**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Гребенков Дмитрий Игоревич

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

**Комплексное число в алгебраической форме** представляются парой действительных чисел (*a*, *b*), где *a* – действительная часть, *b* – мнимая часть. Реализовать класс Complex для работы с комплексными числами.

Обязательно должны быть присутствовать операции

- сложения add, (*a*, *b*) + (*c*, *d*) = (*a* + *c*, *b* + *d*);

- вычитания sub, (*a*, *b*) – (*c*, *d*) = (*a* – *c*, *b* – *d*);

- умножения mul, (*a*, *b*) ´ (*c*, *d*) = (*ac* – *bd*, *ad* + *bc*);

- деления div, (*a*, *b*) / (*c*, *d*) = (*ac* + *bd*, *bc* – *ad*) / (*c*2 + *d*2);

- сравнение equ, (*a*, *b*) = (*c*, *d*), если (*a* = *c*) и (*b* = *d*);

- сопряженное число conj, conj(*a*, *b*) = (*a*, –*b*).

Реализовать операции сравнения модулей.

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

Реализован класс Complex, он содержит две переменные, представляющие действительную и мнимую части числа, а также методы, необходимые для выдачи текущих значений этих переменных, методы, позволяющие устанавливать новые значения этих переменных и методы, отвечающие за выполнение стандартных операций (сложения, вычитания, умножения, деления, сравнения, нахождения сопряженного числа и сравнения по модулю).

Добавлена перегрузка операторов сложения, вычитания, умножения, деления и сравнения для правильного взаимодействия с комплексными числами.

Взаимодействие с пользователем осуществляется при помощи меню.

1. Набор тестов и результат их выполнения

Демонстрация работы основных функций:

Menu:

0. Print menu.

1. Enter the first complex number.

2. Enter the second complex number.

3. Print the first and second numbers.

4. Addition of the first and second numbers.

5. Subtract the second number from the first.

6. Multiply the first number by the second.

7. Divide the first number by the second.

8. Compare the first number with the second.

9. Print the conjugate of the first number.

10. Compare the modules of the first and second numbers.

11. Quit the program.

Choose option: 1

Enter real part: 2

Enter imaginary part: 3

Choose option: 2

Enter real part: 2

Enter imaginary part: 3

Choose option: 3

First number: (2, 3).

Second number: (2, 3).

Choose option: 4

Result: (4, 6).

Choose option: 5

Result: (0, 0).

Choose option: 6

Result: (-5, 12).

Choose option: 7

Result: (1, 0).

Choose option: 8

Numbers are equal.

Choose option: 9

Result: (2, -3).

Choose option: 10

Numbers are equal in absolute value.

Choose option: 1

Enter real part: 2

Enter imaginary part: 2

Choose option: 2

Enter real part: 1

Enter imaginary part: 1

Choose option: 3

First number: (2, 2).

Second number: (1, 1).

Choose option: 4

Result: (3, 3).

Choose option: 5

Result: (1, 1).

Choose option: 6

Result: (0, 4).

Choose option: 7

Result: (2, 0).

Choose option: 8

Numbers are not equal.

Choose option: 9

Result: (2, -2).

Choose option: 10

Numbers are not equal in absolute value.

Choose option: 1

Enter real part: 4

Enter imaginary part: 3

Choose option: 2

Enter real part: 5

Enter imaginary part: 0

Choose option: 3

First number: (4, 3).

Second number: (5, 0).

Choose option: 4

Result: (9, 3).

Choose option: 5

Result: (-1, 3).

Choose option: 6

Result: (20, 15).

Choose option: 7

Result: (0.8, 0.6).

Choose option: 8

Numbers are not equal.

Choose option: 9

Result: (4, -3).

Choose option: 10

Numbers are equal in absolute value.

Choose option: 11

Некорректные данные:

Menu:

0. Print menu.

1. Enter the first complex number.

2. Enter the second complex number.

3. Print the first and second numbers.

4. Addition of the first and second numbers.

5. Subtract the second number from the first.

6. Multiply the first number by the second.

7. Divide the first number by the second.

8. Compare the first number with the second.

9. Print the conjugate of the first number.

10. Compare the modules of the first and second numbers.

11. Quit the program.

Choose option: 0

Menu:

0. Print menu.

1. Enter the first complex number.

2. Enter the second complex number.

3. Print the first and second numbers.

4. Addition of the first and second numbers.

5. Subtract the second number from the first.

6. Multiply the first number by the second.

7. Divide the first number by the second.

8. Compare the first number with the second.

9. Print the conjugate of the first number.

10. Compare the modules of the first and second numbers.

11. Quit the program.

Choose option: 3

Error: first number not entered.

Error: second number not entered.

Choose option: 4

Error: first number not entered.

Error: second number not entered.

Choose option: 5

Error: first number not entered.

Error: second number not entered.

Choose option: 6

Error: first number not entered.

