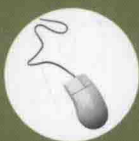


高等院校计算机系列教材



GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI



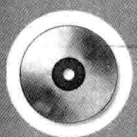
C++与面向对象程序设计实验教程

总主编：陈火旺 主 编：傅 明 湖南省计算机学会规划教材 中南大学出版社

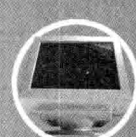


C++YMXDXCXSJSYJC
C++YUMIANXIANGDUIXIANG
CHENGXUSHEJISHIYANJIAOCHENG

高等院校计算机系列教材



GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI



C++与面向对象程序设计实验教程

总主编：陈火旺 湖南省计算机学会规划教材 中南大学出版社

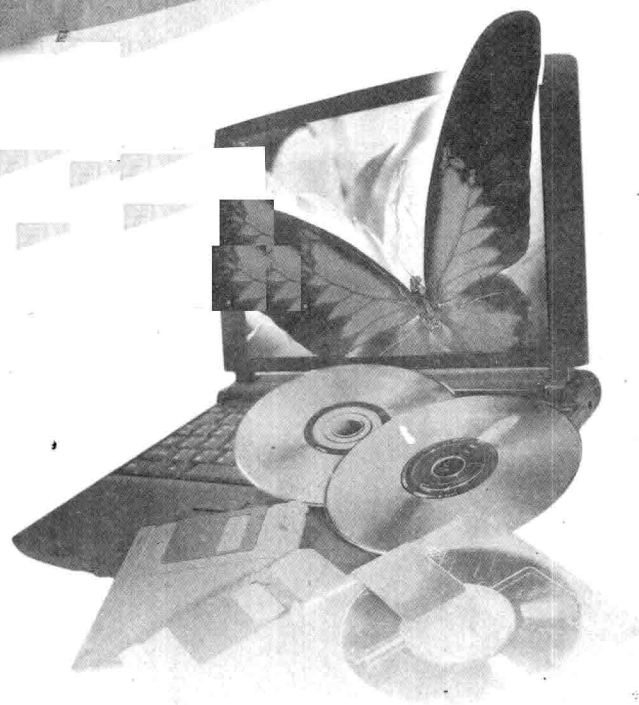
主 编：傅 明

副主编：涂光平 王新祥 陈超云

编 委：（按姓氏笔画排序）

刘翌南 刘 霞 刘国英 肖晓丽

陈 曦 何建新 夏卓群 满君丰



高等院校计算机系列教材编委会

总 主 编 陈火旺

执行总主编 孙星明

副 总 主 编 李仁发 陈志刚

编 委(按姓氏笔画排序)

王志英	刘任任	刘 宏	刘振宇
孙星明	羊四清	阳小华	阳爱民
余绍黔	吴宏斌	张新林	李仁发
李正华	李 军	李勇帆	李 峰
杨路明	沈 岳	肖建华	肖晓丽
陈火旺	陈志刚	罗庆云	金可音
胡志刚	赵 欢	徐建波	殷建平
郭国强	高守平	庾 清	黄国盛
龚德良	傅 明	彭民德	曾碧卿
蒋伟进	鲁荣波	谭骏珊	谭敏生

目 录

实验一	熟悉开发环境和创建 Win32 窗口程序	(1)
实验二	基本数据类型、表达式和基本语句	(5)
实验三	类和对象、继承和派生	(7)
实验四	多态函数和虚函数、运算符重载	(13)
实验五	MFC 基本应用程序的建立	(18)
实验六	对话框、按钮以及编辑框控件	(22)
实验七	列表框、组合框及滑动条等控件	(33)
实验八	选单、工具栏和状态栏	(44)
实验九	框架窗口、视图和文档	(56)
实验十	数据库编程	(70)

实验一 熟悉开发环境和创建 Win32 窗口程序

一、目的和要求

- (1) 掌握 Visual C++ 6.0 的启动和退出方法。
- (2) 熟悉用应用程序向导创建一个 Win32 窗口程序。
- (3) 掌握源代码文件的新建、打开、保存和关闭等基本操作。
- (4) 掌握 Visual C++ 项目的编译、链接和执行。
- (5) 掌握代码简单语法错误修正和调试的一般过程。

