**Assignment # 7**

**Sai Kiran Reddy Gokula**

**Output:**

A screenshot of a computer

Description automatically generated

**Console Output:**

Enter the first complex number: 3.5 5.5

Enter the second complex number: -3.5 1

(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i

(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i

(3.5 + 5.5i) \* (-3.5 + 1.0i) = -17.75 + -15.75i

(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i

|(3.5 + 5.5i)| = 6.519202405202649

**Output:**

A screenshot of a computer

Description automatically generated

**Console Output:**

Enter the first complex number: 10.5 5

Enter the second complex number: -10.5 15

(10.5 + 5.0i) + (-10.5 + 15.0i) = 0.0 + 20.0i

(10.5 + 5.0i) - (-10.5 + 15.0i) = 21.0 + -10.0i

(10.5 + 5.0i) \* (-10.5 + 15.0i) = -185.25 + 105.0i

(10.5 + 5.0i) / (-10.5 + 15.0i) = -0.1051 + -0.6i

|(10.5 + 5.0i)| = 11.629703349613008

**Source Code:**

package a7;

import java.util.Scanner;

/\*\*

\* CPSC 24500-001- Object-Oriented Programming

\* Assignment 7

\* Complex class represents a complex number in the form a + bi, where a and b are real numbers.

\* The numbers a and b are known as the real part and imaginary part of the complex number

\* Performs basic operations on complex numbers.

\*/

public class Complex implements Comparable<Complex> {

private double a; // Real part

private double b; // Imaginary part

/\*\*

\* Complex Constructor with the given real and imaginary parts.

\* @param a The real part of the complex number

\* @param b The imaginary part of the complex number

\*/

public Complex(double a, double b) {

this.a = a;

this.b = b;

}

/\*\*

\* Complex Constructor with the real part and 0 as the imaginary part.

\* @param a The real part of the complex number

\*/

public Complex(double a) {

this(a, 0);

}

/\*\*

\* Complex - Copy Constructor of the given complex number.

\* @param complexNumber The complex number to copy

\*/

public Complex(Complex complexNumber) {

this(complexNumber.a, complexNumber.b);

}

//Constructs a Complex object representing the number 0 for a, b

public Complex() {

this(0, 0);

}

/\*\*

\* Gets the real part of the complex number.

\* @return The real part

\*/

public double getRealPart() {

return a;

}

/\*\*

\* Gets the imaginary part of the complex number.

\* @return The Imaginary part

\*/

public double getImaginaryPart() {

return b;

}

/\*\*

\* Adds the given complex number to this complex number. Formula : a + bi + c + di = (a + c) + (b + d)i

\* @param other The complex number to add

\* @return The result of the addition

\*/

public Complex add(Complex other) {

return new Complex(a + other.a, b + other.b);

}

/\*\*

\* Subtracts the given complex number from this complex number. Formula: a + bi - (c + di) = (a - c) + (b - d)i

\* @param other The complex number to subtract

\* @return The result of the subtraction

\*/

public Complex subtract(Complex other) {

return new Complex(a - other.a, b - other.b);

}

/\*\*

\* Multiplies this complex number by the given complex number. formula (a + bi) \* (c + di) = (ac - bd) + (bc + ad)i

\* @param other The complex number to multiply by

\* @return The result of the multiplication

\*/

public Complex multiply(Complex other) {

double realPart = a \* other.a - b \* other.b;

double imaginaryPart = b \* other.a + a \* other.b;

return new Complex(realPart, imaginaryPart);

}

/\*\*

\* Divides this complex number by the given complex number. (a + bi)/(c + di) = (ac + bd)/(c^2+d^2) + (bc - ad)i/(c^2+ d^2)

\* @param other The complex number to divide by

\* @return The result of the division

\*/

public Complex divide(Complex other) {

double denominator = other.a \* other.a + other.b \* other.b;

double realPart = (a \* other.a + b \* other.b) / denominator;

double imaginaryPart = (b \* other.a - a \* other.b) / denominator;

return new Complex(realPart, imaginaryPart);

}

/\*\*

\* Calculates the absolute value of this complex number. |a+bi|= √(a^2+b^2 )

\* @return The absolute value

\*/

public double abs() {

return Math.sqrt(a \* a + b \* b);

}

// Override toString method

@Override

public String toString() {

if (b != 0) {

return String.format("%.1f + %.1fi", a, b);

} else {

return String.format("%.1f", a);

}

}

/\*\*

\* Compares this complex number to another complex number based on their absolute values.

\* @param other The complex number to compare to

\* @return complex number

\*/

@Override

public int compareTo(Complex other) {

return Double.compare(this.abs(), other.abs());

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the first complex number: ");

double a1 = scanner.nextDouble();

double b1 = scanner.nextDouble();

Complex c1 = new Complex(a1, b1);

System.out.print("Enter the second complex number: ");

double a2 = scanner.nextDouble();

double b2 = scanner.nextDouble();

Complex c2 = new Complex(a2, b2);

// Addition

System.out.println("("+c1 + ") + (" + c2 + ") = " + c1.add(c2));

// Subtraction

System.out.println("("+c1 + ") - (" + c2 + ") = " + c1.subtract(c2));

// Multiplication

Complex c = c1.multiply(c2);

System.out.println("("+c1 + ") \* (" + c2 + ") = " + c.getRealPart()+" + "+c.getImaginaryPart()+"i");

// Division

Complex c3 = c1.divide(c2);

System.out.printf("("+c1 + ") / (" + c2 + ") = %.4f + %.1fi%n", c3.getRealPart(),c3.getImaginaryPart());

// Absolute value

System.out.println("|(" + c1 + ")| = " + c1.abs());

}

}