ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский Технический Университет Связи и Информатики» (МТУСИ)

Кафедра информатики

Лабораторная работа по языкам программирования №2

«Совместное использование операций»

Выполнила:

Студентка 2 курса

Группы БСТ-1602

Серкова Олеся

Вариант №14

Преподаватель:

Скрыпникова М.К.

Москва, 2018

### 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.

### Индивидуальное задание

1. Для вариантов 1-5 определить класс - "Комплексное число" в виде действительной и мнимой частей комплексного числа.
2. Составить пользовательскую функцию, которая получает значение типа "Комплексное число", вычисляет значение математической функции y(z), заданной в варианте, и возвращает полученное значение.
3. Составить пользовательскую функцию, которая получает значение стандартного типа ***complex***, вычисляет значение математической функции y(z), заданной в варианте, и возвращает полученное значение.
4. Составить пользовательскую функцию, которая вычисляет таблицу значений заданной математической функции y(z1), где z1, значение типа "Комплексное число", и у(z), где z, значение типа ***complex.*** Таблица вычисляется для комплексных чисел, для изменения действительной части комплексных значений в диапазоне -1<=Re(z)<=1, а мнимой в диапазоне 1<=Im(z)<=1 с шагом 0.5 по обоим аргументам.
5. Составить пользовательскую функцию, которая вводит значение комплексного числа с клавиатуры в объект z типа "Комплексное число", вычисляет значение у(z), сравнивает его с таким же вычислением, но для значения типа ***complex.*** На экран функция выдает вычисленные значения и результат сравнения.
6. Составить функцию main, органицующую работу всех функций.

Вариант 14. y(z) = 2 + z tg(z)

**Код программы:**

**//my.h**

#pragma once

#include <iostream>

#include <cmath>

#include <complex>

#include "iomanip"

using namespace std;

class Complex

{

double re, im; //re - действительная часть, im - мнимая часть

public:

Complex() { re = im = 0; } // конструктор по умолчанию

Complex(double a) { re = a, im = 0; } // конструктор преобразования

Complex(double a, double b) { re = a, im = b; } // конструктор инициализации

friend Complex Tg(Complex); // функция, вычисляющая тангенс комплексного числа

friend Complex operator \* (Complex, Complex); // оператор умножения

friend Complex operator + (Complex, Complex); // оператор сложения

friend ostream& operator << (ostream&, Complex&); // потоковая операция вывода

friend istream& operator >> (istream&, Complex&); // потоковая операция ввода

double getre() { return re; } // функция, возвращающая действительную часть

double getim() { return im; } // функция, возвращающая мнимую часть

};

Complex Tg(Complex z)

{

Complex a;

a.re = sin(z.re\*2)/(cos(z.re\*2)+cosh(z.im\*2));

a.im = sinh(z.im\*2)/(cos(z.re \* 2) + cosh(z.im \* 2));

return a;

}

Complex operator \* (Complex x, Complex y) // перегрузка оператора \*

{

Complex b;

b.re = x.re \* y.re - x.im \* y.im;

b.im = x.re \* y.im + x.im \* y.re;

return b;

}

Complex operator + (Complex x, Complex y) // перегрузка оператора -

{

Complex c;

c.re = x.re + y.re;

c.im = x.im + y.im;

return c;

}

ostream& operator << (ostream& S, Complex& a) { S << "(" << a.re << "," << a.im << ")"; return S; } // перегрузка оператора вывода <<

istream& operator >> (istream& S, Complex& a) { S >> a.re >> a.im; return S; } // перегрузка оператора ввода >>

Complex Fun(double re, double im) // 2 + z tg(2z) пользовательская

{

Complex z = Complex(re, im);

return (Complex(2) + z \* Tg(Complex(2) \* z));

}

complex <double> fun(double re, double im) // 2 + z tg(2z) системная

{

complex <double> z = complex <double>(re, im);

return (complex <double>(2) + z \* tan ((complex <double>) 2 \* z));

}

void check() // функция сравнения двух типов

{

Complex z;

cin >> z;

complex <double> z1(z.getre(), z.getim());

Complex ans = Fun(z.getre(), z.getim());

complex <double> aans = fun(real(z1), imag(z1));

if ((real(aans) - ans.getre() < 0.0001) && (imag(aans) - ans.getim() < 0.0001))

cout << endl << "Результаты совпадают:" << endl;

else

cout << endl << "Результаты не совпадают:" << endl;

cout << "\n f(z) = " << ans << " (пользовательский тип)" << endl << "f(z1) = " << aans << " (встроенный тип)" << endl;

