

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Отчёт по Лабораторной работе №1
“Простые Классы“
по курсу “Объектно-Объективное Программирование“
III Семестр

Студент:	Аксенов А. Е.
Группа:	М8О-208Б-18
Преподаватель:	Журавлёв А.А.
Оценка:	
Дата:	30.09.19

1. Тема: Простые классы в C++.

2. Код программы:

complex.h

```
#ifndef D_COMPLEX_H
#define D_COMPLEX_H

#include <iostream>

struct Complex {
private:
    double real;
    double image;

public:
    void read(std::istream& is);
    void awrite(std::ostream& os) const;

    Complex add(const Complex& rhs) const;
    Complex sub(const Complex& rhs) const;
    Complex mul(const Complex& rhs) const;
    Complex div(const Complex& rhs) const;
    Complex conj(const Complex& rhs) const;

    int mod_comp(const Complex& rhs) const;
    int equ(const Complex& rhs) const;

};

#endif
```

comlex.cpp

```
#include "complex.h"
#include <cmath>

#include <iostream>

void Complex::read(std::istream& is) {
    is >> real >> image;
}

void Complex::awrite(std::ostream& os) const {
    os << real << ' ' << image << "\n";
}

Complex Complex::add(const Complex& rhs) const {
    Complex d;
    d.real = real + rhs.real;
    d.image = image + rhs.image;
    return d;
}

Complex Complex::sub(const Complex& rhs) const {
    Complex d;
    d.real = real - rhs.real;
    d.image = image - rhs.image;
    return d;
}

Complex Complex::mul (const Complex& rhs) const {
    Complex d;
    d.real = real * rhs.real - image * rhs.image;
    d.image = real * rhs.image + rhs.real * image;
```

```

    return d;
}

Complex Complex::div (const Complex& rhs) const {
    Complex d;
    d.real = (real * rhs.real + image * rhs.image)/(rhs.real * rhs.real + rhs.image * rhs.image);
    d.image = (image * rhs.real - real * rhs.image)/(rhs.real * rhs.real + rhs.image * rhs.image);
    return d;
}

int Complex::equ (const Complex& rhs) const {
    int result = 0;
    if (real == rhs.real && image == rhs.image) {
        result = 1;
    } else {
        result = 0;
    }
    return result;
}

Complex Complex::conj (const Complex& rhs) const {
    Complex d;
    d.real = real;
    d.image = image * -1;
    return d;
}

int Complex::mod_comp (const Complex& rhs) const {
    int result = 0;
    double a = sqrt(real*real + image*image);
    double b = sqrt(rhs.real*rhs.real + rhs.image*rhs.image);

    if (a > b) {
        result = 1;
    } else if (a < b) {
        result = -1;
    } else {
        result = 0;
    }
    return result;
}

```

lab1.cpp

```

#include <iostream>

#include "complex.h"

int main() {
    Complex a;
    Complex b;
    //std::cout << "Enter numbers:\n[real] and [image]" << std::endl;
    a.read(std::cin);
    b.read(std::cin);
    //std::cout << "Real:\n";
    //std::cout << "Image:\n";
    std::cout << "SUM:\n";
    a.add(b).awrite(std::cout);
    std::cout << "SUB:\n";
    a.sub(b).awrite(std::cout);
    std::cout << "MULTIPLY:\n";
    a.mul(b).awrite(std::cout);
    std::cout << "DIVIDE:\n";
    a.div(b).awrite(std::cout);
}

```

```

std::cout << "EQUAL:\n";

switch (a.equ(b)) {
case 0:
    std::cout << "Numbers are not equal" << std::endl;
    break;

case 1:
    std::cout << "Numbers are equal" << std::endl;
    break;
}

std::cout << "Conjugate number for a:\n";
a.conj(b).awrite(std::cout);
std::cout << "Module compare for a and b:\n";

switch (a.mod_comp(b)) {
case 1:
    std::cout << "Module a more module b" << std::endl;
    break;

case -1:
    std::cout << "Module a less than module b" << std::endl;
    break;

case 0:
    std::cout << "Module a is equal to module b" << std::endl;
    break;
}

return 0;
}

```

3. Ссылка на репозиторий:

https://github.com/fallfire13/oop_exersize_1

4. Набор testcases:

test_00.test

30 16
20 47

test_00.result

SUM:
50 63
SUB:
10 -31
MULTIPLY:
-152 1730
DIVIDE:
0.518206 -0.417785
EQUAL:
Numbers are not equal
Conjugate number for a:
30 -16
Module compare for a and b:
Module a less than module b

test_01.test

6 7
6 7

test_01.result

SUM:
12 14
SUB:
0 0
MULTIPLY:
-13 84
DIVIDE:
1 0
EQUAL:
Numbers are equal
Conjugate number for a:
6 -7
Module compare for a and b:
Module a is equal to module b

test_02.test

12 13
0 15

test_02.result

SUM:
12 28
SUB:
12 -2
MULTIPLY:
-195 180
DIVIDE:
0.866667 -0.8
EQUAL:
Numbers are not equal
Conjugate number for a:
12 -13
Module compare for a and b:
Module a more module b

test_03.test

27 9
8 0

test_03.result

SUM:
35 9
SUB:
19 9
MULTIPLY:
216 72
DIVIDE:
3.375 1.125
EQUAL:
Numbers are not equal
Conjugate number for a:
27 -9
Module compare for a and b:
Module a more module b

5. Результаты выполнения тестов:

```
fallfire13@DESKTOP-M7F3IHA:~/oop_lab1/tests$ bash test.sh ../lab1
```

```
Test test_00.test: SUCCESS
```

```
Test test_01.test: SUCCESS
```

```
Test test_02.test: SUCCESS
```

```
Test test_03.test: SUCCESS
```

6. Объяснение результатов работы программы:

- 1) Программа вводит две пары действительных чисел (a, b) и (c, d), представляющих комплексные числа в алгебраической форме.
- 2) Программа выводит результат сложения чисел (a; b) и (c, d).
- 3) Программа выводит результат вычитания числа (c, d) из числа (a, b).
- 4) Программа выводит результат умножения чисел (a, b) и (c, d).
- 5) Программа выводит результат деления числа (a, b) на число (c, d).
- 6) Программа сравнивает два числа (a, b) и (c, d).
- 7) Программа находит число, сопряжённое (a, b).
- 8) Программа выводит результат сравнения модулей чисел (a, b) и (c, d).

7. Вывод: 1) Ознакомились с простыми классами в C++ и усвоили навык работы с ними; 2) Написана программа, производящая операции с простым классом.