COSC 237

Assignment # 2

DUE DATE: 10/18/2022 (Tuesday)

For this assignment you have 2 programming problems involving Java **user-created classes**. For each, turn in the source files, for the user-created class and the client (Complex.java, ComplexClient.java, Matrix.java, and MatrixClient.java). Submit the source files (both 1 & 2):

1. e-mail (izimand@towson.edu) - only one e-mail for the entire team (COSC237.section, your name, Assignment#2 in the Subject box). **You must Cc** all your teammates - I may need to contact all of you as a group. **AND**

2. turn in a hard copy of the assignment with all your names on it. Please staple all pages together (a must!).

You will work in teams of (at most) three students, but all teammates should contribute seriously (please let me know if that's not the case). In principle, you should not discuss the homework with anyone, except your partners and me. If you do discuss it with someone else (better don't), you should say this in the preamble of the assignment. You should only submit work that is completely your own and you should be able to explain your solutions if asked to do so.
Make sure you handle **invalid input, including type mismatch** - input validation is a requirement! Focus on **good design**, **good style**, and, most importantly, **reuse of code**. Start early and remember the rules about late assignments/syllabus → **no late assignments!**

NOTE: Try to use this quote as a guide when writing code (I do!): *"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."*
--- Martin Fowler, Refactoring: Improving the Design of Existing Code

HONORS STATEMENT: "I will not cheat, fabricate materials, plagiarize, or help another to undertake such acts of academic dishonesty, nor will I protect those who engage in acts of academic dishonesty. I pledge on my honor as a student, that I have not received or provided any unauthorized help on this assignment."

---

1. A complex ("imaginary") number has the form a + bi, where a is called the real part, b is called the imaginary part, and i = sqrt(-1). A complex number a + bi can be expressed as the ordered pair of real numbers (a, b).
Arithmetic operations on two complex numbers (a, b) and (c, d) are as follows:

Addition: (a, b) + (c, d) = (a + c, b + d)
Subtraction: (a, b) - (c, d) = (a - c, b - d)
Multiplication: (a, b) * (c, d) = (a * c - b * d, a * d + b * c)
Division: (a, b) / (c, d) = ((a * c + b * d)/(c^2 + d^2), (b * c - a * d)/(c^2 + d^2))
Absolute value: |(a, b)| = sqrt(a^2 + b^2)

Design and implement a **ComplexNumber** class that represents the real and imaginary parts as double values and provides at least the following methods:

- **Constructors** for default and explicit initialization.
- A method to **read** (get input) a complex number. Look at the sample output screen for the design required.
- A method to **print** a complex number as (a, b). Have 2 decimals for both.
- A method called **getReal** that returns the real part of a complex number.
- A method called **getImaginary** that returns the imaginary part of a complex number.
- Methods **equals, copy, getCopy, toString**.
- Arithmetic methods to **add**, **subtract**, **multiply**, and **divide** two complex numbers.
- A method called **cAbs** to implement the absolute value of a complex number.

To test your class write a client that has at least a function **menu()** with options for the methods implemented and an option to exit. Your program should loop until the user chooses to exit. In this loop you are required to use a **switch** statement for all possible cases (similar design as the one used for Problem#1 in Assignment#1).

