

**C++程序设计实验报告**



实验题目 运算符重载

学 号 201806061108

姓 名 胡皓睿

班 级 计实1801

提交日期 　2019/5/6

1. 实验目的：

掌握运算符重载的相关内容。

1. 实验内容：

1. 书本p327页习题2。定义一个复数类Complex，重载运算符+，-，\*，/，使之能用于复数的加、减、乘、除。运算符重载函数作为Complex类的成员函数。在主函数定义两个复数，键入实部和虚部，并测试求两个复数之和、差、积和商。

2. 书本p327页习题3。定义一个复数类Complex，重载运算符“+”，使之能用于复数的加法运算。参加运算的两个运算量可以都是类对象，也可以其中有一个是整数，顺序任意。例如：c1+c2，i+c1，c1+i均合法（设i为整数，c1，c2位复数）。编程序，分别求两个复数之和、整数和复数之和

3. 完成书本p327习题4-5。有两个矩阵a和b，均为2行3列。求两个矩阵之和。重载运算符“+”，使之能用于矩阵相加（如c=a+b），

重载流插入运算符“<<”和流提取运算符“>>”，使之能用于该矩阵的输入和输出

4. 完成书本p327习题6。请编写程序，处理一个复数与一个double数相加的运算，结果存放在一个double型的变量d1中，输出d1的值，再以复数形式输出此值。定义Complex（复数）类，在成员函数中包含重载类型转换运算符：

Operator double（）{return real；}

5. 完成书本p327习题7。定义一个Teacher（教师）类和一个Student（学生）类，二者有一部分数据成员是相同的，例如num（号码），name（姓名），sex（性别）。编写程序，将一个Student对象（学生）转换为Teacher（教师）类，只将以上3个相同的数据成员移植过去。可以设想：一位学生大学毕业了，留校担任教师，他原有的部分数据对现在的教师身份来说仍然是有用的，应当保留并成为其教师数据的一部分。

三、实验过程

1. 书本p327页习题2。定义一个复数类Complex，重载运算符+，-，\*，/，使之能用于复数的加、减、乘、除。运算符重载函数作为Complex类的成员函数。在主函数定义两个复数，键入实部和虚部，并测试求两个复数之和、差、积和商。

（1）程序代码

#include <iostream>

using namespace std;

class Complex

{

private:

float a, b;

public:

Complex(float x, float y);

Complex(void);

Complex operator +(Complex& c2);

Complex operator -(Complex& c2);

Complex operator \*(Complex& c2);

Complex operator /(Complex& c2);

void show();

};

Complex::Complex(float x, float y)

{

a = x;

b = y;

}

Complex::Complex(void)

{

;

}

Complex Complex::operator +(Complex& c2)

{

Complex c;

c.a = a + c2.a;

c.b = b + c2.b;

return c;

}

Complex Complex::operator -(Complex& c2)

{

Complex c;

c.a = a - c2.a;

c.b = b - c2.b;

return c;

}

Complex Complex::operator \*(Complex& c2)

{

Complex c;

c.a = a \* c2.a - b \* c2.b;

c.b = b \* c2.a + a \* c2.b;

return c;

}

Complex Complex::operator /(Complex& c2)

{

Complex c;

c.a = (a \* c2.a + b \* c2.b) / (b \* b + c2.b \* c2.b);

c.b = (b \* c2.a - a \* c2.b) / (b \* b + c2.b \* c2.b);

return c;

}

void Complex::show()

{

cout << '(' << a;

if (b > 0) cout << '+';

cout << b << "i)\n";

}

int main()

{

float n[4];

cin >> n[0] >> n[1] >> n[2] >> n[3];

Complex a(n[0], n[1]), b(n[2], n[3]);

Complex c = a + b, d = a - b, e = a \* b, f = a / b;

c.show();

