**ЛАБОРАТОРНАЯ РАБОТА №1**

**СОЗДАНИЕ МНОГОФАЙЛОВЫХ ПРОЕКТОВ**

**Цель**:

Целью данной лабораторной работы освоить на практике создание многофайловых проектов на языке Си/Си++, познакомиться с директивами условной компиляции.

**Задания:**

1. Напишите программу–калькулятор комплексных чисел. Для реализации необходимо разработать абстрактный тип данных– комплексное число. Программа должна реализовывать арифметические операции над комплексными числами. Программа должна быть представлена в виде многофайлового проекта, все прототипы функций, объявления структур должны быть вынесены в заголовочный файл с соответствующим названием. Всего должно быть три файла: файл с объявлениями, файл реализации и файл с функцией main(), демонстрирующий работу с новым типом данных. Файл с объявлениями должен называться Complex.h, файл с реализацией функций должен называться Complex.с, файл с функцией main() может называться main.с Программа должна обеспечивать удобный интерфейс пользователя для работы с ней.

2. На основе разработанного в предыдущем задании типа данных комплексное число написать программу, которая считывает информацию из файла complex.txt — количество комплексных чисел в переменную n, а сами комплексные числа в массив p. Затем происходит поиск комплексного числа с максимальным модулем в массиве p.

**Задание №1**

Lab 1.1.cpp

#include "pch.h"

#include "windows.h"

#include "Complex.h"

#include <iostream>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("color 0A");

Complex a;

a.set();

a.out();

a.plus();

a.minus();

a.multiplication();

a.division();

system("pause");

return 0;

}

Complex.cpp

#include "pch.h"

#include "Complex.h"

#include<iostream>

#include<string>

using namespace std;

void Complex::set()

{

cout << "Введите вещественную и мнимую часть перового числа" << endl;

cin >> x1 >> y1;

cout << "Введите вещественную и мнимую часть перового числа" << endl;

cin >> x2 >> y2;

}

void Complex::out()

{

cout << x1 << "+i\*" << y1 << endl;

cout << x2 << "+i\*" << y2 << endl;

}

void Complex::plus()

{

cout << "Сложение" << endl;

cout << x1 + x2 << "+i\*" << y1 + y2 << endl;

}

void Complex::minus()

{

cout << "Вычитание" << endl;

cout << x1 - x2 << "+i\*" << y1 - y2 << endl;

}

void Complex::multiplication()

{

cout << "Умножение" << endl;

cout << x1 \* x2 - y1 \* y2 << "+i\*" << x1 \* y2 + x2 \* y1 << endl;

}

void Complex::division()

{

cout <<"Деление"<< endl;

cout << ((x1\*x2+y1\*y2)/(x1\*x1+y1\*y1)) << "+i\*" <<((x1\*y2-x2\*y1)/ (x1\*x1 + y1 \* y1)) << endl;

}

Complex.h

#pragma once

#include<iostream>

#include<string>

using namespace std;

class Complex

{

private:

double x1, x2, y1, y2;

public:

void set();

void out();

void plus();

void minus();

void multiplication();

void division();

};

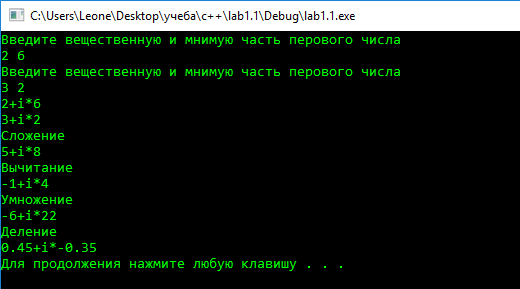


Рис.1

**Задание №2**

Lab 1.2.cpp

#include "pch.h"

#include "windows.h"

#include <iostream>

#include <vector>

#include <fstream>

#include "Complex.h"

using namespace std;

const Complex readNumber(ifstream&); // Считывает одно комплексное число из файла

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("color 0A");

ifstream f("complex.txt");

int maxModule = -1;

Complex max;

int n;

f >> n;

for (int i = 0; i < n; i++)

{

Complex c;

c = readNumber(f);

if (c.countModule() > maxModule)

{

max = c;

maxModule = c.countModule();

}

}

cout << max << " Максимальный модуль:" << max.countModule() << endl;

f.close();

system("pause");

return 0;

}

const Complex readNumber(ifstream& f)

{

double re, im;

f >> re >> im;

f.get();

return Complex(re, im);

}

Complex.cpp

#include "pch.h"

#include "Complex.h"

Complex::Complex()

{

setNumber(0, 0);

}

Complex::Complex(double re, double im)

{

setNumber(re, im);

}

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

{

this->im = im;

this->re = re;

}

double Complex::getRe()

{

return re;

}

double Complex::getIm()

{

return im;

}

double Complex::countModule()

{

return sqrt(re \* re + im \* im);

}

Complex& Complex::operator= (const Complex& c)

{

re = c.re;

im = c.im;

return \*this;

}

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

{

out << c.re;

if (c.im < 0)

{

out << c.im << "i";

}

else

{

out << "+" << c.im << "i";

}

return out;

}

Complex::~Complex()

{

}

Complex.h

#pragma once

#include <iostream>

#include <cmath>

#ifndef COMPLEX\_H\_

#define COMPLEX\_H\_

class Complex {

double re; // Действительная часть числа

double im; // Мнимая часть числа

public:

Complex(); // Создаёт число равное 0

Complex(double, double); // Создает число с известными мнимой и действительной частями

void setNumber(double, double); // Задаёт действительную и мнимую часть

double getRe(); // Возвращает действительную часть числа

double getIm(); // Возвращает мнимую часть числа

double countModule(); // Возвращает модуль комплексного числа

Complex& operator= (const Complex&); // Перегруженный оператор присвоения

friend std::ostream& operator<< (std::ostream&, const Complex&); // Вывод

~Complex(); // Деструктор

};

#endif /\* COMPLEX\_H\_ \*/

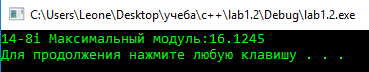


Рис.2

**Вывод:** В ходе выполнения лабораторной работы №1 был реализован многофайловый проект, а также создан класс для работы с комплекснами числами.