CS102A Introduction to Computer Programming
Fall 2020 Lab 6

Objective

1. Learn how to use static method.
2. Learn method overloading.
3. Learn how to use two dimensional arrays.
4. Learn invoking methods with array arguments and getting back the values.

1 Exercise

1.1 Exercise 1

Create a class named `MyTriangle` that contains two static methods

```
public static double area(double a, double b, double c)
```

```
public static double perimeter(double a, double b, double c)
```

to compute area and perimeter of a triangle respectively given three valid sides `a`, `b`, and `c`. And add a static method

```
/** Return true if the sum of any two sides is greater than the third side
    . **/
public static boolean isValid(double a, double b, double c)
```

In the `main` method of `MyTriangle`, test the three methods you write.

1. Get `a`, `b`, and `c` from the console.
2. If `a` is `-1`, exit your program and print *Bye* .

3. If `a` is not `-1`, use `isValid` to check the input.
4. If the input is valid, compute the area and perimeter and print them.
5. If the input is not valid, return false and print *The input is invalid.*
6. Go to 1.

> **Tip**
> To call a method in the same class, you can try `method_name()`.

Sample input and output:

```
Please input three numbers for a, b, c:
1 1 2
The input is invalid.
Please input three numbers for a, b, c:
2 3 4
The area is 2.905
The perimeter is 9.000
Please input three numbers for a, b, c:
3.2 4.3 3.4
The area is 5.377
The perimeter is 10.900
Please input three numbers for a, b, c:
-1
Bye~
```

1.2 Exercise 2

In the `MyTriangle` class created in Exercise 1, add two another static overloaded methods

```
1 public static double area(double bottom, double height)
```

```
1 public static double area(double a, double b, int angleOfAandB)
```

to compute the area. The first method is to compute area by bottom and height: `area` $= 1/2 \times$ `bottom` $\times$ `height`. The second method is to compute area by two sides `a`, `b` and the angle between the two sides(`angleOfAandB`): `area` $= 1/2 \times$ a $\times$ b $\times \sin($`angleOfAandB`$)$.

Then create another class `Lab6E2` that contains the `main` method. In the main method:

1. Read bottom and height from the console to compute area by calling the corresponding method you created in `MyTriangle`.
2. Read two sides `a`, `b`, and `angleOfAandB` from the console to compute area by calling the corresponding method you created in `MyTriangle`.

> **Tip**
> To call a static method in another class `class_name` under the same file directory, you can try `class_name.method_name()`.

Sample input and output:

```
Please input two numbers for bottom and height:
4 5.6
The area is 11.200
Please input two numbers for a and b:
3 5.6
Please input a number in (0, 180) for angle (angle is a float variable):
55
The area is 6.881
```

1.3 Exercise 3

Write a program to get students' grades from their courses and then print the scores and average scores in a grade table.

1. Prompt the user to enter the number of students (less than 10) and the number of courses (less than 10).
2. Prompt the user to enter the course scores for each student. The scores from different courses are entered on separate lines. On each line, there are scores for a course for each student.
3. Print a grade table. The first row shows the course names and the first column shows the student names. The last row shows the average scores of each course and the last column shows the average scores of each students.

Sample input and output:

```
Please enter the number of subjects: 3
Please enter the number of students: 4
32 44 52 32
```

```
89 92 80 94
11 22 32 23

          Course1    Course2    Course3    Average
Student1     32         89         11       44.00
Student2     44         92         22       52.67
Student3     52         80         32       54.67
Student4     32         94         23       49.67
Average     40.00      88.75      22.00
```

1.4 Exercise 4

Write a program to calculate the product of n matrices.

1. Read the number of matrices from user.
2. Read the elements of all the matrices from user. Before the elements of each matrix, the user should input the rows and columns of that matrix.
3. Print the result.

Sample input and output:

```
Please enter the number of matrices: 3
Enter the number of row and column of matrix 1: 3 5
Enter the elements of the matrix:
6 -7 3 -5 1
0  4 8  2 3
3 -2 1 -7 2
Enter the number of row and column of matrix 1: 5 1
Enter the elements of the matrix:
0
9
-3
4
1
Enter the number of row and column of matrix 1: 1 3
Enter the elements of the matrix:
-1 3 9
The results:
```

```
91   -273 -819
-23  69   207
47   -141 -423
```

1.5 Exercise 5

Sudoku is a famous mathematical game in which players fill numbers 1–9 in a 9×9 square. The square satisfies that every row and every column contain 1–9 only once. Specially, the square is divided into 9 sub-squares, and every sub-squares also contains 1–9 only once. Write a program to judge whether a 1–9 square is a Sudoku square.

1. Get a 9×9 square from console.
2. If it is a Sudoku square, print *Yes*.
3. If it is not a Sudoku square, print *No*.

Sample input and output:

```
2 9 3 7 1 5 4 8 6
8 6 1 2 4 9 5 3 7
7 4 5 8 6 3 1 9 2
6 7 8 9 2 1 3 4 5
1 3 9 5 7 4 2 6 8
4 5 2 6 3 8 7 1 9
9 2 4 3 8 7 6 5 1
3 8 6 1 5 2 9 7 4
5 1 7 4 9 6 8 2 3
Yes
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
No
1 2 3 1 2 3 1 2 3
```

```
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9
1 2 3 1 2 3 1 2 3
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9
1 2 3 1 2 3 1 2 3
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9
No
```