Error: second number not entered.

Choose option: 7

Error: first number not entered.

Error: second number not entered.

Choose option: 8

Error: first number not entered.

Error: second number not entered.

Choose option: 9

Error: first number not entered.

Choose option: 10

Error: first number not entered.

Error: second number not entered.

Choose option: 12

Error: incorrect input.

Choose option: 1

Enter real part: 2

Enter imaginary part: 2

Choose option: 3

First number: (2, 2).

Error: second number not entered.

Choose option: 4

Error: second number not entered.

Choose option: 5

Error: second number not entered.

Choose option: 6

Error: second number not entered.

Choose option: 7

Error: second number not entered.

Choose option: 8

Error: second number not entered.

Choose option: 9

Result: (2, -2).

Choose option: 10

Error: second number not entered.

Choose option: 2

Enter real part: 0

Enter imaginary part: 0

Choose option: 7

Error: division by zero.

Choose option: 11

1. Листинг программы

/\*

Гребенков Дмитрий М8О-207Б-19

ЛР 2 Вариант 1

Комплексное число в алгебраической форме представляются парой действительных чисел (a, b),

где a - действительная часть, b - мнимая часть. Реализовать класс Complex для работы с комплексными числами.

Обязательно должны быть присутствовать операции

- сложения add, (a, b) + (c, d) = (a + c, b + d);

- вычитания sub, (a, b) – (c, d) = (a – c, b – d);

- умножения mul, (a, b) ´ (c, d) = (ac – bd, ad + bc);

- деления div, (a, b) / (c, d) = (ac + bd, bc – ad) / (c2 + d2);

- сравнение equ, (a, b) = (c, d), если (a = c) и (b = d);

- сопряженное число conj, conj(a, b) = (a, –b).

Реализовать операции сравнения модулей.

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 <cstring>

#include <string>

#include <cmath>

class Complex {

public:

Complex () {}

Complex (double x, double y) {

a = x;

b = y;

}

double getRe () {

return a;

}

double getIm () {

return b;

}

void set (double x, double y) {

a = x;

b = y;

}

void add (Complex x) {

a += x.a;

b += x.b;

}

void sub (Complex x) {

a -= x.a;

b -= x.b;

}

void mul (Complex x) {

double buffer = (a \* x.a) - (b \* x.b);

b = (a \* x.b) + (b \* x.a);

a = buffer;

}

void div (Complex x) {

double buffer = ((a \* x.a) + (b \* x.b)) / ((x.a \* x.a) + (x.b \* x.b));

b = ((b \* x.a) - (a \* x.b)) / ((x.a \* x.a) + (x.b \* x.b));

a = buffer;

}

bool equ (Complex x) {

if ((a == x.a) && (b == x.b)) {

return true;

}

else {

return false;

}

}

Complex conj () {

return (Complex (a, - b));

}

bool com (Complex x) {

double moduleFirst = sqrt (pow (a, 2) + pow (b, 2));

double moduleSecond = sqrt (pow (x.a, 2) + pow (x.b, 2));

return (moduleFirst == moduleSecond);

}

~Complex () {}

private:

double a;

double b;

};

Complex operator+ (Complex first, Complex second) {

first.add (second);

return first;

}

Complex operator- (Complex first, Complex second) {

first.sub (second);

return first;

}

Complex operator\* (Complex first, Complex second) {

first.mul (second);

return first;

}

Complex operator/ (Complex first, Complex second) {

first.div (second);

return first;

}

bool operator== (Complex first, Complex second) {

return (first.equ (second));

}

int main () {

int option = 0;

double real = 0;

double imaginary = 0;

Complex first;

Complex second;

Complex third;

bool firstFlag = false;

bool secondFlag = false;

std::cout << "Menu:" << std::endl;

std::cout << "0. Print menu." << std::endl;

std::cout << "1. Enter the first complex number." << std::endl;

std::cout << "2. Enter the second complex number." << std::endl;

std::cout << "3. Print the first and second numbers." << std::endl;

std::cout << "4. Addition of the first and second numbers." << std::endl;

std::cout << "5. Subtract the second number from the first." << std::endl;

std::cout << "6. Multiply the first number by the second." << std::endl;

std::cout << "7. Divide the first number by the second." << std::endl;

std::cout << "8. Compare the first number with the second." << std::endl;

std::cout << "9. Print the conjugate of the first number." << std::endl;

std::cout << "10. Compare the modules of the first and second numbers." << std::endl;

std::cout << "11. Quit the program." << std::endl;

while (option != 11) {

std::cout << "Choose option: ";

std::cin >> option;

switch (option) {

case 0: {

std::cout << "Menu:" << std::endl;

std::cout << "0. Print menu." << std::endl;

std::cout << "1. Enter the first complex number." << std::endl;

std::cout << "2. Enter the second complex number." << std::endl;

std::cout << "3. Print the first and second numbers." << std::endl;

std::cout << "4. Addition of the first and second numbers." << std::endl;

std::cout << "5. Subtract the second number from the first." << std::endl;

std::cout << "6. Multiply the first number by the second." << std::endl;

std::cout << "7. Divide the first number by the second." << std::endl;

std::cout << "8. Compare the first number with the second." << std::endl;

std::cout << "9. Print the conjugate of the first number." << std::endl;

std::cout << "10. Compare the modules of the first and second numbers." << std::endl;

std::cout << "11. Quit the program." << std::endl;

break;

