

《面向对象程序设计与 C++》自编作业手册

翟高寿

北京交通大学计算机学院

2012 年 2 月编写

2016 年 2 月修订

2018 年 8 月修订

2019 年 9 月修订

2020 年 9 月修订

编修说明

《面向对象程序设计与 C++》自编作业手册根据自制电子课件 2007-2008-II 版形成初稿并根据教学实践和多次修改完成。

作业题标记说明：

1、程序作业注释性[前缀]标记：

FS 复数运算程序

ZD 质点平面运动轨迹跟踪程序

XS 学生管理程序

D&SJX 平面坐标点与三角形面积计算程序

YLS 有理数运算程序

JZ 矩阵运算程序

Y&YZT 圆与圆柱体程序

PMTX 平面图形拼盘程序

TYBDX 通用比大小程序

DZ 堆栈程序

YCCL 字符串异常处理程序

FXSEK 泛型算法库应用程序

BZRQL 标准容器类应用程序

ZJW 最简不使用 MFC 类库的 Windows 应用程序

ZJMFC 最简使用 MFC 类库的 Windows 应用程序

ZJMFCAW 最简 MFC AppWizard [exe]项目程序

2、题号后标☆或★者为必做题，既不标☆也不标★者为选做题，其中标★且要求在课程平台上提交的必做题作业如下：

- 1、[D&SJX]平面坐标点与三角形面积计算程序（对应作业编号为 6）
- 2、[XS_C 版]学生管理程序（对应作业编号为 7）
- 3、[FS_D 版]复数运算程序（对应作业编号为 10）
- 4、[Y&YZT]圆与圆柱体程序（对应作业编号为 13）
- 5、[TYBDX]通用比大小程序（对应作业编号为 15）
- 6、[YCCL]字符串异常处理程序（对应作业编号为 17）

作业实验报告提交要求

（1）以**学号姓名作业注释标记**（譬如 19280000 翟高寿 D&SJX）为文件夹名称，里面存放源程序文件（包括.h 和.cpp 文件）、作业实验报告 Word 文档（命名与文件夹名称保持一致，譬如 19280000 翟高寿实验报告 D&SJX.docx）。（2）作业实验报告 Word 文档中仅需给出开发运行平台描述、测试用例设计、运行结果截屏和总结或问题答案。其中，程序运行结果截图可适当调整窗口宽度及大小，并利用 Alt 按键+PrtScn 按键方式屏幕拷贝整个运行窗口，而总结部分可包含问题、心得、经验、启示、同学间作业检查发现等。（3）针对每道作业打包和压缩一次，命名与文件夹名称保持一致，即以**学号姓名作业注释标记**为压缩文件名称，譬如 19280000 翟高寿 D&SJX.rar

1[FS_A☆] 设计、编写和调试面向对象 C++程序，支持和实现任意两个复数的加法运算。要求程序设计一个复数类（规范化设计规格要求参表所示）：（A）拥有两个 double 类型的成员变量分别表示实部和虚部；（B）拥有四个成员函数分别实现复数对象的设置、显示及加法运算功能。其中，一个复数对象设置成员函数（返回值类型为 void、无参数）要求根据用户键盘输入实现复数对象整体（包括所有成员变量）的设置；另一个复数对象设置成员函数（返回值类型为 void、两个 double 类型参数）要求根据传入的两个参数分别实现复数对象实部和虚部成员变量的设置；复数对象显示成员函数（返回值类型为 void、无参数）要求程序能够以类似于 5+4i、5-4i、5、4i、-4i 的形式简明、正确地显示复数对象的取值；复数类的加法成员函数（返回值类型及参数类型均为程序自定义的复数类）要求程序能够正确实现两个参数复数对象（当前对象和参数对象）的加法运算，并返回结果对象。主函数则总控实现三个复数类对象（两个充当加数对象的复数和一个表示和对象的复数）的创建，包括其中两个复数对象的设置（程序能够友好提示用户分别输入实部和虚部值、进而正确接收和调用复数对象设置成员函数实现复数对象的设置）和显示以及二者相加与结果显示的功能（注意全部是通过调用复数类的对应成员函数来实现而非其自身直接实现）。

class CComplex
double mR; //实部 double mI; //虚部
void SetValue(); //复数对象设置操作函数 1 void SetValue(double zR, double zI); //复数对象设置操作函数 2 void ShowValue(); //复数对象显示操作函数 CComplex Add(CComplex zC); //复数加法运算功能函数

