

Отчёт по лабораторной работе № 01 по курсу 2

студента группы М80-208Б-18, № по списку 2

Адреса www, e-mail, jabber, skype alek.maria@yandex.ru

Работа выполнена: "29" сентября 2019г.

1. Тема:

Простые
классы

2. Цель работы: Изучение системы сборки на языке C++, изучение систем контроля версии. Изучение основ работы с классами в C++.

3. Задание (вариант № 2):

Комплексное число в тригонометрической форме представляются парой действительных чисел (r, φ) , где r – радиус (модуль), φ – угол. Реализовать класс **Complex** для работы с комплексными числами. Обязательно должны быть присутствовать операции

- сложения **add**, $(r_1, \varphi_1) + (r_2, \varphi_2)$;
- вычитания **sub**, $(r_1, \varphi_1) - (r_2, \varphi_2)$;
- умножения **mul**, $(r_1, \varphi_1) * (r_2, \varphi_2)$;
- деления **div**, $(r_1, \varphi_1) / (r_2, \varphi_2)$;
- операции сравнения **equ**, $(r_1, \varphi_1) = (r_2, \varphi_2)$, если $(r_1 = r_2)$ и $(\varphi_1 = \varphi_2)$;
- сопряженное число **conj**, **conj** $(r, \varphi) = (r, -\varphi)$.

4. Адрес репозитория на GitHub https://github.com/PowerMasha/oop_exercise_01

5. Код программы на C++

main.cpp

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

int main(){
    complex m1;
    complex m2;
    printf("Введите первое комплексное число\n");
    for(int i = 0; i < 2; ++i){
        m1.read(i);
    }
    printf("Введите второе комплексное число\n");
    for(int i = 0; i < 2; ++i){
        m2.read(i);
    }
    printf("Первое комплексное число, модуль длины (r) угол (u)\n");
    for(int i = 0; i < 2; ++i){
        std::cout << m1.get(i)<<' ';
    }
    std::cout << "\n";
    printf("Второе комплексное число, модуль длины (r) угол (u)\n");
    for(int i = 0; i < 2; ++i){
        std::cout << m2.get(i)<<' ';
    }
    std::cout << "\n";
    //SUMMA
    complex res1(0,0);
```

```

complex res2(0,0);
double X1=m1.per(1);
double X2=m2.per(1);
double Y1=m1.pir(1);
double Y2=m2.pir(1);
if(X1!=0 && Y1!=0){res1.set(m1.get(0)/X1,m1.get(0)/Y1);} else{
    if(X1==0){res1.set(0,m1.get(0)/Y1);} else {
        res1.set(m1.get(0)/X1,0);
    }
}
if(X2!=0 && Y2!=0){res2.set(m2.get(0)/X2,m2.get(0)/Y2);} else{
    if(X2==0){res2.set(0,m2.get(0)/Y2);} else {
        res2.set(m2.get(0)/X2,0);
    }
}
complex sum = res1.add(res2);
std::cout << "\n";
std::cout << "Координаты вектора суммы:\n";
for(int i = 0; i < 2; ++i){
    std::cout << sum.get(i) << ' ';
}
std::cout << "\n";
//Разность
complex sub = res1.sub(res2);
std::cout << "\n";

std::cout << "Координаты вектора разности:\n";
for(int i = 0; i < 2; ++i){
    std::cout.setf(std::ios::fixed);
    std::cout.precision(2);
    std::cout << sub.get(i) << ' ';
}
std::cout << "\n";

    complex product = m1.multiply(m2);
std::cout << "Произведение: \n";
for(int i = 0; i < 2; ++i){
    if(i==0){std::cout << product.get(i) << "*";}
    if(i==1){std::cout << "(cos("<< product.get(i) << ") + i*sin("<< product.get(i) << "))";}
}

std::cout << "\n";

complex del = m1.div(m2);
std::cout << "Деление : \n";
for(int i = 0; i < 2; ++i){
    if(i==0){std::cout << del.get(i) << "*";}
    if(i==1){std::cout << "(cos("<< del.get(i) << ") + i*sin("<< del.get(i) << "))";}
}