SAMPLE OUTPUT:

```
Your options for Complex arithmetic are:
----------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 2

Enter complex number (real imaginary): 3.4 5.6
Enter complex number (real imaginary): 1.23 2.56
First complex number is: (3.40, 5.60)
Second complex number is: (1.23, 2.56)
Result: (3.40, 5.60) - (1.23, 2.56) = (2.17, 3.04)
   Command number 1 completed.

Your options for Complex arithmetic are:
----------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 4

Enter complex number (real imaginary): 11.2 22.1
Enter complex number (real imaginary): 1.45 3.56
First complex number is: (11.20, 22.10)
Second complex number is: (1.45, 3.56)
Result: (11.20, 22.10) / (1.45, 3.56) = (6.42, -0.53)
   Command number 2 completed.
```

```
Your options for Complex arithmetic are:
---------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 2

Enter complex number (real imaginary): 1.78 4.5
Enter complex number (real imaginary): 3.56 8.9
First complex number is: (1.78, 4.50)
Second complex number is: (3.56, 8.90)
Result: (1.78, 4.50) - (3.56, 8.90) = (-1.78, -4.40)
   Command number 3 completed.

Your options for Complex arithmetic are:
---------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 3

Enter complex number (real imaginary): 2.22 3.33
Enter complex number (real imaginary): 1.24 2.45
First complex number is: (2.22, 3.33)
Second complex number is: (1.24, 2.45)
Result: (2.22, 3.33) * (1.24, 2.45) = (-5.41, 9.57)
   Command number 4 completed.

Your options for Complex arithmetic are:
---------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 6

Enter complex number (real imaginary): 1.11 2.22
Enter complex number (real imaginary): 1.11 2.22
First complex number is: (1.11, 2.22)
Second complex number is: (1.11, 2.22)
The complex numbers are equal.
   Command number 5 completed.
```

```
Your options for Complex arithmetic are:
---------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 6

Enter complex number (real imaginary): 1.2 2.3
Enter complex number (real imaginary): 11.2 2.3
First complex number is: (1.20, 2.30)
Second complex number is: (11.20, 2.30)
The complex numbers are NOT equal.
   Command number 6 completed.

Your options for Complex arithmetic are:
---------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 5

Enter complex number (real imaginary): 11.1 22.2
The complex number is: (11.10, 22.20)
Result: |(11.1, 22.2)| = 24.82
   Command number 7 completed.

Your options for Complex arithmetic are:
---------------------------------------
 1) Add 2 complex numbers
 2) Subtract 2 complex numbers
 3) Multiply 2 complex numbers
 4) Divide 2 complex numbers
 5) Absolute value of a complex number
 6) Compare 2 complex numbers
 0) EXIT
Please enter your option: 0
Testing completed.
```

---

2. Design and implement a Java class for matrix arithmetic using square matrices (same number of rows and columns). Your solution will use a class called **Matrix.** When designing the class method members, use as a guide the program you wrote for the previous assignment (you should have all the algorithms figured out by now). Provide at least the following methods:

- **Constructors** for default and explicit initialization.
- A method to **get the size** of the matrix

- A method to **get the element** in row r and column c.
- A method to **set the element** in row r and column c to a particular value.
- A method to **initialize** the matrix with random numbers between a lower limit and an upper limit.
- Methods for **addition**, **subtraction**, and **multiplication**
- A method **equals** to compare 2 matrices for equality
- A method to **multiply** a matrix **by a constant**.
- A method to **transpose** a matrix.
- A method to find the **trace** of a matrix.
- A method to **print** a matrix.
- A method **copy** and a method **getCopy** for a matrix.

To make your job easier, I give you here the layout of the Matrix class (Matrix.java):

```
//ASSIGNMENT #2: MATRIX ARITHMETIC/class Matrix
import java.util.Scanner;
import java.util.Random;

public class Matrix {
    public final int MAX = 20;

    private int size;
    private int[][] table = new int[MAX][MAX];

    public Matrix() {... }
    public Matrix(int s) {... }
    public int getSize() {... }
    public int getElement(int r, int c) {... }
    public void setElement(int r, int c, int value) {... }
    public void init(int low, int up) {... }
    public void print() {... }
    public Matrix add(Matrix a){... }
    public Matrix subtract(Matrix a) {... }
    public Matrix multiply(Matrix a) {... }
    public Matrix multiplyConst(int whatConst) {... }
    public Matrix transpose() {... }
    public int trace() {... }
    public boolean equals(Matrix a){... }
    public void copy(Matrix a) {... }
    public Matrix getCopy() {... }
}
```

Write the code for class Matrix. Next, write a client that has at least a function **menu()** with options for the methods implemented and an option to exit. Your program should loop until the user chooses to exit. In this loop you are required to use a **switch** statement for all possible cases (similar design as the one used in the previous program). Look at the sample output to figure out how the client should work. Again, to make your job easier, I get you started with the client program:

```
//ASSIGNMENT #2: MATRIX ARITHMETIC/client
import java.util.Scanner;
import java.util.Random;

public class MatrixClient {
    public static final int MAX = 20;
```

```
    public static final int LOW = 1;
    public static final int UP = 10;

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int choice; //operation to be executed from menu
        int numCommands = 0; //display counter
        int size; //for subarray processing
        int value; //multiply matrix by this scalar
        int tr; //return from trace()
        //MISSING  CODE: input size. Valid type? Valid range?

        Matrix first = new Matrix(size);
        Matrix second = new Matrix(size);
        Matrix result = new Matrix(size);
        choice = menu(); //priming read;
        while (choice != 0)  {

              ---
         }
         ---
      }
      ---
```