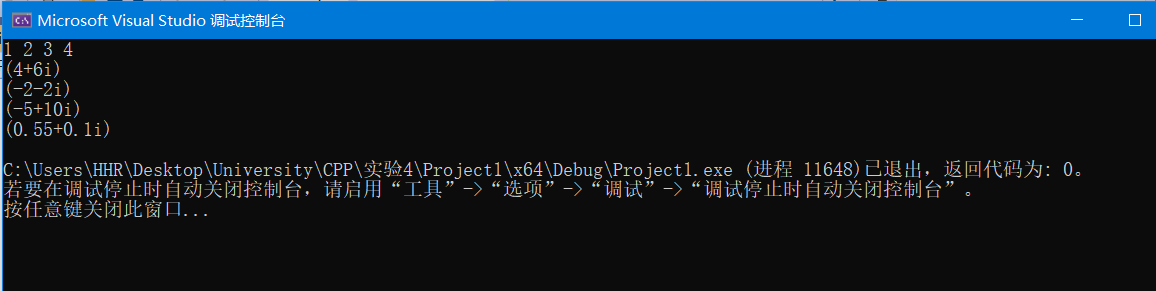
d.show();

e.show();

f.show();

return 0;

}（2）运行结果



（3）结果分析

考查了运算符重载的简单运用。

2. 书本p327页习题3。定义一个复数类Complex，重载运算符“+”，使之能用于复数的加法运算。参加运算的两个运算量可以都是类对象，也可以其中有一个是整数，顺序任意。例如：c1+c2，i+c1，c1+i均合法（设i为整数，c1，c2位复数）。编程序，分别求两个复数之和、整数和复数之和

（1）程序代码

#include <iostream>

using namespace std;

class Complex

{

private:

float a, b;

public:

Complex(float x, float y);

Complex(void);

Complex operator +(Complex& c2);

Complex operator +(int n);

friend Complex operator +(int n, Complex& c2);

void show();

};

Complex::Complex(float x, float y)

{

a = x;

b = y;

}

Complex::Complex(void)

{

;

}

Complex Complex::operator +(Complex& c2)

{

Complex c;

c.a = a + c2.a;

c.b = b + c2.b;

return c;

}

Complex Complex::operator +(int n)

{

Complex c;

c.a = a + n;

c.b = b;

return c;

}

Complex operator +(int n, Complex& c2)

{

Complex c;

c.a = n + c2.a;

c.b = c2.b;

return c;

}

void Complex::show()

{

cout << '(' << a;

if (b > 0) cout << '+';

cout << b << "i)\n";

}

int main()

{

float n[4], a;

cin >> n[0] >> n[1] >> n[2] >> n[3] >> a;

Complex c1(n[0], n[1]), c2(n[2], n[3]);

Complex c3 = c1 + c2, c4 = c1 + a, c5 = a + c1;

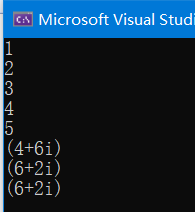
c3.show();

c4.show();

c5.show();

return 0;

}（2）运行结果



（3）结果分析

对不同类型数据进行运算的重载。

3. 完成书本p327习题4-5。有两个矩阵a和b，均为2行3列。求两个矩阵之和。重载运算符“+”，使之能用于矩阵相加（如c=a+b），

重载流插入运算符“<<”和流提取运算符“>>”，使之能用于该矩阵的输入和输出

（1）程序代码

#include <iostream>

#include <iomanip>

using namespace std;

class matrix

{

private:

float num[2][3];

public:

matrix();

void show();

friend matrix operator +(matrix&, matrix&);

friend ostream & operator << (ostream&, matrix&);

friend istream& operator >>(istream&, matrix&);

};

istream& operator >>(istream& input, matrix& c)

{

for (int x = 0; x < 2; x++)

for (int y = 0; y < 3; y++)

input >> c.num[x][y];

return input;

}

ostream& operator << (ostream& output, matrix& c)

{

for (int x = 0; x < 2; x++)

{

for (int y = 0; y < 3; y++)

{

output << setw(6) << c.num[x][y];

}

cout << '\n';

}

return output;

}

matrix operator +(matrix& a, matrix& b)

{

matrix c;

for (int x = 0; x < 2; x++)

for (int y = 0; y < 3; y++)

c.num[x][y] = a.num[x][y] + b.num[x][y];

return c;

}

void matrix::show()

{

}

matrix::matrix()

{}

int main()

{

matrix m1, m2;