2[FS_B☆] 设计、编写和调试面向对象 C++程序，支持和实现任意两个复数的加、减、乘、除运算（通过对应四个成员函数来实现），要求复数类拥有构造函数和析构函数，并拥有支持复数对象正确设置和显示功能的成员函数。其中，复数类的加/减/乘/除成员函数（返回值类型及参数类型均为程序自定义的复数类）要求程序能够正确实现两个参数复数对象（当前对象和参数对象）的加/减/乘/除运算，并返回结果对象；要求构造函数支持缺省情况下的复数对象初始化（即未指定实部与虚部的情况下初始化为 0，只指定实部的情况下虚部初始化为 0，指定实部和虚部的情况下对应设置实部和虚部），同时显示符号串 “<=” 和复数对象取值（显示方式与要求同作业题[FS_A]复数对象显示成员函数）并换行，如

<=5+4i

；要求析构函数显示复数对象取值（显示方式与要求同作业题[FS_A]复数对象显示成员函数）和符号串 “=>” 并换行，如

5+4i=>

；其它未尽要求同作业题[FS_A]。复数类规范化设计规格要求参表所示。主函数则总控实现复数类对象的创建、设置、显示及任意两个复数相加/相减/相乘/相除与结果显示功能（注意是通过调用复数类的对应成员函数来实现而非其自身直接实现）。

class CComplex
double mR; //实部 double mI; //虚部
CComplex(double zR=0, double zI=0); //构造函数 ~CComplex(); //析构函数 void SetValue(); //复数对象设置操作函数 1 void SetValue(double zR, double zI); //复数对象设置操作函数 2 void ShowValue(); //复数对象显示操作函数

CComplex Add(const CComplex& zC) const; //复数加法运算功能函数 CComplex Subtract(const CComplex& zC) const; //复数减法运算功能函数 CComplex Multiply(const CComplex& zC) const; //复数乘法运算功能函数 CComplex Divide(const CComplex& zC) const; //复数除法运算功能函数

3[ZD] 设计、编写和调试面向对象 C++程序，支持和实现质点平面运动轨迹跟踪功能（所有质点对象平面运动方式及速度均为每次有效按键[运动方向以上 **S**、下 **X**、左 **Z**、右 **Y** 字母键表示]若干个坐标单位）。要求程序设计一个质点类：（A）拥有三个整型的成员变量分别表示质点对象当前的横、纵坐标值和平面移动速率值；（B）拥有构造函数和析构函数，要求构造函数随机实现质点坐标值的初始化，同时显示符号串 “<=” 和质点坐标取值（用方括号括注和逗号分割横纵座标）并换行，如 <=[0, 0]；而析构函数显示质点坐标取值（用方括号括注和逗号分割横纵座标）和符号串 “=>” 并换行，如 [0, 0]=>；（C）拥有四个成员函数分别实现质点对象的上移、下移、左移、右移功能，相应函数原型均为无参和无返回值（即返回值类型为 void）；（D）拥有一个成员函数实现用户操作界面接收功能函数原型亦为无参和无返回值（即返回值类型为 void），提示和接收用户输入质点运动方向按键，并调用对应函数实现质点的正确移动和方位显示。质点类规范化设计规格要求参表所示。主函数则总控实现一个质点类对象的创建、显示及其在用户键盘操作下的移动功能（注意是通过调用质点类的对应成员函数来实现而非其自身直接实现）。

