**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: Шавандрин Федор Михайлович

Группа: 80-208

Преподаватель: Чернышов Л.Н.

Дата: 11.10.2020

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**

I. Класс Complex:

а) Конструктор;

б) Перегруженные операторы сравнения;

в) Сеттеры;

г) Геттеры.

II. а) Функции проверки угла, сопряженного числа;

б) Перегруженные арифметические операторы;

в) Пользовательский литерал;

г) Перегрузка операторов ввода и вывода;

д) Функции для суммы, разности, умножения и деления двух комплексных чисел.

III. Главная функция:

а) Вывод меню;

б) Выбор пункта меню;

в) Ввод данных;

г) Подсчет результатов с помощью функций;

д) Вывод результата.

1. **Набор тестов**

Строки в таблице обозначают первое и второе комплексные числа, с которыми можно произвести следующие действия: сложить, вычесть, умножить, разделить, сравнить, найти сопряженное число.

Комплексное число представляет собой скобку (r,p), где r - радиус, а p - угол.

Таблица 1. Тестовые данные

|  |  |
| --- | --- |
| Первое комплексное число | Второе комплексное число |
| (14,50) | (78,64) |
| (2,198.7) | (55.7,46) |

1. **Результаты выполнения тестов**

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 1

14 50 78 64

(14,50) + (78,64) = (81.1088,0.996316)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 2

14 50 78 64

(14,50) - (78,64) = (78.6506,1.17784)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 3

14 50 78 64

(14,50) \* (78,64) = (1092,114)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 4

14 50 78 64

(14,50) / (78,64) = (0.179487,-14)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 5

14 50 78 64

(14,50) < (78,64) (lesser)

(14,50) <= (78,64) (not greater)

(14,50) != (78,64) (not equal)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 6

14 50 78 64

Conjugate number for (14,50) is (14,-50)

Conjugate number for (78,64) is (78,-64)

-----------------------------------------------------------------------------------------

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 1

2 198.7 55.7 46

(2,198.7) + (55.7,46) = (55.0791,2.05203)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 2

2 198.7 55.7 46

(2,198.7) - (55.7,46) = (56.6998,2.01802)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 3

2 198.7 55.7 46

(2,198.7) \* (55.7,46) = (111.4,244.7)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 4

2 198.7 55.7 46

(2,198.7) / (55.7,46) = (0.0359066,152.7)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 5

2 198.7 55.7 46

(2,198.7) >= (55.7,46) (not lesser)

(2,198.7) > (55.7,46) (greater)

(2,198.7) != (55.7,46) (not equal)

Menu:

1. Sum two complex numbers.

2. Subtract two complex numbers.

3. Multiply two complex numbers.

4. Divide two complex numbers.

5. Compare two complex numbers.

6. Print conjugate for complex number.

7. Quit.

Enter the menu item -> 6

2 198.7 55.7 46

Conjugate number for (2,198.7) is (2,-198.7)

Conjugate number for (55.7,46) is (55.7,-46)

1. **Листинг программы**

/\*

Created by Шавандрин Фёдор

Ссылка на репозиторий https://github.com/Adi6e/oop\_exercise\_02

Реализовать класс Complex для работы с комплексными числами. Реализовать арифметические операции и 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.

\*/

#include <iostream>

#include <cmath>

#include <cstdlib>

#include <sstream>

const double MAX\_AVAILABLE\_ANGLE = 2 \* M\_PI;

class Complex{

public:

Complex(){

r = 1;

phi = 90;

}

Complex(double newr, double newphi){

r = newr;

phi = newphi;

}

bool operator==(Complex &other){

return (this->r == other.r) && (this->phi == other.phi);

}

bool operator!=(Complex &other){

return !((this->r == other.r) && (this->phi == other.phi));

}

bool operator>(Complex &other){

return (this->r \* cos(this->phi)) > (other.r \* cos(other.phi));

}

bool operator>=(Complex &other){

return (this->r \* cos(this->phi)) >= (other.r \* cos(other.phi));

}

bool operator<(Complex &other){

return (this->r \* cos(this->phi)) < (other.r \* cos(other.phi));

}

bool operator<=(Complex &other){

return (this->r \* cos(this->phi)) <= (other.r \* cos(other.phi));

}

void SetR(double radius){

if (radius < 0){

std::cout << "Radius can't be lesser than zero!" << std::endl;

return;

}

this->r = radius;

}

void SetAngle(double angle){

this->phi = angle;

}

double GetR() const{

return this->r;

}

double GetAngle() const{

return this->phi;

}

friend std::istream& operator>>(std::istream &in, Complex &complex);

private:

double r;

double phi;

};

void Check\_bounds(double angle){

if (angle > MAX\_AVAILABLE\_ANGLE){

angle -= MAX\_AVAILABLE\_ANGLE;

}

if (angle < MAX\_AVAILABLE\_ANGLE){

angle += MAX\_AVAILABLE\_ANGLE;