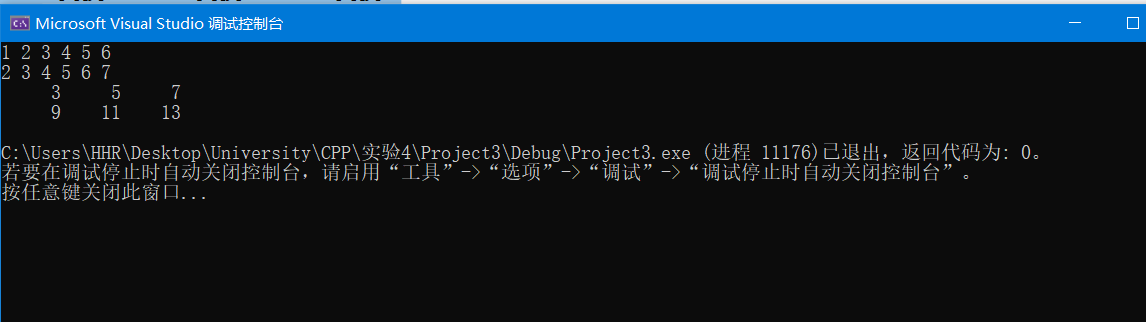
cin >> m1 >> m2;

matrix m3 = m1 + m2;

cout << m3;

return 0;

}（2）运行结果



（3）结果分析

考察了运算符重载的有关知识

4. 完成书本p327习题6。请编写程序，处理一个复数与一个double数相加的运算，结果存放在一个double型的变量d1中，输出d1的值，再以复数形式输出此值。定义Complex（复数）类，在成员函数中包含重载类型转换运算符：

Operator double（）{return real；}

（1）程序代码

#include <iostream>

using namespace std;

class Complex

{

public:

Complex()

{

real = 0;

image = 0;

}

Complex(double r, double i)

{

real = r;

image = i;

}

operator double()

{

return real;

}

public:

double real, image;

};

int main()

{

int a, b, c, d;

cin >> a >> b >> c >> d;

Complex c1(a, b), c2(c, d), c3;

double dl;

dl = 2.5 + c1;

cout << dl << endl;

return 0;

}（2）运行结果



5. 完成书本p327习题7。定义一个Teacher（教师）类和一个Student（学生）类，二者有一部分数据成员是相同的，例如num（号码），name（姓名），sex（性别）。编写程序，将一个Student对象（学生）转换为Teacher（教师）类，只将以上3个相同的数据成员移植过去。可以设想：一位学生大学毕业了，留校担任教师，他原有的部分数据对现在的教师身份来说仍然是有用的，应当保留并成为其教师数据的一部分。

（1）程序代码

#include<iostream>

#include<cstring>

using namespace std;

class Teacher

{

private:

string num, name;

char sex, degree;

public:

Teacher(string nu, string na, char se, char de = '0') :num(nu), name(na), sex(se), degree(de) {};

void display();

};

class Student

{

private:

string num, name;

char sex;

int score;

public:

Student(string nu, string na, char se, int sc = 0) :num(nu), name(na), sex(se), score(sc) {};

void display();

operator Teacher()

{

Teacher t(num, name, sex);

return t;

}

};

void Teacher::display()

{

cout << num << endl << name << endl << sex << endl << degree << endl;

}

void Student::display()

{

cout << num << endl << name << endl << sex << endl << score << endl;

}

int main()

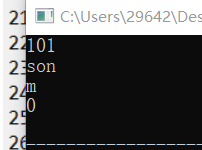
{

Student s("101", "son", 'm');

Teacher(s).display();

}

（2）运行结果



（3）结果分析

对不同类型数据进行转换。

四、小结与收获

这次实验主要是基于对象的程序设计的一些相关知识。以后用的最多的其实就是基于对象的程序设计，所以这个实验是十分重要的。在这个实验中我对运算符重载中的单目双目运算符，还有输入输出等各种重载都有了更深刻的理解，这让我更好地掌握了解决复杂工程问题的程序设计语言基础。而这次实验中各种不同的重载方式对我日后处理复杂代码有很大帮助是我将工程基础和专业知识用于求解软件领域复杂工程问题的基础。