二、上机内容

熟悉 Windows 操作系统的环境和基本操作以及 Visual C++ 6.0 开发平台,并在此基础上利用向导程序创建一个 Win32 窗口程序示例 HelloWorld。

三、上机步骤

- (1) 打开计算机,启动 Windows 操作系统。
 - (2) 创建 Visual C++ 6.0 的工作文件夹,例如“D:\实验源代码”,以后自己创建的所有应用程序都可以放在这个文件夹下,这样便于管理和查找。
 - (3) 选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”。
 - (4) 单击 File | New 菜单项,弹出 New 对话框。
 - (5) 在 New 对话框中选择 Project 标签,从左边的列表框中选择项目类型 Win32 Application,在 Project name 框中,输入新项目名 Ex1_1,并指定项目存储位置“D:\实验源代码”,如图 1-1 所示。
 - (6) 在 Platforms 框中,指定项目的开发平台,确认“Win32”被选中。
 - (7) 完成以上设置以后,单击 OK 按钮,根据所选的项目类型,会出现相应的 Wizard(向导),如图 1-2 所示。
 - (8) 选中 A typical “Hello World” application 单选按钮,单击 Finish 按钮,弹出项目信息对话框,如图 1-3 所示。
 - (9) 单击 OK 按钮,生成项目框架。
- 完成上述步骤以后,AppWizard 生成了如下文件:
- ① 头文件 Ex1_1.h
这个文件是应用程序的主头文件,它含有所有全局符号和用于包含其他头文件的#include 伪指令。
 - ② 源文件 Ex1_1.cpp
这个文件是应用程序的主源文件。