}

void table() // функция вывода таблицы результатов

{

for (double i = -1; i <= 1; i += 0.5)

for (double j = -1; j <= 1; j += 0.5)

{

cout << "\n" << setw(4) << i << setw(8) << j << setw(16) << Fun(i, j) << setw(24) << fun(i, j);

};

}

**//main.cpp**

#include "my.h"

void main()

{

setlocale(LC\_ALL, "Russian");

cout << endl << "Re(z)\tIm(z)\t\t Пользовательский тип\t Встроенный тип";

table(); // вывод таблицы значений

cout << endl << endl << "Введите комплексное число:" << endl;

check(); // сравнение пользовательского и встроенного типов

cout << endl;

system("pause");

}

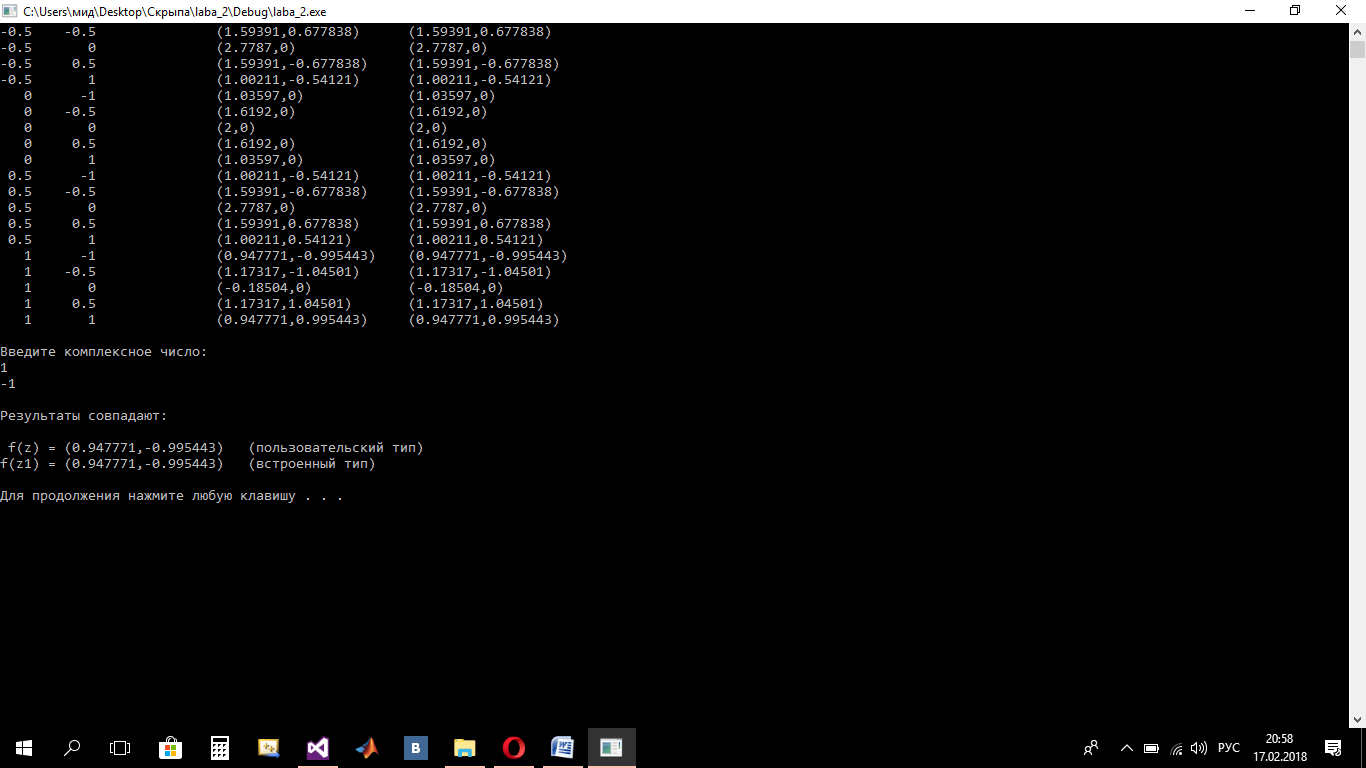


Рисунок 1 - Результат работы программы