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

«Динамическое выделение памяти»

Вариант 9

Выполнил:

студент группы БСУ1801

Копылов М.А.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

# Цель работы

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

# 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

**Общее Задание**

**Вариант 9.**

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

|  |  |  |  |
| --- | --- | --- | --- |
| Варианты | 1-5 | 6-10 | 11-15 |
| Название класса | Complex | Complex | Rational |
| Приватные данные-члены | double re; (действительная часть)  double im;  (мнимая часть) | double r;  (модуль)  double phi;  (аргумент) | int nominator;  (числитель)  int denominator;  (знаменатель)  Функция сокращения дроби |
| Публичные функции-члены | double Re() const;  double Im() const;  double R() const;  double Phi() const; | | int GetNominator() const;  int GetDenominator() const; |
| Пользовательские функции | Complex y(const Complex &z); | Complex y(const Complex &z); | Rational y1(const Rational &x);  double y2(double x); |
| 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 sin(const Complex &z);  Complex tg(const Complex &z);  Complex th(const Complex &z);  Complex sh(const Complex &z);  Complex pow(const Complex &z1, const Complex &z2); |  |

**Индивидуальное задание**

# Выполнение

Листинг 1 – программный код файла complex.cpp

#include "Header.h"

#include "math.h"

#include<iostream>

//Конструктор по умолчанию

Complex::Complex():

m\_r{ sqrt(0)}, m\_phi{ atan2(0, 0) }

{}

//Конструктор преобразования

Complex::Complex(double re):

m\_r{ sqrt(re \* re) }, m\_phi{ atan2(0, 0) }

{}

//Конструктор инициализации

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

m\_r{ sqrt(re \* re + im \* im) }, m\_phi{ atan2(im, re) }

{}

//Публичные функции-члены

double Complex::Re() const

{

return m\_r \* cos(m\_phi);

}

double Complex::Im() const

{

return m\_r \* sin(m\_phi);

}

double Complex::R() const

{

return m\_r;

}

double Complex::Phi() const

{

return m\_phi;

}

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(), Re() \* right.Im() + Im() \* right.Re());

return \*this;

}

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

{

\*this = Complex((Re() \* right.Re() + Im() \* right.Im()) / (pow(right.Re(), 2) + pow(right.Im(), 2)), (Im() \* right.Im() - Re() \* right.Im()) / (pow(right.Re(), 2) + pow(right.Im(), 2)));

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;

}

//Перегрузка операторов вывода

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

{

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

return out;

}

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

{

double re, im;

in >> re >> im;

c.m\_r = sqrt(re \* re + im \* im);

c.m\_phi = atan2(im, re);

return in;

}

//Перегрузка тригонометрической функции

Complex tg(const Complex& c)

{

return Complex

(

sin(2\*c.Re())/(cos(2\*c.Re())+cosh(2\*c.Im())),

sinh(2\*c.Im())/(cos(2\*c.Re()) + cosh(2 \* c.Im()))

);

}

Листинг 2 – программный код файла Headher.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\_r \* static\_cast<double>(cos(m\_phi));

}

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

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

private:

double m\_r = 0; //модуль

double m\_phi = 0; //аргумент

};

Complex tg(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 res = z \* tg(z) + 2;

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.