class CParticle
int mX, mY; //横、纵坐标 int mV; //质点移动速率
CParticle(int zV=1); //构造函数 ~CParticle(); //析构函数 void SetPosition(int zX, int zY); //质点对象位置设置操作函数 void SetVelocity(int zV); //质点对象速率设置操作函数 void ShowPosition(); //质点对象位置显示操作函数 void MoveTrace(); //基于键盘控制的质点对象跟踪操作函数 void MoveEastward(); //质点对象右移操作函数 void MoveWestward(); //质点对象左移操作函数 void MoveNorthward(); //质点对象上移操作函数 void MoveSouthward(); //质点对象下移操作函数

4[XS_A☆] 设计、编写和调试面向对象 C++程序，支持和实现学生基本信息及其成绩的录入、输出及比较等管理功能。要求程序设计一个学生类：（A）拥有一个字符数组类型的成员变量用于表示学生的学号，拥有一个字符指针类型的成员变量用于表示学生的姓名，拥有一个整数数组类型的成员变量来存放和表示学生所学课程的成绩（假定所有学生修课均为 5 门）；（B）拥有构造函数、拷贝构造函数和析构函数，要求构造函数（无参数）实现学生对象学号的初始化（取值为“无效”），而析构函数则在判断学生姓名所占空间不为空的情况下释放其所占用空间；（C）拥有两个成员函数分别实现学生对象所有数据成员属性（注意姓名需首先根据情况进行对应大小的空间申请）的设置（无参数，函数内部接收输入后设置，注意友好性）和显示功能；（D）拥有两个学生成绩比较成员函数（二者返回值类型均为 void，但一个成员函数拥有一个形如 const CStudent & zS 学生类常量引用的参数对象，而另一个成员函数则拥有一个形如 const CStudent *pS 学生类常量参数对象指针），这两个比较函数均根据当前学生对象和参数学生对象的总成绩比较结果分别输出“A 比 B 学习成绩好!”、“A 和 B 学习成绩不相上下!”、“A 比 B 学习成绩差!”（注意函数功能实现时 A、B 分别以当前学生对象和参数

学生对象的实际姓名替代)。学生类规范化设计规格要求参表所示。主函数则总控实现至少两个学生对象的创建、设置和显示及两个学生成绩比较与学生与自身比较等功能（注意是通过调用学生类的对应成员函数来实现而非其自身直接实现）。

class CStudent
char mNo[9]; //学号 char *mpName; //姓名 int mScore[5]; //五门功课成绩
CStudent(); //构造函数 CStudent(const CStudent& zS); //拷贝构造函数 ~CStudent(); //析构函数 void SetStudent(); //学生对象（学号姓名成绩）设置操作函数 void ShowStudent() const; //学生对象显示操作函数 void CompareScore(const CStudent &zS) const; //学生成绩比较函数 1 void CompareScore(const CStudent *pS) const; //学生成绩比较函数 2

5[XS_B☆] 设计、编写和调试面向对象 C++程序，支持和实现全班学生基本信息及其成绩的录入和输出等管理功能。要求程序设计一个学生类，其设计规格参作业题[XS_A]，但要求学号在构造函数中能自动生成，由固定不变字符串“1728”和计数部分构成。为此，需要为学生类增加和定义一个初始值为 0 的整型静态成员，其伴随学生对象的创建（即构造函数的调用）将顺次递增，进而相应初始化对应学生学号为 17280001、17280002、17280003，以此类推。学生类规范化设计规格要求参表所示。要求主函数总控实现全班所有学生对象（以学生对象数组形式定义和使用）的创建、设置和统一输出（注意是通过调用学生类的对应成员函数来实现而非其自身直接实现）。

class CStudent
char mNo[9]; //学号 char *mpName; //姓名 int mScore[5]; //五门功课成绩 static int mCount; //学生对象计数
CStudent(); //构造函数 CStudent(const CStudent& zS); //拷贝构造函数 ~CStudent(); //析构函数 void SetStudent(); //学生对象（学号姓名成绩）设置操作函数 void ShowStudent() const; //学生对象显示操作函数 void CompareScore(const CStudent &zS) const; //学生成绩比较函数 1 void CompareScore(const CStudent *pS) const; //学生成绩比较函数 2

