**CIS 368**

**Section 50**

**Assignment 5**

**Michael Stiffler - 2731139**

**Tarik Taye - 2715602**

Problem Description:

Design a class named Complex for representing complex numbers and the methods add, subtract, multiply, divide, abs for performing complex-number operations, and override toString method for returning a string representation for a complex number. The toString method returns a + bi as a string. If b is 0, it simply returns a. Provide three constructors Complex(a, b), Complex(a), and Complex(). Complex() creates a Complex object for number 0 and Complex(a) creates a Complex object with 0 for b. Also provide the getRealPart() and getImaginaryPart() methods for returning the real and imaginary part of the complex number, respectively. Your Complex class should also implement the Cloneable interface. Write a test program that prompts the user to enter two complex numbers and display the result of their addition, subtraction, multiplication, and division. What is asked of us:

Submit the following items:

1. Compile, Run, and Submit word or pdf file to blackboard (you must submit the program regardless whether it complete or incomplete, correct or incorrect)

2. Submit screen shots of sample run

**Just compile and run,**

**Code asks you:**

Enter the first complex number:

Example: 3.5 5.5

Enter the second complex number:

Example: -3.5 1

Results:

(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

False

3.5

5.5

THE PROGRAM IS COMPLETE

**Diagram**:

Table

Description automatically generated with medium confidence

Screenshots:

Sample Run

**Enter the first complex number:**

Text

Description automatically generated with low confidence

**Enter the second complex number:**

Text

Description automatically generated

**Results:**

Text

Description automatically generated

**Code:**

import java.util.Scanner;  
import java.lang.Math;  
  
public class Test {  
 public static void main(String[] args) throws CloneNotSupportedException {  
 Scanner input = new Scanner(System.in);  
 System.out.print("Enter the first complex number: ");  
 double a = input.nextDouble();  
 double b = input.nextDouble();  
 Complex c1 = new Complex(a, b);  
 System.out.print("Enter the second complex number: ");  
 double c = input.nextDouble();  
 double d = input.nextDouble();  
 Complex c2 = new Complex(c, d);  
 System.out.println("(" + c1 + ")" + " + " + "(" + c2 + ")" + " = "  
 + c1.add(c2));  
 System.out.println("(" + c1 + ")" + " - " + "(" + c2 + ")" + " = "  
 + c1.subtract(c2));  
 System.out.println("(" + c1 + ")" + " \* " + "(" + c2 + ")" + " = "  
 + c1.multiply(c2));  
 System.out.println("(" + c1 + ")" + " / " + "(" + c2 + ")" + " = "  
 + c1.divide(c2));  
 System.out.println("|" + c1 + "| = " + c1.abs());  
 Complex c3 = (Complex) c1.clone();  
 System.out.println(c1 == c3);  
 System.out.println(c3.getRealPart());  
 System.out.println(c3.getImaginaryPart());  
 }  
}  
  
class Complex implements Cloneable {  
 private double real;  
 private double imaginary;  
  
 Complex(double real, double imaginary) {  
 this.real = real;  
 this.imaginary = imaginary;  
 }  
  
 Complex(double real) {  
 this.real = real;  
 this.imaginary = 0;  
 }  
  
 Complex() {  
 this.real = 0;  
 this.imaginary = 0;  
 }  
  
 public String getRealPart() {  
 return String.valueOf(this.real);  
 }  
  
 public String getImaginaryPart() {  
 return String.valueOf(this.imaginary);  
 }  
  
 public Complex add(Complex c2) {  
  
 double newA = this.real + Double.parseDouble(c2.getRealPart());  
 double newB = this.imaginary + Double.parseDouble(c2.getImaginaryPart());  
 return new Complex(newA, newB);  
 }  
  
 public Complex subtract(Complex c2) {  
 double newA = this.real - Double.parseDouble(c2.getRealPart());  
 double newB = this.imaginary - Double.parseDouble(c2.getImaginaryPart());  
 return new Complex(newA, newB);  
 }  
  
 public Complex multiply(Complex c2) {  
 double newAC = this.real \* Double.parseDouble(c2.getRealPart());  
 double newBD = this.imaginary \* Double.parseDouble(c2.getImaginaryPart());  
  
 double newBC = this.imaginary \* Double.parseDouble(c2.getRealPart());  
 double newAD = this.real \* Double.parseDouble(c2.getImaginaryPart());  
  
 double left = newAC - newBD;  
 double right = newBC + newAD;  
 return new Complex(left, right);  
 }  
  
 public Complex divide(Complex c2) {  
 double newAC = this.real \* Double.parseDouble(c2.getRealPart());  
 double newBD = this.imaginary \* Double.parseDouble(c2.getImaginaryPart());  
  
 double newBC = this.imaginary \* Double.parseDouble(c2.getRealPart());  
 double newAD = this.real \* Double.parseDouble(c2.getImaginaryPart());  
  
 double cSquared = Math.pow(Double.parseDouble(c2.getRealPart()), 2);  
 double dSquared = Math.pow(Double.parseDouble(c2.getImaginaryPart()), 2);  
  
 double left = (newAC + newBD) / (cSquared + dSquared);  
 double right = (newBC - newAD) / (cSquared + dSquared);  
 return new Complex(left, right);  
 }  
  
 @Override  
 public String toString() {  
 return this.real + " + " + this.imaginary + "i";  
 }  
  
 public String abs() {  
 double aSquared = Math.pow(this.real, 2);  
 double bSquared = Math.pow(this.imaginary, 2);  
  
 return String.valueOf(Math.sqrt(aSquared + bSquared));  
 }  
  
 @Override  
 protected Object clone()  
 throws CloneNotSupportedException {  
 return super.clone();  
 }  
}