**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»

КОНСПЕКТ

по дисциплине:

«Программирование в системах информационной безопасности»

на тему:

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

Вариант №17

Выполнил:

студент 4 курса

группы БСУ1801

Неронов Ф.А.

Проверил:

Барков В.В.

Москва 2021

1. **Цель работы:**

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

1. **Общее задание.**

Выберите для выполнения лабораторной работы свой вариант.

Для выбранного варианта определите класс, включив в него:

* конструктор по умолчанию;
* конструктор инициализации;
* конструктор преобразования базового типа к типу, определяемому разрабатываемым классом;

В разрабатываемом классе перегрузите потоковые операции для объектов класса. Для выполнения задания предложенного варианта перегрузите необходимые математические операции.

При разработке класса вашего варианта учтите индивидуальные уточнения для функций – членов класса.

Разработайте для объектов вашего класса предложенные в каждом варианте пользовательские функции.

Разработайте функцию main, организующую ввод данных и демонстрацию работы разработанных функций

1. **Индивидуальное задание**
2. **Результаты**

|  |  |
| --- | --- |
| z = a + ib  a = 1; b = 0  y(z) = 2.4 | z = a + ib  a = 1; b = 1  y(z) = -2.5 - 4i |

1. **Выполнение**

Листинг 1 *–* Исходный код файла main\_f.cpp

#include "head.h"

Complex y(Complex& z)

{

Complex z1 = Complex(1, 0);

Complex z2 = pow1(z, Complex(5, 0));

Complex z3 = Complex(th(z) / 2, 0);

Complex z4 = z1 + z2 + z3;

return z4;

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a, b;

cout << "\nВведите реальную часть компексного числа, потом комплексную\n";

cin >> a >> b;

Complex h(a, b);

cout << "\nРезультат работы функции:\n";

cout << y(h) << "\n";

system("pause");

return 0;

}

Листинг 2 *–* Исходный код файла head.h

#pragma once

#include <iostream>

using namespace std;

class Complex

{

public:

Complex()

{

Re = 0;

Im = 0;

}

Complex(double re, double im)

{

Re = re;

Im = im;

}

operator double()

{

return R();

}

double R()

{

return r();

}

double Phi()

{

return phi();

}

double Re, Im;

Complex operator+(Complex& a);

Complex operator-(Complex& a);

Complex operator/(Complex& a);

Complex operator\*(Complex& a);

friend std::ostream& operator<<(std::ostream& out, const Complex& ex);

private:

double r()

{

return sqrt(Re \* Re + Im \* Im);

}

double phi()

{

return tan(Im / Re);

}

};

extern Complex pow1(Complex, Complex);

extern double sh(Complex);

extern double ch(Complex);

extern double th(Complex);

Листинг 3 *–* Исходный код файла complex.cpp

using namespace std;

#include <iostream>

#include "head.h"

//Переопределение операций числа типа complex

//Сложение

Complex pow1(Complex z1, Complex z2)

{

Complex h = z1;

for (int i = 1; i < (double)z2; i++)

{

h = h \* z1;

}

return h;

}

double sh(Complex z)

{

return (exp((double)z) - exp(-(double)z)) / 2;

}

double ch(Complex z)

{

return (exp((double)z) + exp(-(double)z)) / 2;

}

double th(Complex z1)

{

return sh(z1) / ch(z1);

}

ostream& operator<<(ostream& stream, const Complex& instance)

{

if (instance.Im >= 0)

{

stream << instance.Re << " + " << instance.Im << "i";

}

else

{

stream << instance.Re << " " << instance.Im << "i";

}

return stream;

}

Complex Complex::operator+(Complex& b)

{

Complex c;

c.Re = Re + b.Re;

c.Im = Im + b.Im;

return c;

}

//Вычитание

Complex Complex::operator-(Complex& b)

{

Complex c;

c.Re = Re - b.Re;

c.Im = Im - b.Im;

return c;

}

//Умножение

Complex Complex::operator\*(Complex& b)

{

Complex c;

c.Re = Re \* b.Re - Im \* b.Im;

c.Im = Im \* b.Re + Re \* b.Im;

return c;

}

//Деление

Complex Complex::operator/(Complex& b)

{

Complex c;

c.Re = (Re \* b.Re + Im \* b.Im) / (b.Re \* b.Re + b.Im \* b.Im);

c.Im = (Im \* b.Re + Re \* b.Im) / (b.Re \* b.Re + b.Im \* b.Im);

return c;

}

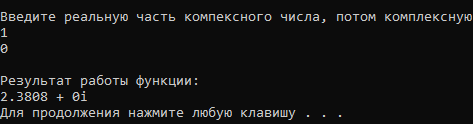


Рис.4.1 – Результат выполнения программы

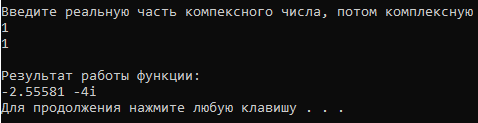


Рис.4.2 – Результат выполнения программы

https://github.com/BSU1801/Barkov\_lab8.git

1. **Вывод:**

В этой лабораторной работе мы научились создавать простые классы, создавать конструкторы, перегружать операции, создавать объекты классов и передавать их в функции.