6[D&SJX★] 设计、编写和调试面向对象 C++程序，支持和实现基于三个平面坐标点的三角形面积计算功能。要求程序设计一个平面坐标点类和一个三角形类：平面坐标点类拥有两个整型数据成员分别表示其横纵座标，并拥有构造函数（支持缺省情况下的点对象初始化，即无参情况下横纵座标均初始化为 0，一个参数情况下认为其用于设置横坐标而纵座标初始化为 0，两个参数情况下对应分别设置横、纵座标）、析构函数、设置函数（返回值类型为 void、两个参数类型均为整型）、计算点间距离的函数（返回值类型为 double、参数类型均为点类对象）及横坐标值提取函数（无参数、返回值类型为整型）和纵坐标值提取函数（无参数、返回值类型为整型）等至少五个成员函数；三角形类拥有一个长度为 3 的平面坐标点类对象数组作为数据成员来表示三角形三个顶点，并拥有构造函数、析构函数、设置函数和面积计算

等共计至少四个成员函数，其中构造函数和设置函数的参数序列要求为三个形如 `const CPoint &zP1, ...` 的点类常量对象引用，面积计算函数原型要求为无参和 `double` 返回值类型。平面坐标点类和三角形类规范化设计规格要求参表所示。要求主函数总控实现一个三角形对象的创建，计算其面积和输出；重新设置三角形（三个顶点）后再次计算其面积和输出（注意是通过调用三角形类及平面坐标点类的对应成员函数来实现而非主函数自身直接实现）。

class CPoint
int mX, mY; //横、纵坐标
CPoint(int zX=0, int zY=0); //构造函数
~CPoint(); //析构函数
void SetPoint(int zX, int zY); //设置操作函数
int GetX() const; //横坐标提取操作函数
int GetY() const; //纵坐标提取操作函数
double Distance(const CPoint& zP) const; //点间距离计算函数

class CTriangle
CPoint mVertex[3]; //三角形顶点
CTriangle(const CPoint &zP1=CPoint(0,0), const CPoint &zP2=CPoint(0,1), const CPoint &zP3=CPoint(1,0)); //构造函数
~CTriangle(); //析构函数
void SetTriangle(const CPoint &zP1, const CPoint &zP2, const CPoint &zP3); //设置操作函数
double Area() const; //面积计算函数

7[XS_C★] 设计、编写和调试面向对象 C++ 程序，支持和实现学生基本信息及其成绩的录入、输出及比较等管理功能。基本要求同作业题[XS_B]，但要求增加一个全局函数来实现任意两个学生对象成绩（函数原型要求无返回值、两个形如 `const CStudent &zS1, ...` 的学生类常量引用的参数对象，且该函数定义为学生类的友元函数）的比较并通过主函数调用和测试其功能实现情况。

8[DZ_A] 设计、编写和调试面向对象 C++ 程序，支持和实现整数堆栈的入栈和出栈操作功能。

9[FS_C★] 设计、编写和调试面向对象 C++ 程序，支持和实现任意两个复数的加、减、乘、除运算。基本要求同作业题[FS_B]，但要求复数的加、减、乘、除运算分别通过四个全局函数而非成员函数方式实现（均定义为复数类的友元函数），其函数原型要求返回值类型为复数类，参数序列为两个形如 `const CComplex& zC1, ...` 的复数类常量引用的参数对象。

10[FS_D★] 设计、编写和调试面向对象 C++ 程序，支持和实现任意两个复数的加、减、乘、除运算。基本要求同作业题[FS_B]，但要求复数的加、减、乘、除运算均通过运算符重载方式实现，其中加、减采用成员运算符方式，而乘、除采用友元运算符方式。另外，还应建立取模函数支持复数到实数的转换（类类型转换函数实现方式）。