}

case 1: {

std::cout << "Enter real part: ";

std::cin >> real;

std::cout << "Enter imaginary part: ";

std::cin >> imaginary;

first.set (real, imaginary);

firstFlag = true;

break;

}

case 2: {

std::cout << "Enter real part: ";

std::cin >> real;

std::cout << "Enter imaginary part: ";

std::cin >> imaginary;

second.set (real, imaginary);

secondFlag = true;

break;

}

case 3: {

if (firstFlag) {

std::cout << "First number: (" << first.getRe () << ", " << first.getIm () << ")." << std::endl;

}

else {

std::cout << "Error: first number not entered." << std::endl;

}

if (secondFlag) {

std::cout << "Second number: (" << second.getRe () << ", " << second.getIm () << ")." << std::endl;

}

else {

std::cout << "Error: second number not entered." << std::endl;

}

break;

}

case 4: {

if (firstFlag && secondFlag) {

third = first + second;

std::cout << "Result: (" << third.getRe () << ", " << third.getIm () << ")." << std::endl;

}

else {

if (!firstFlag) {

std::cout << "Error: first number not entered." << std::endl;

}

if (!secondFlag) {

std::cout << "Error: second number not entered." << std::endl;

}

}

break;

}

case 5: {

if (firstFlag && secondFlag) {

third = first - second;

std::cout << "Result: (" << third.getRe () << ", " << third.getIm () << ")." << std::endl;

}

else {

if (!firstFlag) {

std::cout << "Error: first number not entered." << std::endl;

}

if (!secondFlag) {

std::cout << "Error: second number not entered." << std::endl;

}

}

break;

}

case 6: {

if (firstFlag && secondFlag) {

third = first \* second;

std::cout << "Result: (" << third.getRe () << ", " << third.getIm () << ")." << std::endl;

}

else {

if (!firstFlag) {

std::cout << "Error: first number not entered." << std::endl;

}

if (!secondFlag) {

std::cout << "Error: second number not entered." << std::endl;

}

}

break;

}

case 7: {

if (firstFlag && secondFlag) {

if ((pow (second.getRe (), 2) + pow (second.getIm (), 2)) != 0) {

third = first / second;

std::cout << "Result: (" << third.getRe () << ", " << third.getIm () << ")." << std::endl;

}

else {

std::cout << "Error: division by zero." << std::endl;

}

}

else {

if (!firstFlag) {

std::cout << "Error: first number not entered." << std::endl;

}

if (!secondFlag) {

std::cout << "Error: second number not entered." << std::endl;

}

}

break;

}

case 8: {

if (firstFlag && secondFlag) {

if (first == second) {

std::cout << "Numbers are equal." << std::endl;

}

else {

std::cout << "Numbers are not equal." << std::endl;

}

}

else {

if (!firstFlag) {

std::cout << "Error: first number not entered." << std::endl;

}

if (!secondFlag) {

std::cout << "Error: second number not entered." << std::endl;

}

}

break;

}

case 9: {

if (firstFlag) {

third = first.conj ();

std::cout << "Result: (" << third.getRe () << ", " << third.getIm () << ")." << std::endl;

}

else {

std::cout << "Error: first number not entered." << std::endl;

}

break;

}

case 10: {

if (firstFlag && secondFlag) {

if (first.com (second)) {

std::cout << "Numbers are equal in absolute value." << std::endl;

}

else {

std::cout << "Numbers are not equal in absolute value." << std::endl;

}

}

else {

if (!firstFlag) {

std::cout << "Error: first number not entered." << std::endl;

}

if (!secondFlag) {

std::cout << "Error: second number not entered." << std::endl;

}

}

break;

}

case 11: {

break;

}

default: {

std::cout << "Error: incorrect input." << std::endl;

break;

}

}

}

return 0;

}

1. Вывод

Получены основные навыки работы с перегрузкой операторов.

Литература:

1. C++ | Перегрузка операторов [Электронный ресурс] URL: [https://metanit.com/cpp/tutorial/5.14.php](https://metanit.com/cpp/tutorial/5.1.php) (дата обращения 09.10.2020)
2. Введение в перегрузку операторов в C++ | Уроки C++ - Ravesli [Электронный ресурс] URL: <https://ravesli.com/urok-130-vvedenie-v-peregruzku-operatorov/> (дата обращения 09.10.2020)