程设十四次作业

1. 重载运算符“+”，参加运算的两个运算量可以都是类对象，也可以有整数，顺序任意

```
#include <iostream>
using namespace std;
class Complex
{
public:
    Complex() { real = 0; imag = 0; }
    Complex(double r, double i) { real = r; imag = i; }
    Complex operator+(Complex& c2);
    Complex operator+(int& i);
    friend Complex operator+(int&, Complex&);
    void display();
private:
    double real;
    double imag;
};

Complex Complex::operator+(Complex& c)
{
    return Complex(real + c.real, imag + c.imag);
}

Complex Complex::operator+(int& i)
{
    return Complex(real + i, imag);
}

void Complex::display()
{
    cout << "(" << real << "," << imag << "i)" << endl;
}

Complex operator+(int& i, Complex& c)
{
    return Complex(i + c.real, c.imag);
}

int main()
{
    Complex c1(3, 4), c2(5, -10), c3;
    int i = 5;
    c3 = c1 + c2;
    cout << "c1+c2=";
    c3.display();
    c3 = i + c1;
    cout << "i+c1=";
    c3.display();
    c3 = c1 + i;
    cout << "c1+i=";
```

```
    c3.display();
    return 0;
}
```

运行结果如下：

```
c1+c2=(8,-6i)
i+c1=(8,4i)
c1+i=(8,4i)
```

2. 有两个矩阵a，b，均为2行3列，求矩阵和。重载运算符“+”

```
#include <iostream>
using namespace std;
class Matrix   //定义Matrix类
{
public:
    Matrix();   //默认构造函数
    friend Matrix operator+(Matrix&, Matrix&);  //重载运算符“+”
    void input();   //输入数据函数
    void display(); //输出数据函数
private:
    int mat[2][3];
};

Matrix::Matrix()  //定义构造函数
{
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 3; j++)
            mat[i][j] = 0;
}

Matrix operator+(Matrix& a, Matrix& b)  //定义重载运算符“+”
{
    Matrix c;
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 3; j++)
        {
            c.mat[i][j] = a.mat[i][j] + b.mat[i][j];
        }
    return c;
}
void Matrix::input()   //定义输入数据函数
{
    cout << "input value of matrix:" << endl;
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 3; j++)
            cin >> mat[i][j];
}

void Matrix::display()                                  //定义输出数据函数
{
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 3; j++)
```

```
        {
            cout << mat[i][j] << " ";
        }
        cout << endl;
    }
}

int main()
{
    Matrix a, b, c;
    a.input();
    cout << endl << "Matrix a:" << endl;
    a.display();
    b.input();
    cout << endl << "Matrix b:" << endl;
    b.display();
    c = a + b;                                          //用重载运算符“+”实现两个矩阵相加
    cout << endl << "Matrix c = Matrix a + Matrix b :" << endl;
    c.display();
    return 0;
}
```

运行结果如下：

> Martrix a:
>
> 1 2 3
> 4 5 6
>
> Matrix b:
> 7 8 9
> 10 11 12
>
> Matrix c = Matrix a + Matrix b :
> 8 10 12
> 14 16 18

3. 基于10.4 重载插入运算符<<和流提取运算符>>

```
#include <iostream>
using namespace std;
class Matrix
{
public:
    Matrix();
    friend Matrix operator+(Matrix&, Matrix&);
    friend ostream& operator<<(ostream&, Matrix&);
    friend istream& operator>>(istream&, Matrix&);
private:
    int mat[2][3];
};

Matrix::Matrix()
{
    for (int i = 0; i < 2; i++)
```

```
        for (int j = 0; j < 3; j++)
            mat[i][j] = 0;
}

Matrix operator+(Matrix& a, Matrix& b)
{
    Matrix c;
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 3; j++)
        {
            c.mat[i][j] = a.mat[i][j] + b.mat[i][j];
        }
    return c;
}

istream& operator>>(istream& in, Matrix& m)
{
    cout << "input value of matrix:" << endl;
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 3; j++)
            in >> m.mat[i][j];
    return in;
}

ostream& operator<<(ostream& out, Matrix& m)
{
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            out << m.mat[i][j] << " ";
        }
        out << endl;
    }
    return out;
}

int main()
{
    Matrix a, b, c;
    cin >> a;
    cout << endl << "Matrix a:" << endl << a << endl;
    cin >> b;
    cout << endl << "Matrix b:" << endl << b << endl;
    c = a + b;
    cout << endl << "Matrix c = Matrix a + Matrix b :" << endl << c <<
endl;
    return 0;
}
```

运行结果与上题相同。

4. 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，再以复数形式输出。成员函数中包含重载类型转换运算符。

```cpp
#include <iostream>
using namespace std;
class Complex
{
public:
    Complex() { real = 0; imag = 0; }
    Complex(double r) { real = r; imag = 0; }
    Complex(double r, double i) { real = r; imag = i; }
    operator double() { return real; }
    void display();
private:
    double real;
    double imag;
};

void Complex::display()
{
    cout << "(" << real << ", " << imag << ")" << endl;
}

int main()
{
    Complex c1(3, 4), c2;
    double d1;
    d1 = 2.5 + c1;
    cout << "d1=" << d1 << endl;
    c2 = Complex(d1);
    cout << "c2=";
    c2.display();
    return 0;
}
```

运行结果如下：

```
d1=5.5
c2=(5.5, 0)
```