11[YLS★] 设计、编写和调试面向对象 C++ 程序，采用运算符重载机制实现任意两个有理数（即分数，由分子和分母构成）的加、减、乘、除及输入/输出操作，要求运算结果为经过化简的结果。

12[ZJ★] 设计、编写和调试面向对象 C++ 程序，采用运算符重载机制实现两个矩阵的加、减、乘及输入/输出操作，要求通过（）运算符重载来提取或设置矩阵元素值（函数原型：返回值为 double&，两个整型参数表示矩阵元素行、列值）。

13[Y&YZT★] 运用面向对象方法设计圆类和圆柱体类并建立继承关系，并为有关类设计构造函数、析构函数、成员数据设置/提取操作、面积/体积计算函数。在主函数中，分别创建一个圆类对象和圆柱体类对象，调用和执行相关成员函数，观察并分析程序运行结果。

14[PMTX★] 运用面向对象方法设计点类和平面图形类（含点类成员），并由后者派生设计圆类（含圆心、半径）、正方形类（含左上角顶点、边长）与矩形类（含左上角顶点、长和宽）、三角形类（含三个顶点）；同时为有关类设计构造函数、成员数据设置/提取操作函数、顶点显示函数及周长与面积计算函数，其中顶点显示函数及周长与面积计算函数在平面图形类中应定义为纯虚函数。在主函数中，分别创建各类对象，调用和执行相关成员函数，观察并分析程序运行结果。

15[TYBDX★] 设计、编写和调试面向对象 C++ 程序，实现任意两个同类数据比大小的通用函数（模板），涵盖普通数值型数据之间的比较、两个字符串之间的比较（按字典序）、两个复数之间的比较（按模大小）等，并通过主函数加以调用和测试。

16[DZ_B] 设计、编写和调试面向对象 C++ 程序，实现任意同一类型的数据堆栈的压栈操作和出栈操作，涵盖普通数值型数据堆栈、字符串堆栈、复数堆栈等，并通过主函数加以调用和测试。

17[YCCL★] 阅读分析下面的字符串异常处理程序：

```
#include <iostream>
#include <string.h>
#include <Windows.h>

using namespace std;

class CString
{
    char *m_pStr;
    int mL;
    static int mMaxL;
public:
    CString(char*, int);
    char & operator[](int zIndex) const;
    int Length() const;
    class CExceptionOfRange
    {
    public:
        CExceptionOfRange(int zR, int zLimit): mIndex(zR), mLimit(zLimit) {}
        int mIndex;
        int mLimit;
    };

    class CExceptionOfSize
    {};

    class CExceptionOfM
    {};
};

int CString::mMaxL = 25;

CString::CString(char *zStr, int zIndex)
{
    if (zIndex < 0 || zIndex > mMaxL)
        throw CExceptionOfSize();
    m_pStr = new char[zIndex];
    strncpy(m_pStr, zStr, zIndex);
    mL = zIndex;
}
```

```

char& CString::operator[](int zIndex) const
{
    if (zIndex >= 0 && zIndex < mL)
    {
        return m_pStr[zIndex];
    }
    else throw CExceptionOfRange(zIndex, mL);
}

int CString::Length() const
{
    return mL;
}

void DisplayString(CString& zStr)
{
    for (int i = 0; i < zStr.Length(); i++)
    {
        if (zStr[i] != '\0')
            cout<<zStr[i];
        else break;
    }
    cout<<endl;
}

void ExceptionCheck()
{
    try
    {
        CString zStr("abcdefghijklmnopqrst", 12);           //第一问有效定义，第二/三/四问应从行首注释本行
        CString zStr("abcdefghijklmnopqrst", 10);           //第二问有效定义，第一/三/四问应从行首注释本行
        CString zStr("abcdefghijklmnopqrstuvwxy*", 26);      //第三问有效定义，第一/二/四问应从行首注释本行
        CString zStr("abcdefghijklmnopqrst", 21);           //第四问有效定义，第一/二/三问应从行首注释本行
        DisplayString(zStr);
        cout<<"[由于上面的语句执行可能发生异常，而这里调用[]运算符重载函数亦可能发生异常，";
        cout<<"所以本行代码会在异常发生时失效！] zStr[11] = " <<zStr[11]<<endl;
    }
    catch (CString::CExceptionOfRange zEOR)
    {
        cerr<<"[发生异常] 字符串下标"<<zEOR.mIndex<<"超出界限[0,"<<zEOR.mLimit<<"]"<<endl;
    }
    catch (CString::CExceptionOfSize)
    {
        cerr<<"[发生异常] 字符串长度非法!"<<endl;
    }
    catch (CString::CExceptionOfM)
    {
        cout<<endl<<"[发生异常] M is found here!"<<endl;
    }
    cout<<"程序将从这里继续执行！"<<endl<<endl;
}

// 定义控制台应用程序的入口点
void main()
{
    ExceptionCheck();
    cout<<"返回主函数继续执行！"<<endl<<endl;
    system("pause");
}