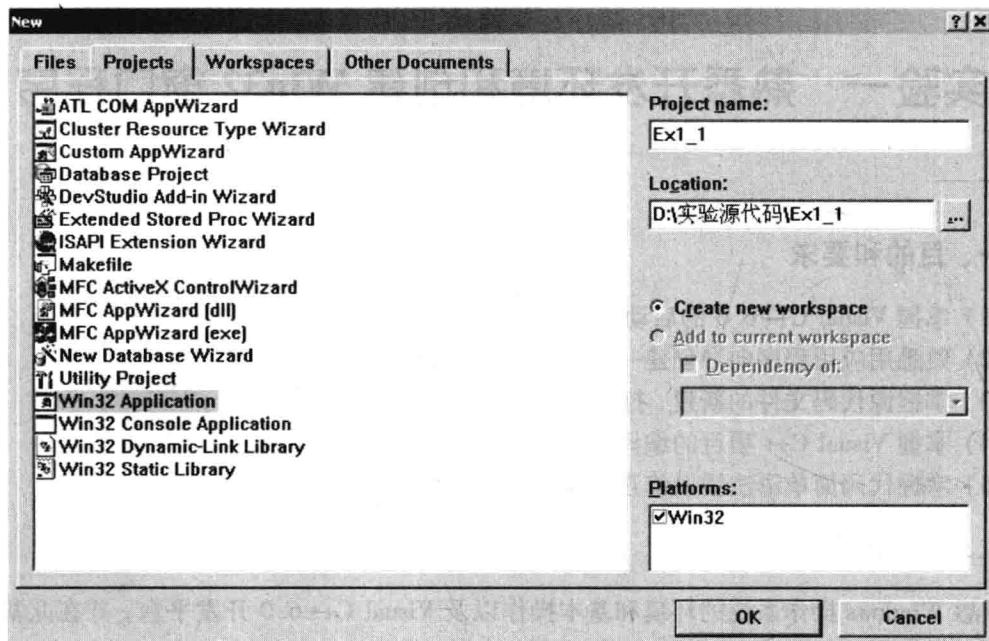


图 1-1 创建 Win32 窗口程序的项目

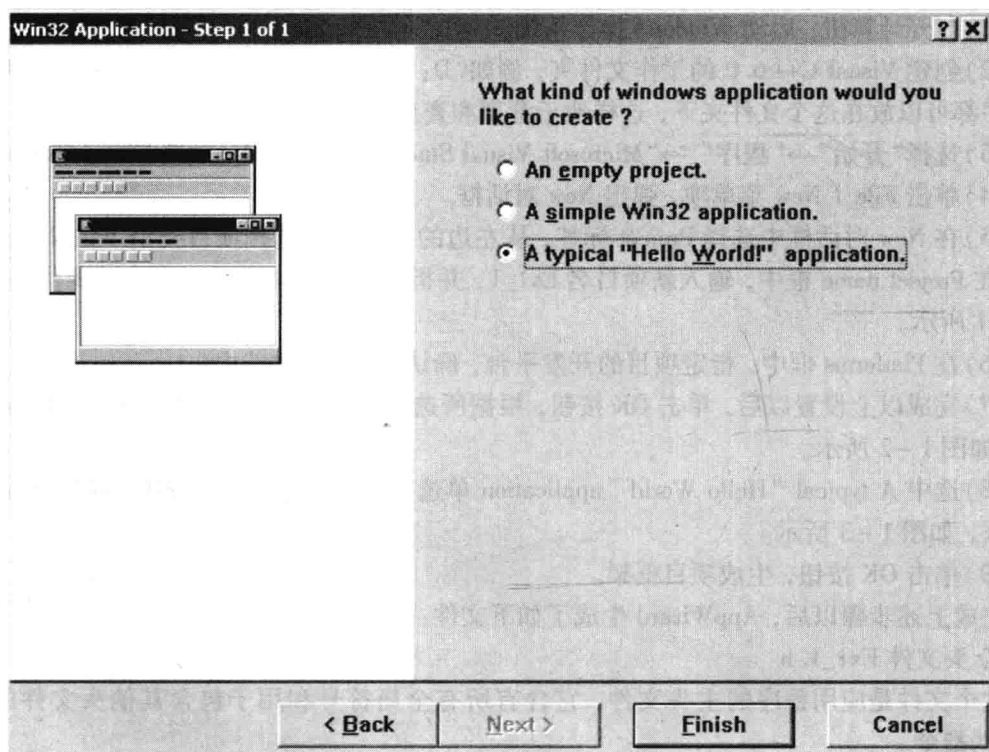


图 1-2 Win32 窗口程序的类型选择

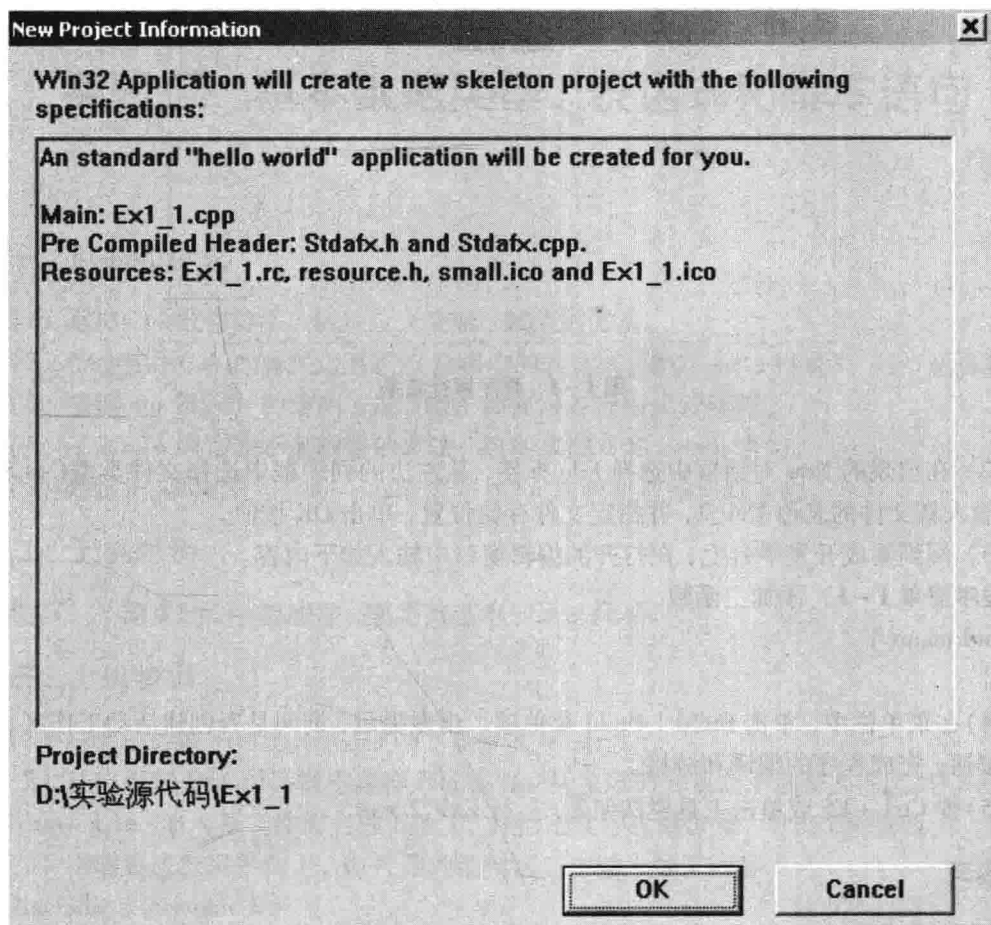


图 1-3 项目信息对话框

③ 资源文件

AppWizard 会创建一些与资源文件相关的文件。如 Ex1_1.rc、resource.h、Ex1_1.rc2，这些文件是项目的资源头文件及其资源文件。资源文件含有一般应用程序的默认菜单定义和加速键表、字符串表。它还指定了默认的 About 对话框和一个图标文件(RES\Ex1_1.ico)。Ex1_1.rc2 用于存放 Visual Studio 不可直接编译的资源。

