**河南开封科技传媒学院**

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

**实验报告**

**2021- 2022学年第 一 学期**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名： | 刘东 | | 学号： | 2136101045 | 专业： | 软件工程 | |
| 所在学院： | | 理工学院 | | | 指导老师： | | 李莉杰 |
| 实验时间： | | 2021.12.15 | | | 实验地点： | | 6503 |
| 课程名称： | |  | | | | | |
| 实验题目： | | 运算符重载 | | | | | |
| 实验目的：  掌握运算符重载的用法 | | | | | | | |
| 实验内容与步骤： | | | | | | | |
| 实验或测试数据记录： | | | | | | | |
| 附录（附加结果或代码）：  #include <iostream>  using namespace std;  class matrix  {  public:  matrix();  void input();  friend matrix operator+(matrix &,matrix &);  void show();  private:  int m[2][3];  };  matrix::matrix()  {  for(int i=0;i<2;i++)  {  for(int j=0;j<3;j++)  {  m[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.m[i][j]=a.m[i][j]+b.m[i][j];  }  }  return c;  }  void matrix::input()  {  for(int i=0;i<2;i++)  {  for(int j=0;j<3;j++)  {  cin>>m[i][j];  }  }  }  void matrix::show()  {  for(int i=0;i<2;i++)  {  for(int j=0;j<3;j++)  {  if(j<2)  {  cout<<m[i][j]<<" ";  }  else  {  cout<<m[i][j]<<endl;  }  }  }  }  int main()  {  matrix a,b,c;  a.input();  b.input();  c=a+b;  c.show();  return 0;  }  #include<iostream>  using namespace std;  class Complex  {  public:  Complex() { real = 1; imag = 0; }  Complex(double r,double i):real(r),imag(i){}  void display();  friend Complex operator + (Complex& c1, Complex& c2);  friend Complex operator - (Complex& c1, Complex& c2);  friend Complex operator \* (Complex& c1, Complex& c2);  friend Complex operator / (Complex& c1, Complex& c2);  private:  double real;  double imag;  };  void Complex::display()  {  cout << "(" << real << "," << imag << "i)" << endl;  }  Complex operator + (Complex& c1, Complex& c2)  {  return Complex(c1.real + c2.real, c1.imag + c2.imag);  }  Complex operator - (Complex& c1, Complex& c2)  {  return Complex(c1.real - c2.real, c1.imag - c2.imag);  }  Complex operator \* (Complex& c1, Complex& c2)  {  Complex c3;  c3.real = c1.real \* c2.real - c1.imag \* c2.imag;  c3.imag = c1.imag \* c2.real + c1.real \* c2.imag;  return c3;  }  Complex operator / (Complex& c1, Complex& c2)  {  Complex c3;  c3.real = (c1.real \* c2.real + c1.imag \* c2.imag)/(c2.real\*c2.real+c2.imag\*c2.imag);  c3.imag = (c1.imag \* c2.real - c1.real \* c2.imag)/(c2.real \* c2.real + c2.imag \* c2.imag);  return c3;  }  void cal\_Comp(Complex& c1, Complex& c2, int c)  {  Complex c3;  switch (c)  {  case 1:  cout << "c1+c2=";  c3 = c1 + c2;  c3.display();  break;  case 2:  cout << "c1-c2=";  c3 = c1 - c2;  c3.display();  break;  case 3:  cout << "c1\*c2=";  c3 = c1 \* c2;  c3.display();  break;  default:  cout << "c1/c2=";  c3 = c1 / c2;  c3.display();  break;  }  }  int main()  {  Complex c1(2.1, 3.2), c2(0.1, 0.5),c3,c4,c5,c6;  cout << "c1:";  c1.display();  cout << "c2:";  c2.display();  cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;  cout << "选择计算类型（选择序号）" << endl;  cout << "\t1、加法" << endl;  cout << "\t2、减法" << endl;  cout << "\t3、乘法" << endl;  cout << "\t4、除法" << endl;  cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;  int i;  cin >> i;  cal\_Comp(c1, c2, i);  system("pause");  return 0;  } | | | | | | | |
| 实验思考（实验过程中出现的问题以及解决方法）： | | | | | | | |