**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Батова Екатерина Дмитриевна

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

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

- сложения add, (*r*1, j1) + (*r*2, j2);

- вычитания sub, (*r*1, j1) – (*r*2, j2);

- умножения mul, (*r*1, j1) ´ (*r*2, j2);

- деления div, (*r*1, j1) / (*r*2, j2) ;

- сравнение equ, (*r*1, j1) = (*r*2, j2), если (*r*1 = *r*2) и (j1 = j2); - сопряженное число conj, conj(*r*, j) = (*r*, –j).

Реализовать операции сравнения по действительной части.

Операции сложения, вычитания, умножения, деления, сравнения на равенство должны быть выполнены в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа **Complex**.

1. Описание программы

Программа состоит из программ, которые помогают выполнить операции над классом Complex(main.cpp, OOP2\_2.cpp, OOP2\_2.h). Операции сложения, вычитания, умножения, деления, сравнения на равенство реализованы в виде перегрузки операторов.Также реализован ввод и вывод комплексных чисел. Для удобства пользования было создано меню, которое состоит из 8 пунктов (операции, выход и само меню). Пользовательский литерал выводится единожды в начале программы. Программа использует консольный ввод данных.

1. Набор тестов

test1

1 //ввод команды

1 2 //ввод комплексного числа

2 3 //ввод комплексного числа

2 //ввод команды

4.5 6 //ввод комплексного числа

8 1 //ввод комплексного числа

3 //ввод команды

5 6 //ввод комплексного числа

3.75 5 //ввод комплексного числа

4 //ввод команды

6 5 //ввод комплексного числа

5 7 //ввод комплексного числа

5 //ввод команды

7 8 //ввод комплексного числа

6 //ввод команды

9 4 //ввод комплексного числа

5 7 //ввод комплексного числа

0 //ввод команды

test2

1 //ввод команды

13 0.5 //ввод комплексного числа

2 4 //ввод комплексного числа

2 //ввод команды

-4.5 -6 //ввод комплексного числа

-8 -10 //ввод комплексного числа

3 //ввод команды

6 9 //ввод комплексного числа

3.5 5 //ввод комплексного числа

4 //ввод команды

5 9 //ввод комплексного числа

5 7 //ввод комплексного числа

5 //ввод команды

7 -9 //ввод комплексного числа

6 //ввод команды

6 6 //ввод комплексного числа

6 6 //ввод комплексного числа

0 //ввод команды

1. Результаты выполнения тестов

test1

costom literal(4 degrees to radians):0.0698132

0.exit

1.addition

2.subtraction

3.multiplication

4.division

5.conjugate number

6.equal

7.menu

enter a number for action:

1

enter a radius:

1

enter a angle (degrees):

2

enter a radius:

2

enter a angle (degrees):

3

3 \* (cos5+i\*sin5)

Enter a number for action:

2

enter a radius:

4.5

enter a angle (degrees):

6

enter a radius:

8

enter a angle (degrees):

1

-3.5 \* (cos5+i\*sin5)

Enter a number for action:

3

enter a radius:

5

enter a angle (degrees):

6

enter a radius:

3.75

enter a angle (degrees):

5

-11.25 \* (cos47.5+i\*sin47.5)

Enter a number for action:

4

enter a radius:

6

enter a angle (degrees):

5

enter a radius:

5

enter a angle (degrees):

7

0.878378 \* (cos-0.22973+i\*sin-0.22973)

Enter a number for action:

5

enter a radius:

7

enter a angle (degrees):

8

7 \* (cos-8+i\*sin-8)

Enter a number for action:

6

enter a radius:

9

enter a angle (degrees):

4

enter a radius:

5

enter a angle (degrees):

7

not equal

Enter a number for action:

0

test2

costom literal(4 degrees to radians):0.0698132

0.exit

1.addition

2.subtraction

3.multiplication

4.division

5.conjugate number

6.equal

7.menu

enter a number for action:

1

enter a radius:

13

enter a angle (degrees):

0.5

enter a radius:

2

enter a angle (degrees):

4

15 \* (cos4.5+i\*sin4.5)

Enter a number for action:

2

enter a radius:

-4.5

enter a angle (degrees):