④ 预编译头文件

Stdafx.cpp 和 Stdafx.h 这两个文件用于创建一个预编译的头文件 Ex1_1.PCH 和一个预定义的类型文件 Stdafx.cpp。

(10) 按 CTRL + F5 或单击工具栏按钮, 编译、链接并运行程序, 运行结果如图 1-4 所示。

四、选做实验

(1) 在集成开发平台上, 单击 File | New 菜单项。

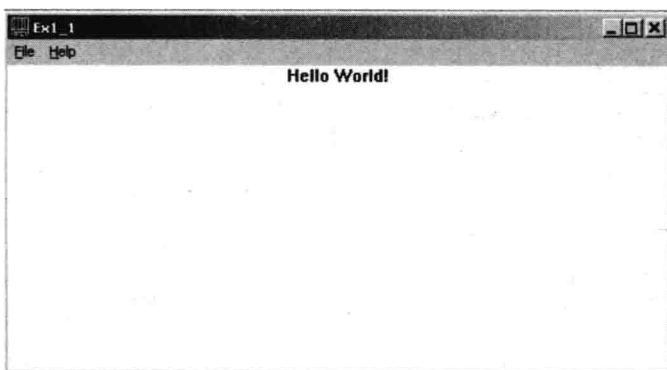


图 1-4 程序运行结果

(2) 在出现的 New 对话框中选择 File 标签, 从左边的列表框中选择文件类型 C++ Source File, 输入新文件的名称 Ex1_2, 并指定文件存储位置, 单击 OK 按钮。

