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

Дисциплина: Программирование в системах информационной безопасности

Лабораторная работа №8

Понятие класса. Члены класса. Друзья класса. Перегрузка операций

Выполнил:

студент группы БСУ1801 Убушуев Б.С

Проверил:

старший преподаватель кафедры ИБ

Барков В. В.

Москва 2021

### ЗАДАНИЕ

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.

Разработайте для объектов вашего класса предложенные в каждом варианте пользовательские функции. Разработайте функцию main, организующую ввод данных и демонстрацию работы разработанных функций

**ИНДИВИДУАЛЬНЫЙ ВАРИАНТ ЗАДАНИЯ №25(10)**

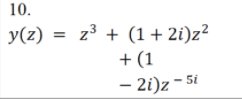
В соответствии с Вашим вариантом разработать класс. Название класса, приватные данные-члены и необходимые публичные функции-члены указаны в таблице ниже.

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.

Таблица 1 – Индивидуальный вариант задания

|  |  |  |  |
| --- | --- | --- | --- |
| 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); |  |



# 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());

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;

}

//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 sin(const Complex& c)

{

return Complex(sin(c.Re()) \* cosh(c.Im()), cos(c.Re()) \* sinh(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 sin(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 = Pow(z, 3) + ((1+2\*i) \* Pow(z, 2)) + ((1 – 2\*i) \* Pow(z, -5\*i)) ;

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;

}

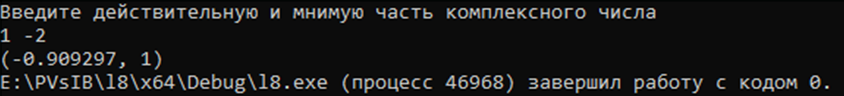


Рисунок 1 – Выполнение l8.exe

**ВЫВОД**

В ходе данной работы получен навык циклических алгоритмов на языке C++ с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.