

Московский Авиационный Институт
(Национальный исследовательский Университет)

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

Лабораторная работа №1
по курсу «ООП»

Тема:
Простые классы.

Студент:	Болдыв А. К.
Группа:	М80-206Б-18
Преподаватель:	Журавлев А. А.
Вариант:	2
Оценка:	
Дата:	

Москва
2019

1. Код программы на языке C++:

complex.hpp:

```
#ifndef __COMPLEX__
#define __COMPLEX__

#include <iostream>

class complex {

public:
    complex();
    complex(double a, double b) : r(a), fi(b) {}

    void read(std::istream&);
    void write(std::ostream&) 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;
    int equ(const complex & o) const;
    complex conj(const complex& o) const;

private:
    double r;
    double fi;
};
#endif
```

complex.cpp:

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

complex::complex() {
    r = 0;
    fi = 0;
}
```

```

complex complex::add(const complex& o) const {
    complex result;
    double cos_1 = cos ( fi * M_PI );
    double cos_2 = cos ( o.fi * M_PI );
    double sin_1 = sin ( fi * M_PI );
    double sin_2 = sin ( o.fi * M_PI );
    double a_1 = r * cos_1;
    double a_2 = o.r * cos_2;
    double b_1 = r * sin_1;
    double b_2 = o.r * sin_2;
    double a_r = a_1 + a_2;
    double b_r = b_1 + b_2;
    result.r = sqrt( a_r * a_r + b_r * b_r );

    //ПЕРВАЯ ЧЕТВЕРТЬ
    if (a_r / result.r > 0 and b_r / result.r >= 0) {
        result.fi = ( acos ( a_r / result.r ) ) / M_PI;
    }
    //ВТОРАЯ
    if (a_r / result.r <= 0 and b_r / result.r > 0) {
        result.fi = ( acos ( a_r / result.r ) ) / M_PI;
    }
    //ТРЕТЬЯ
    if (a_r / result.r >= 0 and b_r / result.r < 0) {
        result.fi = ( asin ( b_r / result.r ) ) / M_PI;
    }
    //ЧЕТВЕРТАЯ
    if (a_r / result.r < 0 and b_r / result.r <= 0) {
        result.fi = - ( acos ( a_r / result.r ) ) / M_PI;
    }

    return result;
}

```

```

complex complex::sub(const complex& o) const {
    complex result;
    double cos_1 = cos ( fi * M_PI );
    double cos_2 = cos ( (o.fi - 1) * M_PI);
    double sin_1 = sin ( fi * M_PI );
    double sin_2 = sin ( (o.fi - 1) * M_PI);
    double a_1 = r * cos_1;
    double a_2 = o.r * cos_2;
    double b_1 = r * sin_1;
    double b_2 = o.r * sin_2;

```

```

double a_r = a_1 + a_2;
double b_r = b_1 + b_2;
result.r = sqrt( a_r * a_r + b_r * b_r );

//ПЕРВАЯ ЧЕТВЕРТЬ
if (a_r / result.r > 0 and b_r /result.r >= 0) {
    result.fi = ( acos ( a_r / result.r) ) / M_PI;
}
//ВТОРАЯ
if (a_r / result.r <= 0 and b_r /result.r > 0) {
    result.fi = ( acos ( a_r / result.r) ) / M_PI;
}
//ТРЕТЬЯ
if (a_r / result.r >= 0 and b_r /result.r < 0) {
    result.fi = ( asin ( b_r / result.r) ) / M_PI;
}
//ЧЕТВЕРТАЯ
if (a_r / result.r < 0 and b_r /result.r <= 0) {
    result.fi = - ( acos ( a_r / result.r) ) / M_PI;
}

return result;
}

```

```

complex complex::mul(const complex& o) const {
    complex result;
    double cos_1 = cos ( fi * M_PI );
    double cos_2 = cos ( o.fi * M_PI );
    double sin_1 = sin ( fi * M_PI );
    double sin_2 = sin ( o.fi * M_PI );
    double a_1 = r * cos_1;
    double a_2 = o.r * cos_2;
    double b_1 = r * sin_1;
    double b_2 = o.r * sin_2;
    double a_r = a_1 + a_2;
    double b_r = b_1 + b_2;
    result.r = r * o.r;
    result.fi = fi + o.fi;

    return result;
}

```

```

complex complex::div(const complex& o) const {
    complex result;

```

```

double cos_1 = cos ( fi * M_PI );
double cos_2 = cos ( o.fi * M_PI );
double sin_1 = sin ( fi * M_PI );
double sin_2 = sin ( o.fi * M_PI );
double a_1 = r * cos_1;
double a_2 = o.r * cos_2;
double b_1 = r * sin_1;
double b_2 = o.r * sin_2;
double a_r = a_1 + a_2;
double b_r = b_1 + b_2;
result.r = r / o.r;
result.fi = fi - o.fi;
return result;
}

```

```

int complex::equ(const complex& o) const {
    if ((r == o.r) && (fi == o.fi)) {
        return 1;
    }
    return 0;
}

```