std::cout << "\n";
std::cout << "Сравнение комплексных чисел по длине вектора и углу:\n";
int k=0;
int l=0;
for(int i = 0; i < 2; ++i){
    if (m1.get(i)==m2.get(i) && i==0){k=1;} else {k=0;}
    if (m1.get(i)==m2.get(i) && i==1){l=1;} else {l=0;}
}

if(k==1){std::cout << "Длины равны\n";} else {std::cout << "Длины не равны\n";}
if(l==1){std::cout << "Углы равны\n";} else {std::cout << "Углы не равны\n";}

```

```

std::cout << '\n';
complex res3(0,0);
complex res4(0,0);
res3.set(m1.get(0),(-1)*m1.get(1));
res4.set(m2.get(0),(-1)*m2.get(1));
std::cout << "sopr_m1:\n";
for(int i = 0; i < 2; ++i){
    std::cout << res3.get(i) << ' ';
}
std::cout << '\n';
std::cout << "sopr_m2:\n";
for(int i = 0; i < 2; ++i){
    std::cout << res4.get(i) << ' ';
}
std::cout << '\n';
}

```

complex.h

```

#ifndef D_COMPLEX_H
#define D_COMPLEX_H
#include <iostream>
#include <cmath>

struct complex {
double a;
double b;
complex(double a,double b);
complex ();
    complex add(const complex& rhs) const;
    complex multiply(const complex& rhs) const;
    complex sub(const complex& rhs) const;
    complex div(const complex& rhs) const;

double get(int i);
double per(int i);
double pir(int i);
void set(double x,double y);
void read(int i);

private:
    double arr[2];
};
#endif // D_COMPLEX_H

```

complex.cpp

```

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

complex::complex(): arr{0,0} {}

complex::complex(double a,double b): arr{a, b} {}
double PI=3.14159;
double complex::get(int i) {
    return arr[i];
}
void complex::set(double x,double y) {arr[0]=x;arr[1]=y;}
void complex::read(int i){std::cin>>arr[i];}

```

```

double complex::per(int i){
double k=cos(arr[i]*PI/180);
if ((arr[i]==90)|| (arr[i]==270)){k=0;}
return k;
}
double complex::pir(int i){
double s=sin(arr[i]*PI/180);
if ((arr[i]==0)|| (arr[i]==180)){s=0;}
return s;
}
complex complex::add(const complex& rhs) const{
complex sum{0,0};
for(int i = 0; i < 2; ++i){
    sum.arr[i]+=arr[i]+rhs.arr[i];
}
return sum;
}

complex complex::sub(const complex& rhs) const{
complex raznost{0,0};
for(int i = 0; i < 2; ++i){
    raznost.arr[i]=arr[i]-rhs.arr[i];
}
return raznost;
}
complex complex::multiply(const complex& rhs) const {
complex result{0,0};
for(int i = 0; i < 2; ++i){
    if (i==0) {result.arr[i] += arr[i]*rhs.arr[i];}
    if (i==1){result.arr[i] += arr[i]+ rhs.arr[i];}
}
return result;
}