SAMPLE OUTPUT:

```
Enter the size of the square matrix: 25
INPUT ERROR!!! Invalid size. Positive and <= 20.
Enter the size of the square matrix: 5

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT

Please enter your option: 7
The original matrix is:
    6    7    2    9    1
   10    5    2    4   10
   10    8    6    4    3
    3    2    2    3    9
    2   10    8    8    4
The copy of this matrix is:
    6    7    2    9    1
   10    5    2    4   10
   10    8    6    4    3
    3    2    2    3    9
    2   10    8    8    4
```

```
Testing for equality. Should be equal!!
The matrices are equal!!
   Command number 1 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 8
First matrix is:
    1    2   10    9   10
    7   10    5    5    9
   10    3    4    1    6
   10    6    1    9    8
    6    6    3   10    1
Second matrix is:
   10    9    4    2    3
    7    6    8    5   10
    8    5    9    7    3
    9    3    5    3    9
    6    3    8    5    7
The matrices are NOT equal!!
   Command number 2 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 1
First matrix is:
    9    4   10    1   10
    8    2    9    3    1
    5    9    8    7    9
   10    7    3    4    1
    7    5    5    2    7
Second matrix is:
   10    9    4    1    1
    9    3    8    3    1
    3    4    7   10    5
    8    9    8    9    2
    1    3    8    2    7
The resulting matrix is:
```

```
   19   13   14    2   11
   17    5   17    6    2
    8   13   15   17   14
   18   16   11   13    3
    8    8   13    4   14
   Command number 3 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT

Please enter your option: 3
First matrix is:
    3   10    1    8    8
    6    9    4   10    7
    7    7    2    5    5
    8    6   10   10    6
    7    3    2   10    6
Second matrix is:
    9    4   10    1    1
    2    7    3    6    7
    2    9    9   10    2
   10    5    8    1    7
    4    9   10    2    4
The resulting matrix is:
  161  203  213   97  163
  208  236  273  124  175
  151  165  199   84  115
  228  268  328  166  164
  197  171  237   67  126
   Command number 4 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 2
First matrix is:
    2    3    1    6    3
   10    6    3    9    2
    4    1    8    5   10
```

```
    1    7    3    6    4
    5    6    4    9    5
Second matrix is:
    4    9    7    7    9
    8    1    2    5    2
    4    1   10    7    8
    3    9    7    6    7
    4    9    7    2    9
The resulting matrix is:
   -2   -6   -6   -1   -6
    2    5    1    4    0
    0    0   -2   -2    2
   -2   -2   -4    0   -3
    1   -3   -3    7   -4
   Command number 5 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 4
Enter the multiplication constant: 10
The original matrix is:
    8    8   10    4    9
    1    3    5    9    6
    9   10   10    4    5
    7    1    8    3    3
    9    1    9    9    3
The resulting matrix is:
   80   80  100   40   90
   10   30   50   90   60
   90  100  100   40   50
   70   10   80   30   30
   90   10   90   90   30
   Command number 6 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 6
The original matrix is:
```

```
    8    6    9    3    1
    5    3   10    1    7
    7    1    1    8    2
   10    8    4   10    2
    5    4    2    7    9
The trace for this matrix is: 31
   Command number 7 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 5
The original matrix is:
    7    5    2    1    8
    1    3    1    5   10
    6    6    3    8    5
    8    5    3   10    5
   10    7    3   10   10
The resulting matrix is:
    7    1    6    8   10
    5    3    6    5    7
    2    1    3    3    3
    1    5    8   10   10
    8   10    5    5   10
   Command number 8 completed.

Your options are:
-----------------
 1) Add 2 matrices
 2) Subtract 2 matrices
 3) Multiply 2 matrices
 4) Multiply matrix by a constant
 5) Transpose matrix
 6) Matrix trace
 7) Make a copy
 8) Test for equality
 0) EXIT
Please enter your option: 0
Testing completed.
```