```

complex complex::conj(const complex& o) const {
    complex result;
    result.r = r;
    result.fi = -fi;
    return result;
}

```

```

void complex::read(std::istream& is) {
    double a, b;
    is >> a;
    is >> b;
    if ( b >= 2 ) {
        b = fmod( b, 2 );
    } else if ( b < 0 ) {
        b = fmod( b, 2 ) + 2;
    }
    this->r = a;
    this->fi = b;
}

```

```

void complex::write(std::ostream& os) const{
    os << this->r << " " << this->fi << "p\n";
}

```

```
}
```

main.cpp:

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

int main() {
    complex a, b, ans;

    a.read(std::cin);
    b.read(std::cin);

    std::cout << "First number : ";
    a.write(std::cout);
    std::cout << "second number : ";
    b.write(std::cout);

    std::cout << "\n +\n =\n";
    ans = a.add(b);
    ans.write(std::cout);

    std::cout << "\n -\n =\n";
    ans = a.sub(b);
    ans.write(std::cout);

    std::cout << "\n *\n =\n";
    ans = a.mul(b);
    ans.write(std::cout);

    std::cout << "\n /\n =\n";
    ans = a.div(b);
    ans.write(std::cout);

    std::cout << "\nconj = ";
    ans = a.conj(a);
    ans.write(std::cout);
    std::cout << "\nconj = ";
    ans = b.conj(b);
    ans.write(std::cout);

    if (a.equ(b) == 1) {
        std::cout << "a = b\n";
    } else {
```

```
    std::cout << "a != b\n";  
}  
  
}
```

CmakeLists.txt:

```
cmake_minimum_required(VERSION 3.1)  
project(lab1)  
  
add_executable(lab1 main.cpp complex.cpp)
```

2. Ссылка на репозиторий на GitHub.

https://github.com/Anton-Boldyrev/oop_exercise_01

3. Набор testcases.

test_01.txt:

```
1 0.5 1 0.5  
First number : 1 0.5p  
second number : 1 0.5p
```

```
+  
=  
2 0.5p
```

```
-  
=  
0 0p
```

*

=

1 1p

/

=

1 0p

conj = 1 -0.5p

conj = 1 -0.5p

a = b

test_02.txt:

1 -0.2 1 0.75

First number : 1 1.8p

second number : 1 0.75p

+

=

0.156918 0.275p

-

=

1.99383 -0.225p

*

=

1 2.55p

/

=

1 1.05p

conj = 1 -1.8p

conj = 1 -0.75p

a != b

test_03.txt:

1 0.3 2 0.75

First number : 1 0.3p

second number : 2 0.75p

+

=

2.37186 0.613285p

-

=

2.09147 -0.0934437p

*

=

2 1.05p

/

=

0.5 -0.45p

conj = 1 -0.3p

conj = 2 -0.75p

a != b

test_04.txt:

2 -3.75 1 -1.95

First number : 2 0.25p

second number : 1 0.05p

+

=

2.86986 0.184341p

-

=

1.32813 0.395932p

*

=

2 0.3p

/

=

2 0.2p

conj = 2 -0.25p

conj = 1 -0.05p
a != b

test_05.txt:

0.5 0.5 0.5 0.5

First number : 0.5 0.5p

second number : 0.5 0.5p

+
=
1 0.5p

-
=
0 0p

*
=
0.25 1p

/
=
1 0p

conj = 0.5 -0.5p

conj = 0.5 -0.5p
a = b

test_06.txt:

0.5 -0.6 2 4.88

First number : 0.5 1.4p

second number : 2 0.88p

+
=
2.03087 0.959023p

-
=
2.09179 -0.196675p

*
=

1 2.28p

/

=

0.25 0.52p

conj = 0.5 -1.4p

conj = 2 -0.88p

a != b

test_07.txt:

1 -0.01 2 -4.95

First number : 1 1.99p

second number : 2 1.05p

+

=

1.03482 -0.892042p

-

=

2.98817 0.0300264p

*

=

2 3.04p

/

=

0.5 0.94p

conj = 1 -1.99p

conj = 2 -1.05p

a != b

5. Объяснение результатов работы программы.

Со стандартного ввода программа считывает два комплексных числа в виде двух пар действительных чисел. Далее преобразует в «стандартный вид», чтобы угол φ был ≥ 0 и $< 2\pi$. Далее вызываются различные классы, в них мы передаем оба числа представленных `double` парами. Сами методы, такие как `Add()`, `Mul()`, `Div()`, `Sub()`, реализованы согласно правилам арифметики. Преобразования происходят согласно правилам операций с комплексными числами представленными в тригонометрической форме. После завершения работы программы на стандартный вывод подается результат выполнения арифметических операций, сопряженные числа и проверка на их равенство.

6. Вывод.

Выполняя данную лабораторную я получил опыт работы с простыми классами, с системой сборки `Cmake`, с системой контроля версий `GitHub`, а также изучил основы работы с классами в `C++`. Создал класс, соответствующий варианту моего задания, реализовал для него арифметические операции сложения, вычитания, умножения, деления, а также операции сравнения.