complex complex::div(const complex& rhs) const {
complex result{0,0};
for(int i = 0; i < 2; ++i){
    if (i==0 && rhs.arr[i]!=0) {result.arr[i] += arr[i] /rhs.arr[i];}
    if (i==1){result.arr[i] += arr[i] - rhs.arr[i];}
}
return result;
}

```

CMakeLists.txt

project(1lab)

add_executable(oop_exercise_01
 main.cpp
 complex.cpp)

set(CMAKE_CXX_FLAGS
 "\${CMAKE_CXX_FLAGS} -Wall -Wextra")

6. Набор testcases

test_01.txt	Ожидаемое действие	Ожидаемый результат
1 30 3.4 45	add((1, 30) ,(3.4, 45))	5.96 6.81

		sub((1, 30) ,(3.4, 45))	-3.65 -2.80
		multiply((1, 30) ,(3.4, 45))	3.4*(cos(75)+i*sin(75))
		div((1, 30) ,(3.4, 45))	2.00*(cos(60.00)+i*sin(60.00))
		sravn((1, 30) ,(3.4, 45))	Длины не равны Углы не равны
		Сопряженные числа	(1, -30) (3.4, -45)
test_02.txt	Ожидаемое действие		Ожидаемый результат
2 90 1 30	add((2,90),(1,30))		1.15 4
	sub((2,90),(1,30))		-1.15 0.00
	multiply ((2,90),(1,30))		2*(cos(120)+i*sin(120))
	div((2,90),(1,30))		2*(cos(60)+i*sin(60))
	sravn((2,90),(1,30))		Длины не равны Углы не равны
	Сопряженные числа		(2, -90), (1, -30)
test_03.txt	Ожидаемое действие		Ожидаемый результат
5 45 3 180	add((5,45),(3,180))		4.07 7.07
	sub((5,45),(3,180))		10.07 7.07
	multiply ((5,45),(3,180))		15*(cos(225)+i*sin(225))

$\text{div}((5,45),(3,180))$

$2*(\cos(-135)+i*\sin(-135))$

$\text{sra}((5,45),(3,180))$

Длины не равны
Углы не равны

Сопряженные числа

$(5, -45) (3, -180)$

7. Результаты выполнения тестов

```
masha@masha-VirtualBox:~/2kurs/oop_exercise_01/tmp$ ./oop_exercise_01 <
~/2kurs/oop_exercise_01/test_01.txt
```

Введите первое комплексное число

Введите второе комплексное число

Первое комплексное число, модуль длины (r) угол (u)

1 30

Второе комплексное число, модуль длины (r) угол (u)

3.4 45

Координаты вектора суммы:

5.96 6.81

Координаты вектора разности:

-3.65 -2.81

Произведение:

$3*(\cos(75)+i*\sin(75))$

Деление :

$0*(\cos(-15)+i*\sin(-15))$

Сравнение комплексных чисел по длине вектора и углу:

Длины не равны

Углы не равны

sopr_m1:

1 -30

sopr_m2:

3 -45

```
masha@masha-VirtualBox:~/2kurs/oop_exercise_01/tmp$ ./oop_exercise_01 <
~/2kurs/oop_exercise_01/test_02.txt
```

Введите первое комплексное число

Введите второе комплексное число

Первое комплексное число, модуль длины (r) угол (u)

2 90

Второе комплексное число, модуль длины (r) угол (u)

1 30

Координаты вектора суммы:

1.15 4

Координаты вектора разности:

-1.15 0.00

Произведение:

$2*(\cos(120)+i*\sin(120))$

Деление :

$2*(\cos(60)+i*\sin(60))$

Сравнение комплексных чисел по длине вектора и углу:

Длины не равны

Углы не равны

sopr_m1:

2 -90

```
sopr_m2:
1 -30
masha@masha-VirtualBox:~/2kurs/oop_exercise_01/tmp$ ./oop_exercise_01 <
~/2kurs/oop_exercise_01/test_03.txt
Введите первое комплексное число
Введите второе комплексное число
Первое комплексное число, модуль длины (r)  угол (u)
5 45
Второе комплексное число, модуль длины (r)  угол (u)
3 180
```

```
Координаты вектора суммы:
4.07 7.07
```

```
Координаты вектора разности:
10.07 7.07
Произведение:
15*(cos(225)+i*sin(225))
Деление :
2*(cos(-135)+i*sin(-135))
Сравнение комплексных чисел по длине вектора и углу:
Длины не равны
Углы не равны
```

```
sopr_m1:
5 -45
sopr_m2:
3 -180
```

8. Объяснение результатов работы программы - вывод

В `complex.h` были заданы методы и свойства этого класса, а в `fractions.cpp` они были описаны. Описанные методы использовались в файле `main.cpp`.

Классы, описывают методы и свойства объектов, позволяют работать с этими объектами, не вдаваясь в подробности их реализации, что является примером абстракции данных. Такой подход незаменим при работе в групповых проектах.