(3) 回到集成开发平台上, 在打开的编辑窗口中输入如下内容。

程序清单 1-1 添加主函数

```
void main()  
{  
}
```

(4) 在菜单栏中, 单击 Build | Build 菜单项。出现提示, 询问是否创建活动工作区, 单击“是”按钮, 完成程序的编译和链接。

(5) 按 Ctrl + F5 或单击工具栏按钮 , 运行 Ex1_2.exe。

练习

请用应用程序向导创建一个控制台应用项目 (Win32 Console Application), 其运行结果如图 1-5 所示。

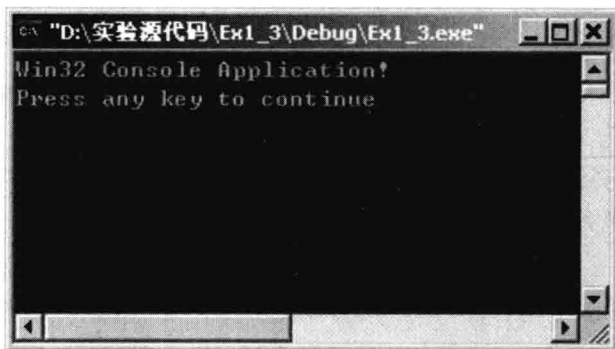


图 1-5 运行结果

实验二 基本数据类型、表达式和基本语句

一、目的和要求

- (1) 掌握C++ 数据类型, 熟悉定义变量、赋值的方法。
- (2) 学会使用C++ 的有关运算符以及相应的表达式: 增1(++)和减1(--)运算符。
- (3) 掌握 cin 和 cout 的使用方法, 能正确使用各种格式转换符。
- (4) 了解C++ 语言表示逻辑量的方法, 熟练掌握 if 和 switch 语句。
- (5) 熟悉用 while 语句、do...while 语句和 for 语句实现循环的方法。

二、上机内容

编写一个简单的C++ 源程序, 熟悉其基本的语法风格。

三、上机步骤

- (1) 在集成开发平台上, 单击 File | New 菜单项。
- (2) 在出现的 New 对话框中选择 File 标签, 从左边的列表框中选择文件类型C++ Source File, 输入新文件的名称 Ex2_1, 并指定文件存储位置, 单击 OK 按钮。
- (3) 回到集成开发平台上, 在打开的编辑窗口中输入如下内容。

```
#include <iostream.h>
void main()
{
    char c1, c2, c3;
    c1 = 97; c2 = 98; c3 = 99;
    cout << c1 << c2 << c3 << endl;
}
```

- (4) 在菜单栏中, 单击 Build | Build 菜单项。出现提示, 询问是否创建活动工作区, 单击“是”按钮, 完成程序的编译和连接。

- (5) 按 Ctrl + F5 或单击工具栏按钮, 运行 Ex2_1.exe。

四、选做实验

在程序 Ex2_1 的基础上进行下面的修改:

- (1) 将结果按数字输出。有两种方法: 一种是将 c1, c2 和 c3 的变量类型由 char 变为 int 或 short; 另一种是变量类型保持不变, 在输出语句中加入类型的强制转换, 如:

```
cout << (short)c1 << (short)c2 << (short)c3 << endl;
```

与 Ex2_1 比较, 看其输出结果有什么不同?

(2) 添加程序代码。在前面的基础上, 使 Ex2_1 变为:

```
#include <iostream.h>
void main()
{
    short c1, c2, c3;
    c1 = 92; c2 = 98; c3 = 99;
    cout << c1 << c2 << c3 << endl;
    short m, n, k;
    m = ++c1;
    n = ++c2;
    k = c3++;
    cout << m << ", " << n << ", " << k << endl;
}
```

编译并运行, 看看结果是否与自己理解的相同? 再在 main 函数体的最后加下列语句:

```
cout << ++m << ", " << n++ << ", " << k++ << endl;
```

看看结果有什么变化?

(3) 设置输出格式。在 main 函数体的最后添加下列语句:

```
cout.width(10);
cout.fill('#');
cout << oct << m << "\t" << hex << n << "\t" << dec << k << endl;
```

观察结果是什么?

练习

用 C 和 C++ 编写一个功能相同的简单程序, 比较它们不同的语法风格。

实验三 类和对象、继承和派生

一、目的和要求

- (1) 掌握类和对象的定义和使用方法。
- (2) 掌握定义构造函数和析构函数、重载构造函数的方法。
- (3) 掌握静态成员函数和常量的用法。
- (4) 掌握多重继承和派生类的方法。
- (5) 能正确理解 public, private 和 protected 这三种访问权限及继承方式的区别。
- (6) 掌握在派生类中初始化基类成员的方法。

二、上机内容

定义一个人员类 CPerson, 包括数据成员: 姓名、编号、性别和用于输入、输出的成员函数。在此基础上派生出学生类 CStudent(增加成绩)和教师类 CTeacher(增加教龄), 并实现对学生和教师信息的输入、输出。

三、上机步骤

1. 输入程序 Ex3_1.h

操作步骤如下:

