**Міністерство освіти і науки України**

**Запорізький національний технічний університет**

кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №1

з дисципліни «Об’єктно-орієнтоване програмування» на тему:

Динамічні класові типи

Варіант № 7

Виконав

студент групи КНТ-126 Костюк П.М.

Прийняла

викладач: Твердохліб Ю.В.

2017

**Лабораторна робота №1**

**Динамічні класові типи**

**Мета роботи:**

Навчитись будувати класи та використовувати їх при створенні програм.

**Індивідуальні завдання**

Створити клас для виконання операцій з комплексними числами. Передбачити операції: складання та віднімання; множення та ділення; обчислення модулю; консольне введення та виведення; ініціалізацію.

Complex.h

#include **<iostream>**

#include **<sstream>**

#include **<cmath>**

#include **<string>**

**using namespace** std;

**class** Complex {

**private**:

**double** re, im;

**public**:

Complex() {

re = 0;

im = 0;

}

Complex(**double** r, **double** i) {

re = r;

im = i;

}

Complex(**const** Complex &ob) {

re = ob.re;

im = ob.im;

};

Complex add(Complex);

Complex subtract(Complex);

Complex multiply(Complex &);

Complex devide(Complex &);

Complex abs(Complex);

**friend** istream &**operator**>>(istream &, Complex &);

**friend** ostream &**operator**<<(ostream &, **const** Complex &);

};

# Complex.cpp

#include **"Complex.h"**

**using namespace** std;

Complex Complex::multiply(Complex &com) {

**double** i, j;

i = **this**->re \* com.re - **this**->im \* com.im;

j = **this**->re \* com.im + com.re \* **this**->im;

**this**->re = i;

**this**->im = j;

**return** \***this**;

}

Complex Complex::devide(Complex &com) {

**double** i, j, k;

k = re \* re + com.im \* com.im;

i = (re \* com.re + im \* com.im) / k;

j = (com.re \* im - re \* com.im) / k;

re = i;

im = j;

**return** \***this**;

}

Complex Complex::add(Complex com) {

**this**->re = **this**->re + com.re;

**this**->im = **this**->im + com.im;

**return** \***this**;

}

Complex Complex::subtract(Complex com) {

**this**->re = **this**->re - com.re;

**this**->im = **this**->im - com.im;

**return** \***this**;

}

ostream &**operator**<<(ostream &out, **const** Complex &com) {

**if** (com.im < 0)

out << com.re << **"+i("** << com.im << **")\n"**;

**else**

out << com.re << **"+i"** << com.im << **"\n"**;

**return** out;

}

istream &**operator**>>(istream &in, Complex &com) {

cout << **"Введите действительную часть комплексного числа "**;

in >> com.re;

cout << **"Введите мнимую часть комплексного числа "**;

in >> com.im;

**return** in;

}

Complex Complex::abs(Complex com) {

**this**->re = std::abs(**this**->re);

**this**->im = std::abs(**this**->im);

**return** \***this**;

}

# main.cpp

#include **<iostream>**

#include **"Complex.h"**

**using namespace** std;

**int** main() {

Complex complex1;

Complex complex2;

cin >> complex1;

cin >> complex2;

cout << **"Выберите операцию:"**;

string operation;

cin >> operation;

**if** (operation == **"+"**){

Complex sum = complex1.add(complex2);

cout << **"Сложение: "** << sum << endl;

}

**if** (operation == **"-"**){

Complex subtract = complex1.subtract(complex2);

cout << **"Вычитание: "** << subtract << endl;

}

**if** (operation == **"abs"**){

Complex abs = complex1.abs(complex2);

cout << **"Модуль: "** << abs << endl;

}

**if** (operation == **"\*"**){

Complex multiply = complex1.multiply(complex2);

cout << **"Умножение: "** << multiply << endl;

}

**if** (operation == **"/"**){

Complex devide = complex1.devide(complex2);

cout << **"Деление: "** << devide << endl;

}

}