Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение

высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

по дисциплине

«Программирование в системах информационной безопасности»

на тему

«Понятие класса. Члены класса. Друзья класса. Перегрузка операций»

Вариант №19

Выполнил:

студент группы БСУ1801

Плотников П.С

Проверил:

старший преподаватель кафедры ИБ

Барков В. В.

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

Москва 2021

# **Цель**

Научиться создавать простые классы, создавать конструкторы, перегружать операции, создавать объекты классов и передавать их в функции

# **Задание**

В соответствии с Вашим вариантом разработать класс. Название класса, приватные данные-члены и необходимые публичные функции-члены указаны в таблице ниже.

Для всех вариантов необходимо перегрузить операции + - \* / и операцию приведения типа к типу double.

Составить пользовательскую функцию y, указанную в индивидуальном задании и необходимые вспомогательные функции

|  |  |
| --- | --- |
| Вариант | 19 |
| Название класса | Complex |
| Приватные данные-члены | double re; (действительная часть)  double im; (мнимая часть) |
| Публичные функции-члены | double Re() const;  double Im() const;  double R() const;  double Phi() const; |
| Пользовательские функции | Complex y(const Complex &z); |
| Вспомогательные функции | Complex sin(const Complex &z);  Complex cos(const Complex &z);  Complex ch(const Complex &z);  Complex sh(const Complex &z);  Complex exp(const Complex &z);  Complex pow(const Complex &z1, const Complex &z2); |

|  |
| --- |
| Complex y(const Complex &z); |
| 19. |

Листинг 1 – программный код файла complex.cpp

#include "Header.h"

#include "math.h"

#include<iostream>

Complex::Complex():

m\_re{ sqrt(0)}, m\_im{ atan2(0, 0) }

{}

Complex::Complex(double re):

m\_re{ re }, m\_im{ 0 }

{}

Complex::Complex(double re, double im) :

m\_re{ re }, m\_im{ im }

{}

double Complex::Re() const

{

return m\_re;

}

double Complex::Im() const

{

return m\_im;

}

double Complex::R() const

{

return sqrt(pow(m\_re, 2) \* pow(m\_im,2));

}

double Complex::Phi() const

{

return atan2(m\_im, m\_re);

}

Complex Complex::operator- ()

{

return Complex(-Re(), -Im());

}

Complex& Complex::operator+=(const Complex& right)

{

\*this = Complex(Re() + right.Re(), Im() + right.Im());

return \*this;

}

Complex& Complex::operator-=(const Complex& right)

{

\*this = Complex(Re() - right.Re(), Im() - right.Im());

return \*this;

}

Complex& Complex::operator\*=(const Complex& right)

{

\*this = Complex(Re() \* right.Re(), Im() \* right.Im());

return \*this;

}

Complex& Complex::operator/=(const Complex& right)

{

\*this = Complex(Re() / right.Re(), Im() / right.Im());

return \*this;

}

Complex Complex::operator+ (const Complex& right)

{

Complex res = \*this;

return res += right;

}

Complex Complex::operator- (const Complex& right)

{

Complex res = \*this;

return res -= right;

}

Complex Complex::operator\* (const Complex& right)

{

Complex res = \*this;

return res \*= right;

}

Complex Complex:: operator/ (const Complex& right)

{

Complex res = \*this;

return res /= right;

}

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

{

out << "(" << c.Re() << ", " << c.Im() << ")";

return out;

}

istream& operator>>(istream& in, Complex& c)

{

in >> c.m\_re >> c.m\_im;

return in;

}

Complex sinh(const Complex& c)

{

return Complex(sinh(c.Re())\*cos(c.Im()), cosh(c.Re())\*sin(c.Im()));

}

Листинг 2 – программный код файла Header.h

#include "math.h"

#include<iostream>

using namespace std;

class Complex

{

public:

Complex();

Complex(double);

Complex(double, double);

double Phi() const;

double R() const;

double Re() const;

double Im() const;

Complex& operator+=(const Complex& right);

Complex& operator-=(const Complex& right);

Complex& operator\*=(const Complex& right);

Complex& operator/=(const Complex& right);

Complex operator+ ( const Complex& right);

Complex operator- ( const Complex& right);

Complex operator\* ( const Complex& right);

Complex operator/ ( const Complex& right);

Complex operator- ();

explicit operator double()

{

return m\_re;

}

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

friend istream& operator>>(istream& in, Complex& c);

private:

double m\_re = 0;

double m\_im = 0;

};

Complex sinh(const Complex& c);

Листинг 3 – программный код файла main.cpp

#include<iostream>

#include "math.h"

#include "Header.h"

using namespace std;

Complex y(Complex& z)

{

Complex i(0, 1);

Complex res = i + z \* sinh(z + 1);

return res;

}

int main()

{

setlocale(LC\_ALL, "RUS");

Complex a, res;

cout << "Введите действительную и мнимую часть комплексного числа" << endl;

cin >> a;

res = y(a);

cout << res;

}

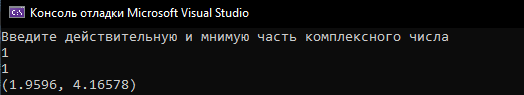


Рисунок 1 – Результат работы с программой

**<https://github.com/Pavel-Plotnikov-BSU1801/-Laboratornai-8-Plotnikov-P.S.-BSU1801>**

**Вывод**

В данной работе были разобрана работа с классами в языке программирования С++, а именно перегрузка операторов и работа с публичными и приватными полями класса.