Карагандинский технический университет имени Абылкаса Сагинова

Кафедра: ИВС

**Лабораторная работа №3**

По дисциплине: «Технология программирования»

Тема:Основа работы с GitHub

**Преподователь:**

В.В.В

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*подпись*) (*дата*)

**Выполнил:**

Ст.гр. ИС-23-2С Лобов М.Ю

Караганда 2024

**Вариант 9.**

Постpоить класс для pаботы с комплексными числами. Класс должен включать соответствующие поля: действительную и мнимую часть числа.

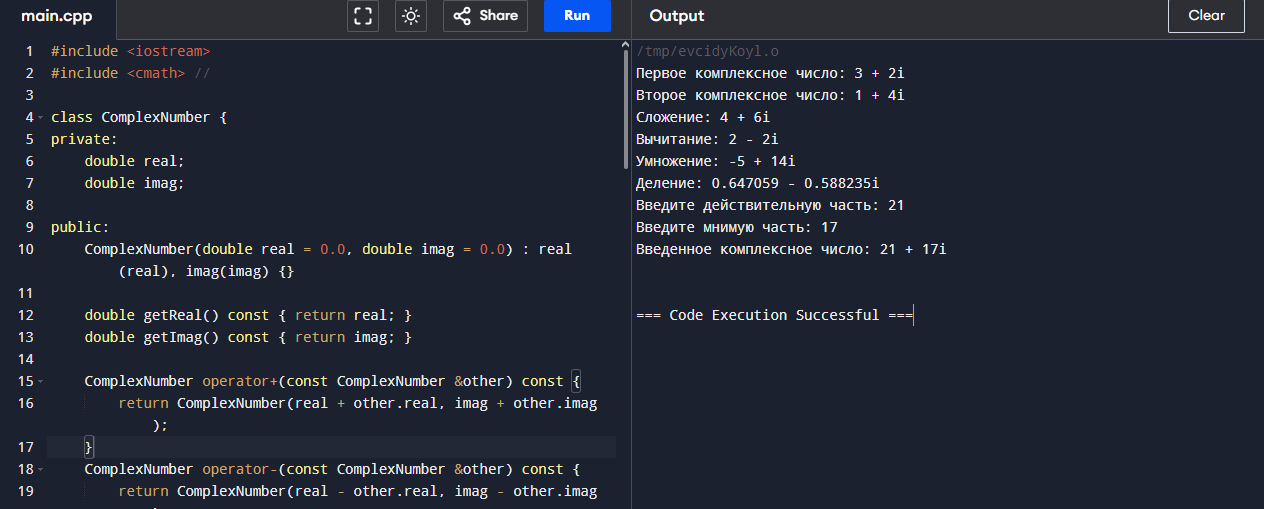
Класс должен обеспечивать пpостейшие функции для pаботы с данными стpуктуpами: сложение, вычитание, умножение, деление, вывод числа в удобной форме и т.д.

Умножение:(a1 + b1i) (a2 + b2i) = (a1a2 - b1b2) + (b1a2 + a1b2)i

Сложение:( a1+b1i) + (a2+b2i) = (a1+a2) + (b1+b2) i.

Вычитание: (a1 + b1i) - (a2 + b2i) = (a1 - a2) + (b1 - b2)i.

Деление: Описание: kompl3_1

 #include <iostream>

#include <cmath> // Для функции sqrt()

class ComplexNumber {

private:

double real; // Действительная часть

double imag; // Мнимая часть

public:

// Конструктор

ComplexNumber(double real = 0.0, double imag = 0.0) : real(real), imag(imag) {}

// Методы для получения действительной и мнимой частей

double getReal() const { return real; }

double getImag() const { return imag; }

// Перегрузка оператора сложения

ComplexNumber operator+(const ComplexNumber &other) const {

return ComplexNumber(real + other.real, imag + other.imag);

}

// Перегрузка оператора вычитания

ComplexNumber operator-(const ComplexNumber &other) const {

return ComplexNumber(real - other.real, imag - other.imag);

}

// Перегрузка оператора умножения

ComplexNumber operator\*(const ComplexNumber &other) const {

double newReal = real \* other.real - imag \* other.imag;

double newImag = imag \* other.real + real \* other.imag;

return ComplexNumber(newReal, newImag);

}

// Перегрузка оператора деления

ComplexNumber operator/(const ComplexNumber &other) const {

double denominator = other.real \* other.real + other.imag \* other.imag;

if (denominator == 0) {

std::cerr << "Ошибка: деление на ноль!" << std::endl;

return ComplexNumber(0, 0);

}

double newReal = (real \* other.real + imag \* other.imag) / denominator;

double newImag = (imag \* other.real - real \* other.imag) / denominator;

return ComplexNumber(newReal, newImag);

}

// Перегрузка оператора вывода

friend std::ostream &operator<<(std::ostream &out, const ComplexNumber &c) {

if (c.imag >= 0)

out << c.real << " + " << c.imag << "i";

else

out << c.real << " - " << -c.imag << "i";

return out;

}

// Перегрузка оператора ввода

friend std::istream &operator>>(std::istream &in, ComplexNumber &c) {

std::cout << "Введите действительную часть: ";

in >> c.real;

std::cout << "Введите мнимую часть: ";

in >> c.imag;

return in;

}

};

// Демонстрационная программа

int main() {

ComplexNumber z1(3, 2); // Создаем первое комплексное число

ComplexNumber z2(1, 4); // Создаем второе комплексное число

std::cout << "Первое комплексное число: " << z1 << std::endl;

std::cout << "Второе комплексное число: " << z2 << std::endl;

// Сложение

ComplexNumber sum = z1 + z2;

std::cout << "Сложение: " << sum << std::endl;

// Вычитание

ComplexNumber difference = z1 - z2;

std::cout << "Вычитание: " << difference << std::endl;

// Умножение

ComplexNumber product = z1 \* z2;

std::cout << "Умножение: " << product << std::endl;

// Деление

ComplexNumber quotient = z1 / z2;

std::cout << "Деление: " << quotient << std::endl;

// Ввод с клавиатуры

ComplexNumber z3;

std::cin >> z3;

std::cout << "Введенное комплексное число: " << z3 << std::endl;

return 0;

}