#include <iostream>

#include <math.h>

#include <conio.h>

#include <vector>

using namespace std;

class Complex

{

private:

double real\_chast, mnimaia\_chast, sum;

public:

Complex() //конструктор по умолчанию

{

mnimaia\_chast = 0;

real\_chast = 0;

sum = 0;

};

Complex(double real\_chast)

{

this->real\_chast = real\_chast;

mnimaia\_chast = 0;

sum = sqrt(real\_chast \* real\_chast);

};

Complex(double real\_chast, double mnimaia\_chast)

{

this->real\_chast = real\_chast;

this->mnimaia\_chast = mnimaia\_chast;

sum = sqrt(real\_chast \* real\_chast + mnimaia\_chast \* mnimaia\_chast);

};

Complex(Complex& value)

{

real\_chast = value.real\_chast;

mnimaia\_chast = value.mnimaia\_chast;

};

~Complex() {}

bool operator != (Complex& other)

{

return !(this->sum == other.sum);

}

bool operator == (Complex& other)

{

return (this->sum == other.sum);

}

bool operator > (Complex& other)

{

return (this->sum > other.sum);

}

bool operator < (Complex& other)

{

return (this->sum < other.sum);

}

void operator = (Complex& other)

{

this->real\_chast = other.real\_chast;

this->mnimaia\_chast = other.mnimaia\_chast;

this->sum = other.sum;

}

Complex operator + (Complex& other)

{

Complex temp;

temp.real\_chast = real\_chast + other.real\_chast;

temp.mnimaia\_chast = mnimaia\_chast + other.real\_chast;

return temp;

}

Complex operator - (Complex& other)

{

Complex temp;

temp.real\_chast = real\_chast - other.real\_chast;

temp.mnimaia\_chast = mnimaia\_chast - other.real\_chast;

return temp;

}

Complex operator \* (Complex& other)

{

Complex temp;

temp.real\_chast = real\_chast \* other.real\_chast;

temp.mnimaia\_chast = mnimaia\_chast \* other.real\_chast;

return temp;

}

Complex operator / (Complex& other)

{

Complex temp;

double r = other.real\_chast \* other.real\_chast + other.mnimaia\_chast \* other.mnimaia\_chast;

temp.real\_chast = (real\_chast \* other.real\_chast + mnimaia\_chast \* other.mnimaia\_chast) / r;

temp.real\_chast = (mnimaia\_chast \* other.real\_chast - real\_chast \* other.mnimaia\_chast) / r;

return temp;

}

friend ostream& operator<<(ostream&, const Complex&);

friend istream& operator>>(istream&, const Complex&);

};

ostream& operator<<(ostream& out, const Complex& other)

{

out << other.real\_chast;

if (other.mnimaia\_chast > 0)

out << "+";

out << other.mnimaia\_chast << "\*i";

return out;

}

istream& operator>>(istream& in, const Complex& other)

{

in >> other.real\_chast >> other.mnimaia\_chast;

return in;

}

///////////////////////////////////////////////////////////

int main()

{

setlocale(LC\_ALL, "");

cout << endl;

Complex cml\_chislo\_1(7, 2);

Complex cml\_chislo\_2(3, 1);

cout << cml\_chislo\_1 << "; " << cml\_chislo\_2 << endl;

cout << " - " << cml\_chislo\_1 + cml\_chislo\_2 << endl;

cout << " + " << cml\_chislo\_1 - cml\_chislo\_2 << endl;

cout << " \* " << cml\_chislo\_1 \* cml\_chislo\_2 << endl;

cout << " / " << cml\_chislo\_1 / cml\_chislo\_2 << endl;

if (cml\_chislo\_1 != cml\_chislo\_2)cout << "\nЧисла не равны\n";

if (cml\_chislo\_1 < cml\_chislo\_2)cout << "1-е число больше 2-го\n";

if (cml\_chislo\_1 > cml\_chislo\_2)cout << "1-е число меньше 2-го\n";

cml\_chislo\_1 = cml\_chislo\_2;

if (cml\_chislo\_1 == cml\_chislo\_2)cout << "Числа равны\n";

cout << cml\_chislo\_1 << " = " << cml\_chislo\_2 << endl;

return 0;

}