-6

enter a radius:

-8

enter a angle (degrees):

-10

3.5 \* (cos4+i\*sin4)

Enter a number for action:

3

enter a radius:

6

enter a angle (degrees):

9

enter a radius:

3.5

enter a angle (degrees):

5

-24 \* (cos61.5+i\*sin61.5)

Enter a number for action:

4

enter a radius:

5

enter a angle (degrees):

9

enter a radius:

5

enter a angle (degrees):

7

1.18919 \* (cos0.135135+i\*sin0.135135)

Enter a number for action:

5

enter a radius:

7

enter a angle (degrees):

-9

7 \* (cos9+i\*sin9)

Enter a number for action:

6

enter a radius:

6

enter a angle (degrees):

6

enter a radius:

6

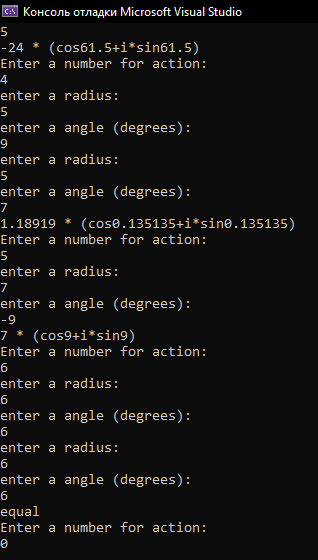
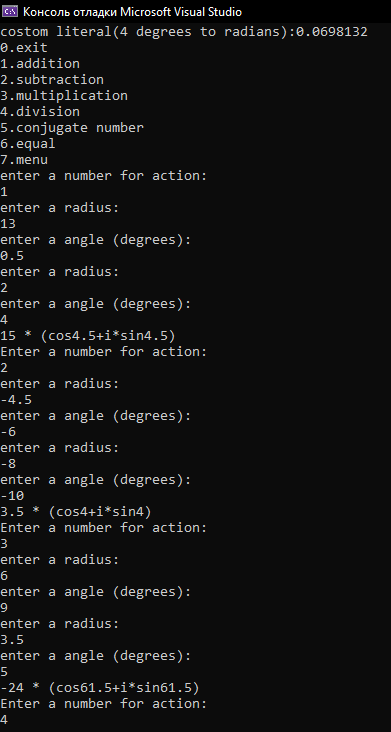
enter a angle (degrees):

6

equal

Enter a number for action:

0



1. Ссылка на github:

<https://github.com/e-k-a/oop_exercise_02>

1. Листинг программы

main.cpp

//М8О-201Б-19 Батова Екатерина 2 вариант

/\*Комплексное число в тригонометрической форме представляются

парой действительных чисел (r, j), где r – радиус (модуль), j – угол.

Реализовать класс Complex для работы с комплексными числами.

Обязательно должны быть присутствовать операции

- сложения add, (r1, j1) + (r2, j2);

- вычитания sub, (r1, j1) – (r2, j2);

- умножения mul, (r1, j1) ´ (r2, j2);

- деления div, (r1, j1) / (r2, j2) ;

- сравнение equ, (r1, j1) = (r2, j2), если (r1 = r2) и (j1 = j2);

- сопряженное число conj, conj(r, j) = (r, –j).

Реализовать операции сравнения по действительной части.

Операции сложения, вычитания, умножения, деления, сравнения на равенство должны быть выполнены в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа Complex.\*/

#define \_USE\_MATH\_DEFINES // используется длдя числа Пи

#include <cmath>

#include <iostream>

#include "OOP2\_2.h"

using namespace std;

long double operator""\_rad(long double deg) {//пользовательский литерал, переводящий градусы в радианы

return deg \* M\_PI / 180.0;

