Basis of Computer Programming (Java A)
Lab Exercise 6

[Experimental Objective]

- Learn how to use static method
- Learn how to use static method from other class
- Learn how to use method overloading
- Learn to develop and invoke methods with array arguments and return values.

[Exercises]

1. Create a class named *MyTriangle* that contains two static methods

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

and

```
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.

Then create another class *Question1* that contains the main method. In the main method, test the two methods you write in *MyTriangle*.
Here is a sample run.

```
Please input three numbers for a, b, c:
3 4 5
The area is 6.000
The perimeter is 12.000
```

2. In the *MyTriangle* class created in Question 1, 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)
```

and modify the method

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

created in Question 1 such that before computing the area, check whether the three sides can create a valid triangle. If the triangle is valid, compute the area and return it; if it is not valid, return -1.0;

Then create another class *Question2* that contains the main method. In the main method, read three sides for a triangle and computes the area if the input is valid. Otherwise, it displays that the input is invalid. Here are samples runs.

```
Please input three numbers for a, b, c:
1 2 3
The input is invalid
```

```
Please input three numbers for a, b, c:
3 4 5
The area is 6.000
```

3. In the *MyTriangle* class created in Question 1, add two another static overloaded methods

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

and

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

to compute area. The first overloaded method is to compute area by bottom and height (area = 1/2 * bottom * height); the second overloaded method is to compute area by two sides a, b and the angle between the two sides (area = 1/2 * a * b * sin(angleOfAandB)).

Then create another class *Question3* that contains the main method. In the main method, firstly, read bottom and height from the Console to compute area by calling the corresponding method you created in MyTriangle; then read two sides a, b and the angle between the two sides from the Console to compute area by calling the corresponding method you created in MyTriangle.
Here is a samples run.

```
Please input two numbers for bottom and height:
3 4
The area is 6.000
Please input two numbers for a and b:
3 4
Please input a number in (0, 180) for angle (angle is a float variable):
90.0
The area is 6.000
```

4. In the *MyTriangle* class created in Question 1, add two another static overloaded methods

```
public static void displayTriangle(int lineofHeight)
```

and

```
public static void displayTriangle(int lineofHeight, char ch)
```

to display a triangle. If you just pass one integer to the method, it will display the lines according to that integer, first line all "0"s, second line all "1"s, third line all "3"s, and so on. And if you pass two arguments to the method, it will also display the number of lines according to the first argument and the display contents are the second argument.

Then create another class *Question3* that contains the main method. In the main method, firstly, read a integer from the Console to display triangle by calling the corresponding method you created in MyTriangle; then read an integer and a char from the Console to display triangle by calling the corresponding method you created in MyTriangle.

Here is a sample run.

```
Please input an integer in [2, 9]:
5
          1
        2 2 2
      3 3 3 3 3
    4 4 4 4 4 4 4
  5 5 5 5 5 5 5 5 5
Please input an integer in [2, 9]:
3
Please input a char:
*
      *
    * * *
  * * * * *
```