**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

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

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра «Информационная безопасность»

**ЛАБОРАТОРНАЯ РАБОТА**

**№5**

по дисциплине «Объектно-ориентированное программирование систем защиты информации»

на тему:

**«Понятие класса. Члены класса. Друзья класса. Перегрузка операторов»**

Вариант №10

Выполнил: студ. гр. БПЗ1501

Лисеенко Я.А.

Проверил: ассистент кафедры ИБ

Барков В.В.

Москва 2018 г

# Цель работы

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

# Результаты

## Complex.cpp

#include "Complex.h"

#include <math.h>

double Complex::getReal() const

{

return real;

}

double Complex::getImaginary() const

{

return imaginary;

}

void Complex::setReal(double real)

{

this->real = real;

}

void Complex::setImaginary(double imaginary)

{

this->imaginary = imaginary;

}

Complex::Complex(){

real = 0.0;

imaginary = 0.0;

}

Complex::Complex(double re, double imag)

{

real = re;

imaginary = imag;

}

Complex epow(const Complex &z)

{

double real, imag;

double expR, expI;

real = z.getReal();

imag = z.getImaginary();

expR = exp(real)\*cos(imag);

expI = exp(real)\*sin(imag);

return Complex(expR, expI);

}

Complex logC(Complex z)

{

double r, phi;

r = sqrt(pow(z.getReal(), 2) + pow(z.getImaginary(), 2));

phi = atan2(z.getImaginary(), z.getReal());

return Complex(log(r), phi);

}

Complex powC(Complex a, Complex b)

{

return epow(b\*logC(a));

}

Complex operator -(Complex a, Complex b)

{

return Complex(a.getReal() - b.getReal(), a.getImaginary() - b.getImaginary());

}

Complex operator +(Complex a, Complex b)

{

return Complex(a.getReal() + b.getReal(), a.getImaginary() + b.getImaginary());

}

Complex operator \*(Complex a, Complex b)

{

return Complex(a.getReal() \* b.getReal() - a.getImaginary() \* b.getImaginary(), a.getImaginary() \* b.getReal() + a.getReal() \* b.getImaginary());

}

Complex operator /(Complex a, Complex b)

{

return Complex(((a.getReal() \* b.getReal()) - (a.getImaginary()\*b.getImaginary()\*(-1))) / (pow(b.getReal(), 2) + pow(b.getImaginary(), 2)), ((a.getReal()\*b.getImaginary()\*(-1)) + (a.getImaginary()\*b.getReal())) / (pow(b.getReal(), 2) + pow(b.getImaginary(), 2)));

}

## Complex.h

#pragma once

#include "iostream"

using namespace std;

class Complex

{

double real, imaginary;

public:

Complex();

double getReal() const;

double getImaginary() const;

void setReal(double real);

void setImaginary(double imaginary);

Complex(double real, double imaginary);

friend Complex operator \*(Complex a, Complex b);

friend Complex operator -(Complex a, Complex b);

friend Complex operator /(Complex a, Complex b);

friend Complex operator +(Complex a, Complex b);

friend Complex powC(Complex z, Complex a);

friend Complex logC(Complex z);

friend Complex epow(const Complex &z);

};

## main.cpp

#include "Complex.h"

int main()

{

setlocale(LC\_ALL, "Rus");

int real, imaginary;

//Input part

printf("Input real part: ");

scanf\_s("%d", &real);

printf("\n");

printf("Input imaginary part: ");

scanf\_s("%d", &imaginary);

printf("\n");

Complex z(real, imaginary);

Complex i (0, 1);

//Output

printf("z= %d %di", real, imaginary);

printf("\nf(z) = z^3+(1 + 2i)z^2+(1-2i)z^(-5i)\n");

Complex res = powC(z, Complex(3, 0))+(Complex(1,0) + Complex(2,0)\*i)\* powC(z, Complex(2, 0))+ (Complex(1,0) - Complex(2,0)\*i)\* powC(z, Complex(-5, 0) \* i);

printf("result: %f %fi", res.getReal(), res.getImaginary());

}



1. Результат работы программы