}

int main() {

Complex ch1, ch2, variable;

bool action = true;

int counter;

std::cout << "costom literal(4 degrees to radians):" << 4.0\_rad << std::endl;

std::cout << "0.exit\n1.addition\n2.subtraction\n3.multiplication" << std::endl;

std::cout << "4.division\n5.conjugate number\n6.equal\n7.menu" << std::endl;

std::cout << "enter a number for action:" << std::endl;

std::cin >> counter;

while (action) {

switch (counter) {

case 0:

action = false;

break;

case 1: //сложение

ch1.input();

ch2.input();

variable = ch1 + ch2;

variable.output(std::cout);

break;

case 2: //вычитание

ch1.input();

ch2.input();

variable = ch1 - ch2;

variable.output(std::cout);

break;

case 3: //умножение

ch1.input();

ch2.input();

variable = ch1 \* ch2;

variable.output(std::cout);

break;

case 4: //деление

ch1.input();

ch2.input();

variable = ch1 / ch2;

variable.output(std::cout);

break;

case 5: //спряженное число

ch1.input();

variable = --ch1;

variable.output(std::cout);

break;

case 6: //сравнение

ch1.input();

ch2.input();

if (ch1 == ch2) { std::cout << "equal" << std::endl; }

else { std::cout << "not equal" << std::endl; }

break;

case 7: //меню

std::cout << "0.exit\n1.addition\n2.subtraction\n3.multiplication\n" << std::endl;

std::cout << "4.division\n5.conjugate number\n6.equal\n7.equal modulus\n8.menu\n" << std::endl;

break;

default:

std::cout << "Oops, that input is invalid. Please try again.\n" << std::endl;

break;

}

if (action) {

std::cout << "Enter a number for action:" << std::endl;

std::cin >> counter;

}

}

}

OOP2\_2.cpp

//М8О-201Б-19 Батова Екатерина 2 вариант

#include <iostream>

#include <math.h>

#include "OOP2\_2.h"

using namespace std;

Complex::Complex() { //инициализация по умолчанию

r = 0;

j = 0;

}

Complex::Complex(double a, double b) {//инициализация

r = a;

j = b;

if (j >= 360) { j = j - 360; }

}

Complex Complex::operator+(Complex add) {//перегрузка оператора сложения

Complex c;

c.r = this->r + add.r;

c.j = this->j + add.j;

c = Complex(c.r, c.j);

return c;

}

Complex Complex::operator-(Complex sub) {//перегрузка оператора сложения вычитания

Complex c;

c.r = this->r - sub.r;

c.j = this->j - sub.j;

c = Complex(c.r, c.j);

return c;

}

Complex Complex::operator\*(Complex mul) {//перегрузка оператора умножения

Complex c;

c.r = this->r \* mul.r - this->j \* mul.j;

c.j = this->r \* mul.j + this->j \* mul.r;

c = Complex(c.r, c.j);

return c;

}

Complex Complex::operator/(Complex div) {//перегрузка оператора деления

Complex c;

if (div.r == 0 && div.j == 0) {

std::cout << "invalid numbers!" << std::endl;

return Complex();

}

else {

c.r = (this->r \* div.r + this->j \* div.j) / (div.r \* div.r + div.j \* div.j);

c.j = (this->j \* div.r - this->r \* div.j) / (div.r\*div.r + div.j \*div.j);

c = Complex(c.r, c.j);

return c;

}

}

bool Complex::operator==(Complex const equ) {//перегрузка оператора сравнения

return (r == equ.r && j == equ.j);

}

Complex Complex::operator--() {//перегрузка оператора сопряженного числа

Complex c;

c.r = r;

if (j == 0) { c.j = j;}

else { c.j = -j; }

return c;

}

void Complex::input() { //ввод комплекного числа

std::cout << "enter a radius:" << std::endl;

cin >> r;

std::cout << "enter a angle (degrees):" << std::endl;

cin >> j;

}

void Complex::output(std::ostream& ch) {//вывод комплексного числа

ch << r << " \* " << "(cos" << j << "+i\*sin" << j << ")" << std::endl;

}

OOP2\_2.cpp

//М8О-201Б-19 Батова Екатерина 2 вариант

class Complex {

private:

double r, j;

public:

Complex();

Complex(double r, double j);

Complex operator+(Complex add);

Complex operator-(Complex sub);

Complex operator\*(Complex mul);

Complex operator/(Complex div);

bool operator==(Complex const equ);

Complex operator--();

void input();

void output(std::ostream& ch);

};

1. Вывод

В данной лабораторной работе были разобрабраны пользовательские литералы, а также перегрузка операторов.