```

- (1) 给出程序运行结果;
- (2) 如果函数 ExceptionCheck()的构造 CString 对象的定义语句改为:
 CString zStr("abcdefghijklmnopqrst", 10);
 则程序运行结果又是什么?
- (3) 如果函数 ExceptionCheck()的构造 CString 对象的定义语句改为:
 CString zStr("abcdefghijklmnopqrstuvwxy*", 26);

则程序运行结果又是什么？

(4)试修改 CString 类的[]成员运算符重载函数,即如果在 CString 对象中检测到一个字符'm',则抛掷一个 CExceptionOfM 类型的异常,否则正常返回。同时,把函数 ExceptionCheck()的构造CString 对象的定义语句改为: CString zStr("abcdefghijklmnpqrst", 21);
然后编译调试并给出运行结果。

18[FXSFK] 编写程序初始化整型数组 arrayData[] = {1, 2, 3, 4, 5, 6, 7, ...}, 提示用户输入一个整数, 然后调用泛型算法库函数 find 给出用户输入数据在该数组中的位置。

19[BZRQL] 在标准 C++类库中, 栈类 stack 的成员函数 push()用于向栈顶添加元素、pop()用于从非空栈的栈顶删除一个元素、empty()用于判断栈是否为空、top()返回非空栈的栈顶元素、size()返回栈中元素个数。试构造一个整数栈, 调用有关函数来体会标准容器的使用方法。

20[ZJW] 修改课堂未使用 MFC 类库的 Windows 应用程序 HelloChina 例程为 HelloXXX(XXX 代表你自己的姓名首字母), 也即向中国问好的信息显示改为向你自己(姓名)问好。上机调试运行相应程序, 理解和体会 Windows 应用程序的组成和运行机理。

21[ZJMFC] 修改课堂使用 MFC 类库的 Windows 应用程序 HelloChina 例程为 HelloXXX(XXX 代表你自己的姓名首字母), 也即向中国问好的信息显示改为向你自己(姓名)问好。上机调试运行相应程序, 理解和体会 MFC 基类库在 Windows 应用程序所起的作用及基于 MFC 的 Windows 应用程序编程要领。

22[ZJMFCAW] 利用 VC++开发环境创建一个最简 MFC AppWizard [exe]项目程序, 其仅需最简选项支持(即单文档、文档/查看体系结构支持[V]、资源语言[中文]、不含数据库、不需要复合文档支持及任何其它支持, MFC 标准风格、需要生成源文件备注、MFC 库共享 DLL 使用方式)。分析该项目类型程序的组成、各部分代码功能及运行机理, 总结 VC++此类项目开发过程、关键问题及解决方案。