。具体表示方式如下：

```
< 对象名 >.< 成员变量 >
< 对象名 >.< 成员函数>(<参数表>)
```

例如：myMeter.m_nPercent，myMeter.SetPos(5)，Meters[0].StepIt()；

需要说明的是，一个类对象只能访问该类的公有型成员，而对于私有型成员则不能访问。上述成员 m_nPercent、SetPos()、StepIt()都是 public（公有）访问类型。

若对象是一个指针，则对象的成员访问形式如下：

```
<对象指针名>-><成员变量>
<对象指针名>-><成员函数>(<参数表>)
```

例如（见例 1.4 主函数中所定义的）：

```
Date *datel; // 定义 Date 类的指针对象
datel=new Date(1998,4,28); // Date()是类 Date 中定义的构造函数，这里是执行构造函数，
//并为类 Date 开辟一片内存单元，且将这片内存单元的首地址赋给指针变量 datel。
datel->showDate(); //访问 Date 类中的成员函数 showDate()
```

“->”是一个表示成员的运算符，它与“.”运算符的区别是：“->”用于表示指向对象的指针的成员，而“.”用于表示一般对象的成员。

1.2.2 类及其成员变量、成员函数的声明和定义

类的定义格式一般可分为说明部分和实现部分。说明部分用来说明该类中的成员，包含数据成员的说明和成员函数的说明。成员函数是用来对数据成员进行操作的，又称为“方法”。实现部分是对成员函数的定义。概括来说，说明部分是告诉使用者“做什么”，而实现部分则告诉使用者“怎么做”。类的一般定义格式如下：

```
class <类名>
{   public:
      公有数据成员的说明;
      公有成员函数的说明;
    protected:
      保护数据成员的说明;
      保护成员函数的说明;
    private:
     私有数据成员的说明;
     私有成员函数的说明;
};
< 各个成员函数的实现 >
```

class 是定义类的关键字，<类名>是用户定义的类的名字，通常用大写的 C 字母开始的标示符作为类名。花括号{ }内是类的说明部分，说明该类的成员。类的成员包含数据成员和成员函数两部分。public 称为类的公有部分，这部分的数据成员和成员函数可由程序中的函数（包括类内和类外）访问，即它对外是完全开放的。private 称为类的私有部分，这一部分的数

据成员和成员函数只能由本类的成员函数访问，而类外部的任何访问都是非法的。实现了访问权限的有效控制。protected 称为类的保护部分，这部分的数据成员和成员函数可以由本类的成员函数访问，也可以由本类的派生类的成员函数访问，而类外的任何访问都是非法的。即它是半隐蔽的。

关键字 public、private 和 protected 被称为访问权限修饰符或访问控制修饰符。它们在类体内（即花括号{ }内）出现的先后顺序不限，并且允许多次出现，可用它们来说明类成员的访问权限。

<各个成员函数的实现>是类定义中的实现部分，这部分包含所有在类体内说明的函数的定义，如例 1.2 中的 void student::sc();

如果一个成员函数在类体内定义了，则实现部分将不再出现。如果所有的成员函数都在类体内定义，则实现部分可以省略。需要说明的是，当类的成员函数的函数体在类的外部定义时，必须由作用域运算符“::”来通知编译系统该函数所属的类。

为了进一步明确类的定义，除了例 1.2 外，下面再给出一个日期类定义的例子。

[例 1.3] 定义日期类

```
#include <iostream.h>
//类的说明部分
class Date
{  public:   //公有部分
      void SetDate(int y,int m,int d);//说明为 year、month、day 设置初值的函数
      int IsLeapYear();   //说明判别是否是闰年的函数
      void Print();       //说明输出函数
   private://私有部分
      int year,month,day; //说明用来表示年、月、日的整型变量
};
//类的实现部分
void Date::SetDate(int y,int m,int d) //year、month、day 设置初值函数的实现
{  year = y;  //年
   month = m; //月
   day = d;   //日
}
//判别是否是闰年函数的实现,能被 4 整除与不能被 100 整除或能被 400 整除
int Date::IsLeapYear()
{  return((year%4 == 0 )&& ((year%100)!= 0)||(year%400 == 0)); }
void Date::Print() //输出函数的实现
{  cout<<year<<"."<<month<<"."<<day<<endl; } // 输出年、月、日
//主函数体
void main()
{  Date da;    //定义 Date 类对象 da
   da.SetDate(2000,6,28);//执行类 Date 的函数 SetDate(2000,6,28)，参数为年、月、日
```

```
    da.IsLeapYear();//执行类 Date 的函数 IsLeapYear()
    da.Print(); //执行类 Print()函数
}
```