- (1) 在集成开发平台上, 单击 File | New 菜单项。
- (2) 在出现的 New 对话框中选择 File 标签, 从左边的列表框中选择文件类型 C/C++ Header File, 输入新文件的名称 Ex3_1.h, 并指定文件存储位置, 单击 OK 按钮。
- (3) 回到集成开发平台上, 在打开的编辑窗口中输入如下内容。

```
#include <string>
#include <iostream.h>
class CPerson
{
public:
    CPerson()
    {
        strcpy(pName, "");
        strcpy(pID, "");
    }
    CPerson(char * name, char * id, bool isman = 1)
    {
```

```
        Input( name, id, isman);
    }

void Input( char * name, char * id, bool isman)
{
    setName( name);
    setID( id);
    setSex( isman);
}

void Output( )
{
    cout << "姓名:" << pName << endl;
    cout << "编号:" << pID << endl;
    char * str = bMan?"男":"女";
    cout << "性别:" << str << endl;
}

public:
    char * getName( ) const
    { return ( char * ) pName; }
    void setName( char * name)
    {
        int n = strlen( name);
        strncpy( pName, name, n);
        pName[ n ] = " \0";
    }
    char * getID( ) const
    { return ( char * ) pID; }
    void setID( char * id)
    {
        int n = strlen id);
        strncpy( pID, id, n);
        pID[ n ] = " \0";
    }

    bool getSex( ) { return bMan; }
    void setSex( bool isman) { bMan = isman; }

private:
    char pName[20];
    char pID[20];
    bool bMan;
```

```
};  
class CStudent :public CPerson  
{  
    public:  
        CStudent( char * name, char * id, bool isman = 1 );  
        ~ CStudent() {}  
        void InputScore( double score1, double score2, double score3 );  
        void Print();  
        CPerson student;  
    private:  
        double dbScore[ 3 ];  
};  
  
class CTeacher:public CPerson  
{  
    public:  
        CTeacher( char * name, char * id, bool isman = 1, int years = 10 );  
        ~ CTeacher() {}  
        void Print();  
    private:  
        int nTeachYears;  
};
```

(4) 选择 File→Save 选单或按快捷键 Ctrl + S 或单击标准工具栏中的 Save 按钮, 保存程序。

2. 输入程序 Ex3_1.cpp

操作步骤如下:

(1) 在集成开发平台上, 单击 File | New 菜单项。

(2) 在出现的 New 对话框中选择 File 标签, 从左边的列表框中选择文件类型 C++ Source File, 输入新文件的名称 Ex3_1.cpp, 并指定文件存储位置, 单击 OK 按钮。

(3) 回到集成开发平台上, 在打开的编辑窗口中输入如下内容。

```
#include <iostream.h>  
#include "Ex3_1.h"  
CStudent:: CStudent ( char * name, char * id, bool isman ) : student( name, id, isman )  
{  
    dbScore[ 0 ] = 0;  
    dbScore[ 1 ] = 0;  
    dbScore[ 2 ] = 0;  
}  
  
void CStudent::InputScore( double score1, double score2, double score3 )
```

```
{
    dbScore[0] = score1;
    dbScore[1] = score2;
    dbScore[2] = score3;
}

void CStudent::Print()
{
    student.Output();
    for(int i=0;i<3;i++)
        cout << "成绩" << i+1 << " : " << dbScore[i] << endl;
}

CTeacher::CTeacher(char * name, char * id, bool isman, int years)
{
    nTeachYears = years;
    Input(name, id, isman);
}

void CTeacher::Print()
{
    Output();
    cout << "教龄:" << nTeachYears << endl;
}

void main()
{
    CStudent stu("ZhangSan", "20015008031");
    cout << stu.getName() << endl;
    cout << stu.student.getName() << endl;
    stu.Print();
    stu.student.setName("LiSi");
    stu.student.setSex(0);
    stu.InputScore(80, 90, 85);
    stu.Print();
    CTeacher tea("Tu", "0840");
    tea.Print();
    tea.setID("200150080222");
    tea.Print();
}
```

3. 运行程序 Ex3_1

操作步骤如下:

(1)单击工具栏中的“Build”按钮或按 F7 键,系统出现一个对话框,询问是否将此项目

的工作文件夹设定源文件所在的文件夹，单击[是]按钮，系统开始编译。若此时出现编译错误，暂时不管它。

(2) 选择“Project”→“Add to Project”→“Files”命令，在弹出的“Insert Files into Project”对话框中，选定前面的程序文件 Ex3_1.h，单击[OK]按钮。

