

# 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

```
1  #include <iostream>
2  using namespace std;
3  class Complex
4  {
5  public:
6      Complex() { real = 0; imag = 0; }
7      Complex(double r, double i) { real = r; imag = i; }
8      Complex operator+(Complex& c2);
9      Complex operator+(int& i);
10     friend Complex operator+(int&, Complex&);
11     void display();
12 private:
13     double real;
14     double imag;
15 };
16
17 Complex Complex::operator+(Complex& c)
18 {
19     return Complex(real + c.real, imag + c.imag);
20 }
21
22 Complex Complex::operator+(int& i)
23 {
24     return Complex(real + i, imag);
25 }
26
27 void Complex::display()
28 {
29     cout << "(" << real << "," << imag << "i)" << endl;
30 }
31
32 Complex operator+(int& i, Complex& c)
33 {
34     return Complex(i + c.real, c.imag);
35 }
36
37 int main()
38 {
39     Complex c1(3, 4), c2(5, -10), c3;
40     int i = 5;
41     c3 = c1 + c2;
42     cout << "c1+c2=";
43     c3.display();
44     c3 = i + c1;
45     cout << "i+c1=";
46     c3.display();
47     c3 = c1 + i;
48     cout << "c1+i=";
```

```

49     c3.display();
50     return 0;
51 }

```

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“+”

```

1  #include <iostream>
2  using namespace std;
3  class Matrix    //定义Matrix类
4  {
5  public:
6      Matrix();    //默认构造函数
7      friend Matrix operator+(Matrix&, Matrix&);    //重载运算符“+”
8      void input();    //输入数据函数
9      void display(); //输出数据函数
10 private:
11     int mat[2][3];
12 };
13
14 Matrix::Matrix()    //定义构造函数
15 {
16     for (int i = 0; i < 2; i++)
17         for (int j = 0; j < 3; j++)
18             mat[i][j] = 0;
19 }
20
21 Matrix operator+(Matrix& a, Matrix& b)    //定义重载运算符“+”
22 {
23     Matrix c;
24     for (int i = 0; i < 2; i++)
25         for (int j = 0; j < 3; j++)
26         {
27             c.mat[i][j] = a.mat[i][j] + b.mat[i][j];
28         }
29     return c;
30 }
31 void Matrix::input()    //定义输入数据函数
32 {
33     cout << "input value of matrix:" << endl;
34     for (int i = 0; i < 2; i++)
35         for (int j = 0; j < 3; j++)
36             cin >> mat[i][j];
37 }
38
39 void Matrix::display()    //定义输出数据函数
40 {
41     for (int i = 0; i < 2; i++)
42     {
43         for (int j = 0; j < 3; j++)

```

```

44     {
45         cout << mat[i][j] << " ";
46     }
47     cout << endl;
48 }
49 }
50
51 int main()
52 {
53     Matrix a, b, c;
54     a.input();
55     cout << endl << "Matrix a:" << endl;
56     a.display();
57     b.input();
58     cout << endl << "Matrix b:" << endl;
59     b.display();
60     c = a + b; //用重载运算符“+”实
    现两个矩阵相加
61     cout << endl << "Matrix c = Matrix a + Matrix b :" << endl;
62     c.display();
63     return 0;
64 }

```

运行结果如下：

```

Matrix 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>>

```

1  #include <iostream>
2  using namespace std;
3  class Matrix
4  {
5  public:
6      Matrix();
7      friend Matrix operator+(Matrix&, Matrix&);
8      friend ostream& operator<<(ostream&, Matrix&);
9      friend istream& operator>>(istream&, Matrix&);
10 private:
11     int mat[2][3];
12 };
13
14 Matrix::Matrix()
15 {
16     for (int i = 0; i < 2; i++)

```

```

17         for (int j = 0; j < 3; j++)
18             mat[i][j] = 0;
19     }
20
21     Matrix operator+(Matrix& a, Matrix& b)
22     {
23         Matrix c;
24         for (int i = 0; i < 2; i++)
25             for (int j = 0; j < 3; j++)
26             {
27                 c.mat[i][j] = a.mat[i][j] + b.mat[i][j];
28             }
29         return c;
30     }
31
32     istream& operator>>(istream& in, Matrix& m)
33     {
34         cout << "input value of matrix:" << endl;
35         for (int i = 0; i < 2; i++)
36             for (int j = 0; j < 3; j++)
37                 in >> m.mat[i][j];
38         return in;
39     }
40
41     ostream& operator<<(ostream& out, Matrix& m)
42     {
43         for (int i = 0; i < 2; i++)
44         {
45             for (int j = 0; j < 3; j++)
46             {
47                 out << m.mat[i][j] << " ";
48             }
49             out << endl;
50         }
51         return out;
52     }
53
54     int main()
55     {
56         Matrix a, b, c;
57         cin >> a;
58         cout << endl << "Matrix a:" << endl << a << endl;
59         cin >> b;
60         cout << endl << "Matrix b:" << endl << b << endl;
61         c = a + b;
62         cout << endl << "Matrix c = Matrix a + Matrix b :" << endl << c <<
endl;
63         return 0;
64     }

```

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。

```

1  #include <iostream>
2  using namespace std;
3  class Complex
4  {
5  public:
6      Complex() { real = 0; imag = 0; }
7      Complex(double r) { real = r; imag = 0; }
8      Complex(double r, double i) { real = r; imag = i; }
9      operator double() { return real; }
10     void display();
11 private:
12     double real;
13     double imag;
14 };
15
16 void Complex::display()
17 {
18     cout << "(" << real << ", " << imag << ")" << endl;
19 }
20
21 int main()
22 {
23     Complex c1(3, 4), c2;
24     double d1;
25     d1 = 2.5 + c1;
26     cout << "d1=" << d1 << endl;
27     c2 = Complex(d1);
28     cout << "c2=";
29     c2.display();
30     return 0;
31 }

```

运行结果如下：

```

d1=5.5
c2=(5.5, 0)

```