

**Sofia University**  
**Department of Mathematics and Informatics**

Course : **OO Programming Java**

Date: **October 31, 2022**

Student Name:

**Homework No. 4**

**Submit the IntelliJ projects, where all Java files developed to solve the problems listed below.**  
**Use comments and Modified-Hungarian notation.**

**Problem No. 1 (use the sample code Lab3bSampleCodeFig7.rar from Week3, Practical work)**

Coupon collector is a classic statistics problem with many practical applications. The problem is to pick objects from a set of objects repeatedly and find out how many picks are needed for all the objects to be picked at least once. A variation of the problem is to pick cards from a shuffled deck of 52 cards repeatedly and find out how many picks are needed before you see one of each suit. Assume a picked card is placed back in the deck before picking another. Write a program to simulate the number of picks needed to get four cards from each suit and display the four cards picked (it is possible a card may be picked twice). Here is a sample run of the program:

```
Queen of Spades
5 of Clubs
Queen of Hearts
4 of Diamonds
Number of picks: 12
```

**Problem 2**

Write a class **AlgebraUtils** with a **static method to multiply two matrices**. The definition of the method is:

```
public static double[][] multiplyMatrix(double[][] a, double[][] b)
```

To multiply matrix **a** by matrix **b**, the number of columns in **a** must be the same as the number of rows in **b**, and the two matrices must have elements of the same or compatible types. Let **c** be the result of the multiplication. Assume the column size of matrix **a** is **n**. Each element  $c_{ij}$  is

$$a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \dots + a_{in} \times b_{nj}$$

For example, for two  $3 \times 3$  matrices **a** and **b**, **c** is

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

where  $c_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + a_{i3} \times b_{3j}$ .

**A. Create a jar file with class **AlgebraUtils****

**B. Create a separate IntelliJ project with class **AlgebraUtilsTst** that imports the class **AlgebraUtils****

and in its `main()` method **generates** two  $3 \times 3$  matrices with **random 2- digit numbers** and **displays** **their product**.

```
Enter matrix1: 1 2 3 4 5 6 7 8 9 Enter
Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2 Enter
The multiplication of the matrices is
1 2 3      0 2.0 4.0      5.3 23.9 24
4 5 6      * 1 4.5 2.2 = 11.6 56.3 58.2
7 8 9      1.1 4.3 5.2    17.9 88.7 92.4
```

### Problem No.3.

Draw a square-shaped spiral (as in the screen capture given below), centered in the application window, using a JavaFX application. One technique is to use a loop that increases the line length after drawing every second line. The direction in which to draw the next line should follow a distinct pattern, such as down, left, up, right.

