

Московский Авиационный Институт
(Национальный исследовательский Университет)

Факультет: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторная работа
по курсу «ООП»**

**Тема:
Простые классы.**

Студент:	Симонов С.Я.
Группа:	М8О-206Б-18
Преподаватель:	Журавлев А.А.
Вариант:	1
Оценка:	
Дата:	

Москва
2019

1. Код программы на языке C++:

complex1.h

```
#ifndef OOP_01_COMPLEX_H
#define OOP_01_COMPLEX_H

#include <iostream>
#include <cmath>

class complex {
public:
    complex();

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

    complex add(const complex& o) const;
    complex sub(const complex & o) const;
    complex mul(const complex & o) const;
    complex div(const complex & o) const;
    bool equ(const complex & o) const;
    complex con1() const;
    int modul(const complex & o) const;
private:
    double array[2];
};

#endif
```

complex1.cpp

```
#include "complex1.h"

void complex::read(std::istream& is) {
    for (int i = 0; i < 2; ++i) {
        is >> array[i];
    }
}

void complex::write(std::ostream& os) const {
```

```

    for (int i = 0; i < 2; ++i) {
        os << array[i] << " ";
    }
}

```

```

complex::complex() {
    array[0] = 0;
    array[1] = 0;
}

```

```

complex complex::add(const complex& o) const {
    complex result;
    for (int i = 0; i < 2; ++i) {
        result.array[i] = array[i] + o.array[i];
    }

    return result;
}

```

```

complex complex::sub(const complex& o) const {
    complex result;
    for (int i = 0; i < 2; ++i) {
        result.array[i] = array[i] - o.array[i];
    }

    return result;
}

```

```

complex complex::mul(const complex & o) const {
    complex result;
    result.array[0] = (array[0] * o.array[0]) - (array[1] * o.array[1]);
    result.array[1] = (array[0] * o.array[1]) + (array[1] * o.array[0]);
    return result;
}

```

```

complex complex::div(const complex & o) const {
    complex result;
    result.array[0] = ((array[0] * o.array[0]) + (array[1] * o.array[1])) / ((o.array[1] *
o.array[1]) + (o.array[0] * o.array[0]));
    result.array[1] = ((array[1] * o.array[0]) - (array[0] * o.array[1])) / ((o.array[0] *
o.array[0]) + (o.array[1] * o.array[1]));
    return result;
}

```

```

bool complex::equ(const complex & o) const {
    if ((array[0] == o.array[0]) && (array[1] == o.array[1])) {

```

```

        return 1;
    }
    return 0;
}

```

```

complex complex::con1() const {
    complex result;
    result.array[0] = array[0];
    result.array[1] = -array[1];
    return result;
}

```

```

int complex::modul(const complex & o) const {
    double a, b;
    a = sqrt((array[0] * array[0]) + (array[1] * array[1]));
    b = sqrt((o.array[0] * o.array[0]) + (o.array[1] * o.array[1]));
    if (a == b) {
        return 0;
    } else if (a > b) {
        return -1;
    }
    return 1;
}

```

main1.cpp:

```

#include <iostream>
#include <cmath>

#include "complex1.h"

int main()
{
    complex a;
    complex b;
    a.read(std::cin);
    b.read(std::cin);
    a.write(std::cout);
    std::cout << "\n";
    b.write(std::cout);
    std::cout << "\n";
    a.add(b).write(std::cout);
    std::cout << "\n";
    a.sub(b).write(std::cout);
    std::cout << "\n";
}

```

```

a.mul(b).write(std::cout);
std::cout << "\n";
a.div(b).write(std::cout);
std::cout << "\n";
if (a.equ(b) == 1) {
    std::cout << "yes";
} else {
    std::cout << "no";
}
std::cout << std::endl;
a.con1().write(std::cout);
std::cout << "\n";
b.con1().write(std::cout);
std::cout << "\n";
if (a.modul(b) == 0) {
    std::cout << "a = b";
} else if (a.modul(b) == -1) {
    std::cout << "a > b";
} else {
    std::cout << "a < b";
}
std::cout << "\n";

return 0;
}

```

2. Ссылка на репозиторий на GitHub.

https://github.com/keoni02032/oop_exercise_01.git/

3. Набор testcases.

test_01.test:

2 2 2 2

test_02.test:

1 2 3 4

test_03.test:

4 3 2 1

test_04.test:

1 -2 1 -2

test_05.test:

1 -2 1 2

4. Результаты выполнения тестов.

test_01.result:

2 2

2 2

4 4

0 0

0 8

1 0

yes

2 -2

2 -2

$|a| = |b|$

test_02.result:

1 2

3 4

4 6

-2 -2

-5 -10

0.44 0.08

no equal

1 -2

3 -4

$|a| < |b|$

test_03.result:

4 3 2 1

4 3

2 1

6 4

2 2

5 10

2.2 0.4

no equal

4 -3

2 -1

$|a| > |b|$

test_04.result:

1 -2 1 -2

1 -2

1 -2

2 -4

0 0

-3 -4

1 0

yes

. 1 2

1 2

|a| = |b|

test_05.result:

1 -2 1 2

1 -2

1 2

2 0

0 -4

5 0

-0.6 -0.8

no equal

1 2

1 -2

|a| = |b|

5 Объяснение результатов работы программы.

1) При запуске скрипта с аргументом `./ 'test (1). sh' ./lab1` объекты `a`, `b` в основной программе получают данные из файлов `test_??test`. Над этими данными выполняются следующие операции: сложение, вычитание, умножение, деление, сравнение, выводится сопряженное число, реализация операции сравнения модулей.

2) Вывод данных объектов `a`, `b` в стандартный поток вывода.

3) Объекты `a` и `b` складываются с помощью функции-члена `add()` класса `complex`, и результат выводится в стандартный поток вывода с помощью функции `write()`.

4) Из объекта `a` вычитается объект `b` с помощью функции-члена `sub()` класса `complex`, и результат выводится в стандартный поток вывода с помощью функции

write()).

5) Объекты *a* и *b* умножаются с помощью функции-члена *mul()* класса *complex*, затем выводится в стандартный поток вывода с помощью функции *write()*.

6) Объект *a* делится на *b* с помощью функции-члена *div()* класса *complex*, и результат выводится в стандартный поток вывода с помощью функции *write()*.

7) Объекты *a* и *b* проверяются на равенство с помощью функции-члена *eq()* класса *complex*, и результат выводится с помощью стандартного потока вывода.

8) Для объектов *a* и *b* вычисляется сопряженное число с помощью функции-члена *con1*, *con2* класса *complex*, и результат выводится в стандартный поток вывода с помощью функции *write()*.

9) Для объектов *a* и *b* производится сравнение модулей с помощью функции-члена *modl* класса *complex*, и результат выводится с помощью стандартного потока вывода.

6. Вывод.

Выполняя данную лабораторную я получил опыт работы с простыми классами, с системой сборки *Stake*, с системой контроля версий *Git*, а также изучил основы работы с классами в *C++*. Создал класс, соответствующий варианту моего задания, реализовал для него арифметические операции сложения, вычитания, умножения, деления, сравнения, получения сопряженного числа, а также реализовал операцию сравнения модулей.