(3) 再次编译。

(4) 运行程序。

四、选做实验

定义一个正方形类，该类包括：

(1) 受保护类数据成员，表示正方形的边长(取值的范围为1~30个‘*’);

(2) 四个成员函数，功能分别为：①取边长；②设置边长；③画正方形；④在构造这类对象时，初始化正方形边长。

编写 main() 函数实现如下功能：

(1) 创建一个边长为5的正方形；

(2) 调用成员函数打印正方形的形状；

(3) 由用户输入正方形边长，修改边长的值；

(4) 重新打印正方形。

按照题目要求正方形 Square 类包括受保护类数据成员 len，包括公有类成员函数——带参构造函数，取边长函数 GetLen()、设置边长函数 SetLen()、画正方形函数 DrawSquare()。

上机操作步骤如下：

(1) 在集成开发平台上，单击 File | New 菜单项。

(2) 在出现的 New 对话框中选择 File 标签，从左边的列表框中选择文件类型 C++ Source File，输入新文件的名称 Ex3_2.cpp，并指定文件存储位置，单击 OK 按钮。

(3) 回到集成开发平台上，在打开的编辑窗口中输入如下内容。

```
#include <iostream.h>

class Square
{
protected:
    int len;
public:
    Square() {}
    Square(int x) { len = x; }
    int SetLen(int x);
    int GetLen() { return len; }
    void DrawSquare();
}

int Square::SetLen(int x)
{
}
```

```
        if( x < 1 || x > 30)
            return 0;
        len = x;
        return 1;
    }

void Square::DrawSquare()
{
    int i, j;
    for(i = 0; i < len; i++)
        for(j = 0; j < len; j++)
            cout << ' * ' ;
        cout << endl;
    }
}

void main()
{
    Square s(5);
    cout << "draw the square, len = " << s.GetLen() << endl;
    s.DrawSquare();
    cout << "input the new side length:" ;
    int x;
    cin >> x;
    if( ! s.SetLen(x) )
        cout << "error, length should be between 1 and 30!";
    else
        { cout << "draw the square, len = " << s.GetLen() << endl;
          s.DrawSquare();
        }
}
```

(4) 编译并运行程序。

练习

一个图形是由一个圆和一个矩形构成,要求求解图形的面积。设计 3 个类,其中两个是基类,一个基类描述圆,一个基类描述矩形;第三个派生类是由一个圆和一个矩形构成的图形类。圆类包含数据成员半径和求圆面积的成员函数,矩形类包含数据成员长和宽、求矩形面积的成员函数。派生的图形类提供输出图形面积的函数及实现初始化的带参构造函数。请编写其源代码,并编译运行。

实验四 多态函数和虚函数、运算符重载

一、目的和要求

- (1) 理解虚函数的特性。
- (2) 理解纯虚函数和抽象类的特性。
- (3) 掌握用虚函数实现运行时的多态和编写通用程序的方法。
- (4) 掌握用友元函数重载运算符的方法。
- (5) 掌握用成员函数重载运算符的方法。

二、上机内容

定义一个抽象类 CShape, 包含纯虚函数 Area() (用来计算面积) 和 SetData() (用来重设形状大小)。然后派生出三角形 CTriangle 类、矩形 CRect 类、圆 CCircle 类, 分别求其面积。最后定义一个 CArea 类, 计算这几个形状的面积之和, 各形状的数据通过 CArea 类构造函数或成员函数来设置。编写一个完整的程序。

三、上机步骤

- (1) 在集成开发平台上, 单击 File | New 菜单项。
- (2) 在出现的 New 对话框中选择 File 标签, 从左边的列表框中选择文件类型 C++ Source File, 输入新文件的名称 Ex4_1, 并指定文件存储位置, 单击 OK 按钮。
- (3) 回到集成开发平台上, 在打开的编辑窗口中输入如下内容。

```
#include <iostream.h>

class CShape
{
public:
    virtual float Area() = 0;
    virtual void SetData(float f1, float f2) = 0;
};

class CTriangle:public CShape
{
public:
    CTriangle(float h = 0, float w = 0)
    { H = h; W = w; }
    float Area()
    { return(float)(H * W * 0.5); }
```