}

}

Complex Conj(Complex &x){

double res = x.GetAngle();

res = -res;

x.SetAngle(res);

return x;

}

Complex operator+(Complex &one, Complex &other){

Complex res;

double fpart = one.GetR() \* cos(one.GetAngle()) + other.GetR() \* cos(other.GetAngle());

double spart = one.GetR() \* sin(one.GetAngle()) + other.GetR() \* sin(other.GetAngle());

res.SetR(sqrt(fpart \* fpart + spart \* spart));

res.SetAngle(acos(fpart / res.GetR()));

return res;

}

Complex operator-(Complex &one, Complex &other){

Complex res;

double newangle = other.GetAngle() + M\_PI;

Check\_bounds(newangle);

return res + other;

}

Complex operator\*(Complex &one, Complex &other){

Complex res;

res.SetR(one.GetR() \* other.GetR());

res.SetAngle(one.GetAngle() + other.GetAngle());

Check\_bounds(res.GetAngle());

return res;

}

Complex operator/(Complex &one, Complex &other){

Complex res;

res.SetR(one.GetR() / other.GetR());

res.SetAngle(one.GetAngle() - other.GetAngle());

Check\_bounds(res.GetAngle());

return res;

}

const Complex operator"" \_complex(const char \*str, size\_t size){

std::stringstream ss;

double one, two;

ss.str(str);

ss.ignore(1);

ss >> one;

ss.ignore(1);

ss >> two;

return {one, two};

}

std::istream& operator>>(std::istream &in, Complex &complex){

in >> complex.r;

in >> complex.phi;

return in;

}

std::ostream& operator<<(std::ostream &out, const Complex &x){

out << "(" << x.GetR() << "," << x.GetAngle() << ")";

return out;

}

Complex Sum(Complex &a, Complex &b){

return a + b;

}

Complex Subtract(Complex &a, Complex &b){

return a - b;

}

Complex Multy(Complex &a, Complex &b){

return a \* b;

}

Complex Divide(Complex &a, Complex &b){

return a / b;

}

int main(){

Complex example = “(14,36)”\_complex;

Complex a,b;

int choice;

bool work = true;

while(work){

std::cout << "Menu:" << std::endl << "1. Sum two complex numbers." << std::endl << "2. Subtract two complex numbers." << std::endl << "3. Multiply two complex numbers." << std::endl << "4. Divide two complex numbers." << std::endl << "5. Compare two complex numbers." << std::endl << "6. Print conjugate for complex number." << std::endl << "7. Quit." << std::endl << "Enter the menu item -> ";

std::cin >> choice;

switch (choice){

case 1:

std::cin >> a >> b;

std::cout << a << " + " << b << " = " << Sum(a,b) << std::endl << std::endl;

break;

case 2:

std::cin >> a >> b;

std::cout << a << " - " << b << " = " << Subtract(a,b) << std::endl << std::endl;

break;

case 3:

std::cin >> a >> b;

std::cout << a << " \* " << b << " = " << Multy(a,b) << std::endl << std::endl;

break;

case 4:

std::cin >> a >> b;

std::cout << a << " / " << b << " = " << Divide(a,b) << std::endl << std::endl;

break;

case 5:

std::cin >> a >> b;

if (a < b){

std::cout << a << " < " << b << " (lesser)" << std::endl;

}

else{

std::cout << a << " >= " << b << " (not lesser)" << std::endl;

}

if (a > b){

std::cout << a << " > " << b << " (greater)" << std::endl;

}

else{

std::cout << a << " <= " << b << " (not greater)" << std::endl;

}

if (a == b){

std::cout << a << " = " << b << " (equal)" << std::endl;

}

else{

std::cout << a << " != " << b << " (not equal)" << std::endl;

}

std::cout << std::endl;

break;

case 6:

std::cin >> a >> b;

std::cout << "Conjugate number for " << a << " is " << Conj(a) << std::endl;

std::cout << "Conjugate number for " << b << " is " << Conj(b) << std::endl << std::endl;

break;

case 7:

work = false;

break;

default:

std::cout << "Wrong menu item!" << std::endl << std::endl;

work = false;

break;

}

return 0;

}

1. **Выводы**

Данная лабораторная работа направлена на изучение перегрузки операторов и пользовательского литерала в С++. Были приобретены навыки работы с пользовательским литералом, изучены его особенности, а также закреплены знания о работе с перегрузкой арифметических операторов, а также операторов сравнения и ввода/вывода.

1. **Список литературы**

1. Перегрузка операторов ввода/вывода [Электронный ресурс] URL: <https://ravesli.com/urok-133-peregruzka-operatorov-vvoda-i-vyvoda/> (дата обращения 07.10.2020).

2. Пользовательский литерал [Электронный ресурс] URL: <https://www.pvsm.ru/programmirovanie/3924> (дата обращения 07.10.2020).

3. Представление комплексных чисел в тригонометрической форме [Электронный ресурс] URL: <https://www.webmath.ru/poleznoe/formules_16_6.php> (дата обращения 06.10.2020).