说明：

(1) 程序中出现的“::”是用来表示某个成员函数是属于哪个类的。该类还可以如下定义：

```
class Date
{  public:
      void SetDate(int y,int m,int d)
      {  year = y; month = m; day = d;  }
      int IsLeapYear()
      {  return((year%4 == 0 )&& ((year%100)!= 0)||(year%400 == 0)); }
      void  Print() //输出函数体
      {  cout<<year<<"."<<month<<"."<<day<<endl; }
   private:
      int year,month,day;
 }
```

可以看出，程序中对成员函数的实现（即函数的定义）都写在了类体内，因此类的实现部分被省略了。

(2) 在类体中不允许对所定义的数据成员进行初始化。

(3) 类中的数据成员的类型可以是任意的，包括整型、浮点型、字符型、数组、指针、引用、无类型(void)以及对象等。

(4) 经常习惯性地将类定义的说明部分或者整个定义部分（包含实现部分）放在一个头文件中。

(5) 程序中的 cout 叫输出流，它可以是一个整数、实数、字符及字符串，cout 中的插入符“<<”可以连续写多个，每个后面可以跟一个要输出的常量、变量、转义序列符、对象以及表达式等。

(6) 与 cout 对应的输入流是 cin，它可以获得多个键盘的输入值，其中提取符“>>”可以连续写多个，每个后面跟一个表达式，该表达式通常是获得输入值的变量或对象。若程序中有如下语句：

```
int nNum1,nNum2,nNum3;
cin>>nNum1>>nNum2>>nNum3;
```

这两条语句是定义 3 个整型变量，并要求用户从键盘上输入 3 个数。输入时，必须在 3 个数值之间加上一些空格来分隔，空格的个数不限，最后按回车键结束输入。

1.2.3 构造函数和析构函数

构造函数和析构函数是在类体中说明的两种特殊的成员函数。构造函数的功能是在创建对象时，使用给定的值将对象初始化；析构函数与构造函数正好相反，其功能是释放一个对象，在删除对象前，用它来做一些内存释放等清理工作。

1. 构造函数

由于在类的定义中不能对数据成员进行初始化，为了能给数据成员自动设置某些初始值，需要使用构造函数。构造函数具有以下特征。

（1）构造函数的名字与类名相同，否则编译程序将把它当作一般的成员函数来处理。

（2）构造函数没有返回值。

（3）构造函数的功能是对对象进行初始化，且一般只对数据成员进行初始化。在构造函数中一般不做赋初值以外的事情。

（4）构造函数不能像其他成员函数那样被显示地调用，它是在对象创建时被调用的。

（5）在一个类中可以定义多个构造函数。

[例 1.4] 定义构造函数

```
#include <iostream.h>
class Date
{  public:
     Date(int y,int m,int d);  //构造函数
     void setDate(int y,int m,int d);//一般的成员函数，设置日期函数
     void showDate();//一般的成员函数，显示日期函数
   private:
     int year;    //私有成员变量，年
     int month;  //私有成员变量，月
     int day;     //私有成员变量，日
};
Date::Date(int y,int m,int d)     //构造函数的实现
{  cout<<"Constructing Function "<<endl;
   year = y;     //为 year 赋值
   month = m;   //为 month 赋值
   day = d;      //为 day 赋值
}
void Date::setDate(int y,int m,int d) //一般成员函数，设置日期函数的实现
{  year=y; //为 year 赋值
   month=m; //为 month 赋值
   day=d; //为 day 赋值
}
inline void Date::showDate() //一般成员函数，显示日期函数的实现
{  cout<<year<<"."<<month<<"."<<day<<endl;  } //输出年、月、日
void main()
{  Date *date1;  //定义指针对象 date1
   date1=new Date(1998,4,28);  //构造函数，为 year、month、day 赋初值
   //以上两条语句可合成写：Date *date1 = new Date(1998,4,28);
   cout<<"Date1 output1:"<<endl; //输出字符串 Date1 output1
```